CCLI PHASE 2: SAGE MATHEMATICAL SOFTWARE

Contents

Title: An Open Approach to Realizing the Potential of Mathematics Software in the Undergraduate Curriculum (shorter?)

1. Project Description

Software for mathematical explorations, often known as "computer algebra systems," have held great promise for mathematics education since the appearance of Maple in the early 1980's and Mathematica in the late 1980's. Studies have documented the improvement in students' understanding [Tom - back this up with one or two sentences]. Yet the use of these systems is rarely integrated well with the curriculum. To accomodate competing systems, textbooks typically offer supplements for several different programs or generic "technology" sidebars. Only rarely does a textbook take a chance by tying itself to the strengths of one particular commercial system. Licensing restrictions for campus use, the expense of personal copies, and/or underpowered hardware means students must often work with the software only in campus labs. The Calculus & Mathematica project generated much excitement in the early 1990's, but today its use seems limited to the two institutions where it originated. Why have we not seen the broad transformative effect of these powerful tools for increasing the understanding of mathematics? What is needed to increase the uptake of software for mathematical exploration and have it be an essential ingredient in the undergraduate mathematics curriculum?

Our proposition is that open software, open standards and open licenses can allow mathematics software to effect the long envisioned transformation of the undergraduate mathematics curriculum. We will test this hypothesis using the open source mathematics software, Sage, integrated into existing open textbooks and new, shorter curricular materials, authored with LaTeX, the open standard for technical typesetting. By creating a system to easily convert textbooks and other materials into electronic Sage worksheets, we will place the full computational power of Sage *directly into* a student's text, usable at all times and from anywhere, simply via a web browser. For the institution and instructor, the cost and time-consuming inconveniences of commercial software are removed by free software, installed on servers accessible with no license restrictions. Further, the essential nature of open software means that curricular decisions and needs can drive the development of the software, with classroom teacher and software developer frequently being one and the same. Leveraging the inherent advantages of an open approach to software and text, the promise of mathematics software for education can be fully realized by faculty and students.

More specifically, we will build and test a model for easily integrating mathematics software and open educational resources into the mathematics curriculum and classroom by:

- integrating Sage with existing mature open textbooks and creating other open Sage-enhanced curricular materials, plus building a model and tools that allow others to easily do the same;
- partnering with diverse institutions to test and refine this model and materials in a wide variety of courses while providing support for teachers;
- conducting formative and summative evaluation to determine the effectiveness of our model in making open materials easy to use, and measuring the resulting impact on teaching practices and the learning of mathematics.

1.1. **Three Crises in the Academy.** The crisis in research publication is well-known. Faculty have come a long way in their efforts to take back control from commercial scientific publishers. Electronic journals and public repositories now publish articles with open licenses that explicitly allow the free sharing of new ideas among research communities and others, without copyright fees and with ubiquitous access via the internet. Government initiatives, such as the Public Access Policy of the National Institutes of Health, are working to accelerate this trend.

Following on the sea change in research publication, the next wave is open textbooks. Again, the problems with commercial control of textbooks, such as high prices, edition churn, and orphaned works, are all too familiar to faculty. Open licenses are now being used to take back control of these critical resources for education, with faculty in mathematics, computer science and business as the leaders. Government is poised to similarly accelerate this trend at all levels, with a recent bill before Congress (cite Durbin) directing the Department of Education to award grants for the creation of open textbooks by faculty, Washington State's initiative to provide open textbooks for their eighty highest-enrollment courses in the community colleges, and California's initiative to create free digital textbooks for its high schools.

With the emergence of viable and comprehensive open source software for mathematics, there is now a spectacular opportunity for faculty to use, modify, create, and extend this critical tool for mathematics education and allow it to reach its full potential in the service of educating students. This project will ease the path for faculty to make the transition to using open software and open textbooks in their courses. Tools for authoring materials using open standards, existing open textbooks, networks made ubiquitous by adherence to open standards, software built from open components and held together with an open industry-standard language, and a dedication to an open development process will combine to make it far easier for schools and faculty to deliver the full power of mathematics software to students. This proposal is an opportunity for government to accelerate these inevitable changes.

2. Why Sage?

New point: Sage can do all of numerical and symbolic and algebraic due to its modular construction and adherence to mathematical constructions/precision/truth (ie "real" mathematical objects). Plug-in Geogebra? (Can we say this about Geogebra?) Sage can do it all, no Matlab-numeric vs. Mathematica-symbolic choice. Comprehensive, and if not, it can be added easily. Combine this with the 100 packages description from elsewhere.

(From previous)

Sage is the most natural computer algebra system to realize the vision outlined above. The open development model, the notebook and LaTeX interfaces, the underlying language and philosophy behind how Sage is built, and the free nature all contribute toward this.

- Open development model - Like mathematics, the Sage model for development is driven by collaboration and openness (is this a controversial statement about mathematics?). Just as research topics drive the direction of pure mathematics, the curriculum and needs of the users directly drives the development. As an example of how teaching has driven the development, one of the investigators on this grant was asked one day by a student if Sage had a 3D vector field plotting function. He quickly wrote one. Encouraged by the open community that is common among open-source projects such as Sage, he posted his code online in the ticket-tracking website. Another developer that had written most of the 3d graphing in Sage suggested a small change that greatly increased the efficiency of the function. Several months later, another person (tsarnold) authored a very similar function which had slightly improved options and published their code on the public Sage notebook server. In Fall 2008, the investigator needed the 3d vector field plotting function for his multivariable class. He made a few changes to the code based on suggestions, and posted it for his class to use. In a closed proprietary system, that's where things likely would have stopped. Since that time, several other deep changes to the internals of Sage have been made to simplify code in functions like the 3d plotting, some of which were directly the result of feedback about the function. The investigator recently taught the same class again and needed the function again. He incorporated the good ideas from the various sources into his function and posted a documented, efficient version of the function to the trac ticket and asked for review. Another Sage developer reviewed the function immediately, with the comment "Very nice, positive review. This is great timing since I am about to teach vector fields in a week or two." The function was incorporated into Sage and released with the next version two weeks later. Now not just the original students benefit from the inveted effort, but the collaborative effort of at least four people from around the world directly led to a function that will serve all students and teachers using Sage. [Screenshot of one of Marshall Hampton's colorful swirling vector fields]

  Another example is the statistics functionality that (may?) be merged from the people in Europe that are using Sage to teach statistics??

- Notebook and LaTeX interfaces—we need to mention the LaTeX interface, since that is how we are going to be doing the textbooks, and what makes it so easy to incorporate Sage into traditional materials.
- The philosophy—here we mention the "genius of Sage" of building the car, plus we should mention that Python serves (literally) as the nuts-and-bolts of Sage, holding it all together, as well as the interface.
- Free (and open-source)—mention the restrictive licensing and fees associated with commercial systems. Also talk about how this lets us get into *diverse* institutions.

Computer algebra systems (CAS) have held great promise for education since their first appearance in the 1960s (Reduce, Macsyma) and accelerated with the introduction of Maple in the early 1980s and Mathematica in the late 1980s. Currently, CAS are widely found in the undergraduate classroom and a considerable amount of mathematics educations research has focused on the use of CAS in the learning of undergraduate mathematics (cite survey article by K. Heid or a more recent article). However, in many cases, computer algebra systems have driven the curriculum (cite Calculus & Mathematica project) and the result has not been the broad transformative effect on mathematics education that was envisioned. The closed development process creates a high barrier for teachers and students to alter or extend the software to meet curriculum needs. For example, external extensions to proprietary systems, such as libraries or packages, can extend proprietary systems, but must be purchased, distributed, and installed by every end-user. However an open development process, such as the open-source model that Sage follows, allows students and faculty to shape the core technology in a timely fashion to support the transformation of teaching and learning of mathematics on a broad scale. Open source software, such as Sage, allows the teaching and learning of mathematics to drive the technology, rather than the technology driving the teaching and learning.

The genius of Sage is the way it unifies over one hundred mature, best-of-breed, open source packages, each excelling at specific types of computations (e.g. MPIR for multi-precision integer and rational arithmetic, or IML for integer matrix computations) or complete libraries for specific areas of mathematics (e.g. GAP for group theory, FLINT for number theory, R for statistics) into a single open source system with a consistent interface (Mention SciPy, NumPy instead of some these packages - what are the most recognizable?). Sage is "building the car, not reinventing the wheel." This unification greatly simplifies the use of mathematical software for both students and teachers. Further, with an open-source license, Sage removes substantial barriers to classroom use of computer algebra systems such as purchase cost, restrictive licensing, expensive hardware, and limited availability on mobile devices. MENTION PYTHON HERE.[i] i

Sage's notebook interface is perhaps its most impressive feature, especially for use in education. Much of the power of the Sage notebook is a direct result of decisions to employ open software and open standards in its design. From within a standard web browser, a user can create a "worksheet" and interactively edit and execute new commands, with computations performed remotely on a server, and output returned to the worksheet. The interface makes it easy to view graphical output, annotate a worksheet, receive help on commands, browse the manuals, browse the source code, manage multiple worksheets, publish completed worksheets, or share and collaborate on worksheets. The Sage project hosts a public Sage server at `sagenb.org` which currently has over 20,000 accounts. So right now, anybody in the world with a network connection can productively harness the full power of Sage — even with minimal hardware or mobile devices, such as an under-powered desktop, a laptop, netbook, smartphone, or cell

[i] An open source approach to software for educational use, such as the approach Sage developers have taken, has many natural advantages. Sage has many impressive capabilities, a large number of which are a direct result of its adherence to the central philosophies of open source. For example, consider the choice of the open source language Python, one of the most popular scripting languages today, to unify the various parts of Sage. This choice then allows Sage to easily integrate the many impressive scientific packages written in Python. But more importantly, Python becomes the programming language for users within Sage, rather than a one-off language more typical of proprietary systems. A user with knowledge of Python is ready to be productive in Sage immediately, while a student new to programming receives a basic education in Python as a by-product of learning Sage — a skill that is readily transferable to a wide variety of applications in mathematics, science and engineering.

phone. Wide-open notebook servers continue to appear throughout the world [Hungary 2009/11/25, http://sage.math.u-szeged.hu/], while many more run behind campus firewalls for dedicated use.

Materials enhanced or created in our project will of course be distributed with open licenses and made freely available as part of Sage's website (XX visitors a month), or as part of Sage itself (over 6,000 downloads a month). But more importantly, the process for creating these materials will be made as easy as possible for authors, and workshops will be organized to train new authors and further refine the process of creating Sage-enhanced textbooks, guides and demonstrations.

```
Notebook
  - anywhere
  - server or local
  - need only web browser
  - annotation possible (latex)
  - AJAX/GMail analogy
  - student designed

Latex
  - acknowledged standard
  - used for open textbooks
  - jsMath
  - Sage notebook
  - a good example of open licenses, standards
  - and also a win for what we want to do

Curriculum Drives Development
  - not conversely
  - stress open development processes
  - illustrate with Jason/vector-field vignette
```

3. Implementation

Our goal is to provide a broad, positive influence on the teaching and learning of mathematics in the classroom by using open curricular materials, integrated with powerful and comprehensive open software, to realize a transformative effect on the teaching and learning of mathematics. Project Sage will revolve around three coordinated efforts: (1) creating Sage-enhanced curricular materials, (2) providing the Sage infrastructure to support these materials, and (3) working with and providing classroom support for teachers using these materials in the classroom for the first time. More specifically our proposal consists of five main activities.

- Convert existing open textbooks to Sage worksheets, and incorporate interactive demonstrations powered by Sage, along with guidance on how to use Sage to further understand the topic of the textbook.
- Create new curricular materials for standard undergraduate courses, especially for courses not covered by the textbooks we work to convert. These will include an organized collection of interactive demonstrations of high technical and pedagogical quality, plus subject-specific guides to introduce Sage to both faculty and students in the context of a particular topic.
- Improve the Sage library and its surrounding infrastructure (e.g., server design, notebook usability, collaboration tools) where the improvements have a direct and obvious benefit for the use of Sage in educational settings.
- Identify and work closely with fifteen (ten?) different undergraduate institutions to test, refine, improve and extend the work described just above. These institutions will be chosen to represent as diverse as possible an array of student populations and institutional profiles.

- These materials will all carry open licenses and be available through the Sage website, which has a global audience. Much, or all, of what we do will be incorporated into Sage itself. Sage Days workshops, MAA PREP workshops, and MAA minicourses will be utilized as opportunities to train faculty (outside of the fifteen test sites) in the use of these materials for their courses, and in the creation of new materials.

3.1. **Sage-enhanced open textbooks.**

3.1.1. *Imagine this.* A student is learning about row-reducing a matrix in a beginning linear algebra course. The electronic version of their textbook is an online Sage (We avoided mentioning Sage specifically below. Should we not mention Sage yet (and just say later that Sage most naturally fits the model)) worksheet they can view from anywhere in the world. Mathematical equations, in publication-quality formatting, describe the procedure. An interactive demonstration, embedded in the text where an example normally would be, allows the student to step through row-reducing a small nice matrix, generated on-the-fly. [MOCKUP SCREENSHOT] A second example lets the student choose the row operation to apply at each step of the reduction. In both cases, each operation is notated and illustrated in color in the matrix. Another example shows the student how to use a built-in command to row-reduce a matrix. With a single click, the student creates an empty code cell under the example to experiment with the command.

In class, the instructor does one simple example on the board. Then the instructor opens the online CAS and has the students guide the choice of row operations at each step for several example matrices of increasing complexity. At the end of class, the instructor clicks a button to publish the class work. After class, students view the worksheet as they review their notes. They can also, with a click, copy the worksheet into their notes, as well as annotate or change it.

The exercises include an interactive problem generator that creates matrices of student-specified sizes and complexity for the student to row-reduce. There is a “Solution” button which generates a step-by-step solution. A group exercise asks students to collaboratively write and test a short function to implement row-reduction using row operations. The student clicks a button to open a new Sage worksheet and types in a few lines using Sage matrices and commands from Python, the easy-to-use and industry-standard language upon which Sage is built. The student evaluates the code with a single click. Satisfied with their work, the student clicks another button, publishing the worksheet to the rest of the group for testing. Later that evening, the student checks the worksheet and sees that other students have tested the function, found a typo, and fixed it.

An advanced exercise guides a student through investigating numerical issues that arise in row reduction. By changing one parameter within a command, the matrices track numeric error bounds via interval arithmetic, and the student discovers that numerical errors can be a big issue for some matrices.

The next week, the instructor goes to the online server and looks over all of the published worksheets for the group projects. The instructor annotates each worksheet online, using an embedded word processor that supports mathematical typesetting.

The instructor writes a quiz over row-reduction. Inside of the LaTeX code, the instructor embeds a CAS command to make a simple matrix that serves as the answer to a quiz problem. The instructor then uses the CAS (from within his LaTeX document) to convert the answer matrix to a matrix for a routine quiz problem that involves only simple operations to row-reduce. The question matrix is automatically incorporated in the printed quiz, while a detailed solution is automatically incorporated in the answer key.

3.1.2. *Infrastructure for textbooks.* Many mathematicians and scientists use the LaTeX language to create technical articles and books. As a result, there is an impressive array of extensions to LaTeX. Sage also includes extensive support for LaTeX, through every mathematical object being *required* to have a LaTeX representation, and the decision to use jsMath to render mathematics beautifully in a Sage worksheet (within a standard web browser). Furthermore, the Sage notebook interface allows a user to add new text, including LaTeX code to display mathematics properly. This is another example of how open standards and open software serendipitouly combine to make powerful tools.

Our project proposes to convert open-source textbooks into folders of interactive Sage worksheets from existing mature open textbooks. The tex4ht translator (an NSF funded project) can convert extremely complex LaTeX into jsMath, which is then simple to convert to the Sage worksheet format since it relies heavily on HTML and jsMath. It is possible to insert Sage code into a LaTeX source file, and have it migrate to the eventual Sage worksheet as an input cell. The end result is a folder of Sage worksheets, each a mixture of text, high-quality typeset mathematics, Sage input cells and Sage interactive demonstrations.

3.1.3. *Targeted textbooks.* Subsection [**?**] lists many open textbooks that could be converted by our process. However, conversion is more that a technical process. We plan to incorporate interactive demonstrations, live Sage code, and guidance on the Sage library itself. So there is a significant care and thought required to integrate these uses of mathematical software properly.

These are our initial candidates for modeling how we can incorporate live code and interactive demonstrations to bring the full power mathematics software to the student *directly within their book*.

- William's number theory book.
- Rob's Linear algebra textbook. Do this first because (a) already has some Sage, (b) already has been designed to include "computational notes," (c) linear algebra in Sage is fairly mature, (d) I'm most familiar with the source for this one.
- Augment Tom's abstract algebra book, with Tom's assistance/involvement. I have a pile of stuff now on group theory, and ideas for new Sage functionality here. I expect to generate a lot for rings and fields Spring 2010 when I teach that material from Tom's book.
- Add Sage hints/experiments to Bogart's combinatorics problem book. Despite a GFDL license the source is not posted, and my one request has seen no answer. So I don't know how much work it will be to "clean-up" the source, etc. A former student, now on the faculty at Seattle U may be interested in being involved with this (I'm not suggesting adding him to the grant).

3.1.4. *Other open textbooks.* Beyond our initial candidates there is a range of books available that could be converted to cover portions of the standard undergraduate curriculum. Each of the textbooks below is licensed with a GNU Free Documentation Licenses (GFDL) or a Creative Commons License that allows anyone to distribute modified versions, typically only requiring attribution of the original author's work and the use of the same license.

- *A First Course in Linear Algebra*, Robert Beezer, `http://linear.pugetsound.edu/`. Second-year university level text, concentrating on understanding how to understand and formulate proofs. Used at thirteen schools since Fall 2007. Contains some Sage commands.
- *Abstract Algebra: Theory and Applications*, Tom Judson, `http://abstract.ups.edu/`. Standard upper-division treatment of groups, rings, domains, fields, Galois Theory. Published commercially in 1992, released as open source in 2007. Used by seven schools in its first year available. Supplement describes group theory in Sage.
- *Combinatorics Through Guided Discovery*, Kenneth Bogart, `http://www.math.dartmouth.edu/news-resources/electronic/kpbogart/`. Problem book in combinatorics with roughly 400 problems designed to teach the subject. NSF funded, released as open source.
- *Elementary Number Theory: Primes, Congruences, and Secrets*, William Stein, `http://wstein.org/ent/`. Published by Springer-Verlag, will be available with open license in Month Year????. Add capsule summary, Add extent of Sage usage (massive?)Open license consistent with paragraph above?
- *Vector Calculus*, Michael Corral, `http://www.mecmath.net/`. Standard treatment of multivariate calculus: vector calculus, partial derivatives, multiple integrals, theorems of Green and Stokes. Impressive graphics.
- *Trigonometry*, Michael Corral, `http://mecmath.net/trig/`. An in-depth, comprehensive and unified treatment of the typical high-school topics. Impressive graphics.
- *Cryptography*, David Kohel, `http://echidna.maths.usyd.edu.au/~kohel/tch/Crypto/crypto.pdf`. Classical ciphers and their cryptanalysis, modern stream ciphers and public-key cryptography. Significant appendices on Sage.

- Could we add my Riemann Hypothesis book with Barry Mazur? http://wstein.org/rh/ It's aimed at undergrads, but isn't a traditional textbook by any stretch.
- *Numerical Analysis*, Steven Pav, `http://scicomp.ucsd.edu/~spav/pub/numas.pdf` (course notes; small book length)
- (A possibility:) http://www.math.duke.edu/education/calculustext/

3.2. curricular materials from us, others.

- Create new subject-specific guides in traditional undergraduate subject areas to assist teachers with integrating the Sage into existing courses.
- Create an organized and searchable library of interactive online demonstrations for classroom use

3.3. Sage infrastructure work (code, servers, notebook).

- Server installs
- Plotting
- Windows porting
- Code fixups

3.4. Test Sites. The heart of Project Sage is getting Sage-enhanced materials in the hands of teachers and students to assist them in the teaching and learning of mathematics. We will work with selected faculty at test sites to provide support for using Sage in their classroom, for the Sage-enhanced materials that we have created, and teach them how to write their own materials. We will recruit two cohorts for our test sites with the first cohort beginning Summer 2011 and the second beginning the following summer. Several institutions have already expressed interest. [Note: We need to list some of these places or even attach letters of support expressing interest.] The teachers at our test sites will receive the following support.

- Teachers will commit to using Sage-enhanced materials in their classrooms for two years.
- Teachers will write and test a Sage-enhanced model for the class that their are teaching.
- Teachers will write and file a report at the end of each year of their two-year commitment.
- After the first year, teachers will train and conduct departmental workshops on Sage and how to use Sage-enhanced materials in the classroom. Teachers in the first cohort will also mentor teachers in the second cohort.

Project Sage will offer the following support for our teacher-participants.

- Teachers will receive a $5000 per year stipend for participating in Project Sage.
- Teachers will receive Sage support as well as support for Sage- enhanced materials and access to a Sage server.
- For each site a selected member of the project team will serve as liason to that site, which will include site visits for help and dissemination. Each member of the project team will serve in this role for two different sites.

3.5. training (Sage Days, PREP, minicourses).

4. Preparation, Qualifications

The five faculty in the Project Sage team together have XX years of experience teaching undergraduates at a wide range of institutions, four are active Sage developers (including its founder), three are authors of open textbooks, all have significant mathematics research experience, and one specializes in mathematics education research. Everyone is familiar with, and has extensive experience using, open source software and tools in their teaching and other professional activities. Besides the applicability of this experience to the project's central theme, they are also extremely adept at working collaboratively while still being separated geographically. Working together, they have the experience and skills to make this project a success.

The Project Sage leadership team will consist of the project PI, Co-PIs, Co-PI/Project Director, and selected faculty from the five universities.

4.1. **Personnel. Dr. William A. Stein**, Associate Professor of Mathematics at the University of Washington, is

Dr. Kiran S. Kedlaya, Associate Professor of Mathematics at the Massachusetts Institute of Technology, is

Dr. Jason Grout, Assistant Professor of Mathematics at Drake University, is

Dr. Robert Beezer, Professor of Mathematics at the University of Puget Sound, is a undergraduate teacher with 31 years of experience, an active researcher in algebraic graph theory, one of the first open textbook authors and a Sage developer. He began writing his open source linear algebra textbook in 2004 and has assisted Dr. Judson with the recent release of his very successful open-source abstract algebra text. He began using Sage in 2007 and began contributing code in early 2009. He will lead the technical process of converting textbooks from LaTeX to Sage worksheets, producing a simple system for other authors to use. The pedagogical implications of this new capability will be explored as he incorporates Sage into existing textbooks on topics he teaches frequently, such as linear algebra, abstract algebra, combinatorics, calculus and cryptography. He will continue to contribute code to Sage where the new functionality enables a more complete experience for an undergraduate student, and will suggest, review and test the project's improvements to interactive demonstrations and the notebook interface.

Dr. Thomas Judson, Associate Professor of Mathematics at Stephen F. Austin Sate University, is an active researcher in both mathematics and mathematics education. He is the author of an open-source undergraduate abstract algebra textbook, and will will work with Dr. Beezer to Sage-enhanced materials for abstract algebra. Dr. Judson has worked extensively with undergraduate mathematics teachers and has mentored graduate students in the teaching of mathematics. In addition to his teaching and research accomplishments, Dr. Judson brings added experience in working with diverse groups both in the U.S. and abroad. As such, he will work with Dr. Hassi to guide research and evaluation efforts for Project Sage. With experience in working with undergraduate faculty and graduate student mentoring, Dr. Judson will also provide insight and assistance in the implementation of Sage-enhanced materials in the undergraduate classroom. Project Sage efforts connected to research in mathematics education will be under the direction of Dr. Judson.

Tip to Maraj-Lisa, Sandra, etc. - cite evaluation section.

4.2. **Preparation.** Here we need to make the point that this is not an exploratory project (i.e., type I). Sage has been classroom-tested, NSF has given prior support for developing class. Rob has been using an open-source textbook with sage examples. Tom is an open-source author. Jason has used Sage in several different courses and shaped Sage capabilities as a result. We are at the institutionalizing stage now. Also mention the MAA PREP grant as evidence of adoption and institutionalization.

"A pilot project, a fourteen page primer on group theory designed to accompany Judson's abstract algebra text, clearly shows the feasibility of this process and clearly identifies the technical changes needed to make the process routine for an author."

4.3. **Prior and existing support.** Rob's sabbatical. William has administered several NSF grants.

5. Dissemination

We will broadly disseminate the results of Project Sage through a variety of forums. The Project Sage curricular materials will be accessible on the Sage website, `http://www.sagemath.org/`, where users can read documentation, make contributions to Sage, download Sage for free, and keep abreast of new developments in Sage. Sage Days, a series of conferences and workshops devoted to the development of Sage, are an ideal venue for dissemination. To date there have been eighteen Sage Days conferences, and these workshops now include sessions dedicated to using Sage in the classroom. We will to apply for special sessions and workshops on Sage in the classroom at the Joint Mathematics Meetings and MathFest, where we will be able to share the results of Project Sage with the undergraduate teaching community. We will also publish articles about the impact of using our materials on the teaching and learning of undergraduate mathematics. Drs. Beezer and Grout are part of a team delivering a workshop on how to use Sage in the classroom under the MAA's Professional Enhancement Program (PREP). We will apply to lead future PREP workshops that will incorporate the results of Project Sage.

6. Evaluation

The evaluation plan will measure the impact of Sage on undergraduate students, faculty, and the curriculum. More specifically, the plan will include methodologies for measuring

(1) the effectiveness of the project in developing Sage integrated curriculum materials such as Sage integrated with open source textbooks,
(2) the effectiveness of the use of Sage integrated materials within the secondary and undergraduate classroom.

The evaluation plan will also include a mechanism for tracking the implementation of Sage materials into the undergraduate and high school classroom by working with selected test sites. We will create an online tracking system to ensure that timely and accurate qualitative and quantitative data are collected for all test sites, including demographic data and statistics as well as service obligations.

Experience with previous, similar projects guides the evaluation plan for this effort. [Note: Add in the qualifications and experience of the evaluators.]

6.1. Formative Evaluation.

The formative component of the evaluation will focus on monitoring the quality of project activities, enabling the project to make mid-course corrections where needed and will address (1) the effectiveness of the project in developing Sage integrated materials. The summative component will focus on the impact of the project on the students and teachers at the test sites and will address (2) the effectiveness of the Sage-integrated materials in the undergraduate and high school classroom. We plan to gather information on the contributions of the project to the fields knowledge about future development of CAS integrated curricular materials.

The formative evaluation will be guided by the following questions:

(1) To what extent are Sage integrated materials and the support for these materials designed to the needs of the test sites.
(2) To what extent are the Sage integrated materials transportable/scalable to other undergraduate and high school classrooms?

The formative aspects of the evaluation will collect information on key aspects of the project to assess and provide feedback on the extent to which the project is designed and implemented to address important needs effectively. The key aspects of the project on which the evaluation will focus are: the need for, and resources to support, Sage enhanced materials in the classroom; and creation of materials that can be used more broadly in high school and university mathematics classrooms. Classroom needs and resource commitments will be documented through teacher surveys and interviews. The basis for selection of participating teachers and classrooms will be based on interest in the project and support from the teacher's institution.

6.2. Summative Evaluation.

The summative evaluation will be guided by the following questions:

(1) What is the impact of CAS integrated materials on teachers content knowledge, pedagogical content knowledge and classroom instructional practice?
(2) What is the impact of CAS integrated materials on in their own classrooms, campuses/districts on their students mathematics learning and participation in advanced-level mathematics courses?
(3) What is the ease of implementing CAS integrated materials on a wide scale basis?

To measure these changes Dr. Judson will collaborate with the project director and other members of the project and will consult with Marja-Liisa Hassi and Sandra Laursen of Ethnography & Evaluation Research (E&ER) at the Center to Advance Research and Teaching in the Social Sciences (CARTSS) at the University of Colorado in the selection of appropriate instruments to identify measures of for specific areas of the project. Drs. Hassi and Laursen have extensive experience evaluating other large mathematics and science education projects (NSF Awards # 0920126, # 0723600 # 0450088). [Note: I am not sure that these are the appropriate projects that we should mention, but I am assuming that Marja-Liisa will rewrite this part of the proposal.] To insure alignment to project goals, measures for the classroom will not be selected until the Sage-enhanced materials have been developed. The measures will be administered to participants at appropriate intervals throughout the project period.

Drs. Hassi and Judson and the project PIs will conduct site visits to partner institutions to observe and document the use of Sage-enhanced materials as the project progresses. Additional data will be collected through periodic surveys of all participants, and interviews of a sample. All surveys and interviews will be prepared and conducted by Drs. Hassi and Judson.

Drs. Hassi and Judson will collaborate with the appropriate institutions to collect demographic data on schools, teachers, and students within participating schools. They will analyze all data using appropriate statistical measures and will prepare one annual report per year documenting all evaluation activities and results and will provide formative evaluation feedback to the project PIs in presentations and correspondence timed to inform design decisions and mid-course corrections.

7. Budget Overview

- Total Salaries
- Total Stipends to test-Sites (travel money here?)
- Total travel, equipment
- Total package for inferastructure

8. Future Directions

more comprehensive approach, ground-up materials
Webwork, Moodle

End of OPEN-VERSION proposal

9. Project Description

Our project will create, implement, and evaluate a sustainable model for integrating computer algebra systems with traditional curricular materials for the undergraduate classroom. We will use open-source tools such as Sage and open curricular materials to accomplish this.

We will accomplish our goal by modifying existing open source materials and creating new curricular materials while continuing to develop Sage to directly support its use in undergraduate education. This project will leverage the inherent advantages of Sage as a free open-source computer algebra system (CAS) together with a growing list of open mathematics textbooks to deliver on the promise of CAS to significantly improve undergraduate mathematics education. We will also implement this model in diverse classrooms and institutions, refining and evaluating the model with an eye towards sustainability. More specifically, to build and evaluate a model for integrating CAS with traditional curricular materials in the classroom, we will:

- integrate Sage with existing mature textbooks and create other Sage-enhanced curricular materials, building a model and tools allowing others to easily do the same;
- partner with diverse institutions to test and refine this model and materials in a wide variety of courses while providing support for teachers;
- conduct formative and summative evaluation to determine the effectiveness of our model and materials and the impact on pedagogy and student learning.

9.1. **Imagine this.** A student is learning about row-reducing a matrix in a beginning linear algebra course. The electronic version of their textbook is an online Sage (We avoided mentioning Sage specifically below. Should we not mention Sage yet (and just say later that Sage most naturally fits the model)?) worksheet they can view from anywhere in the world. Mathematical equations, in publication-quality formatting, describe the procedure. An interactive demonstration, embedded in the text where an example normally would be, allows the student to step through row-reducing a small nice matrix, generated on-the-fly. A second example lets the student choose the row operation to apply at each step of the reduction. In both cases, each operation is notated and illustrated in color in the matrix. Another example shows the student how to use a built-in command to row-reduce a matrix. With a single click, the student creates an empty code cell under the example to experiment with the command.

In class, the instructor does one simple example on the board. Then the instructor opens the online CAS and has the students guide the choice of row operations at each step for several example matrices of increasing complexity. At the end of class, the instructor clicks a button to publish the class work. After class, students view the worksheet as they review their notes. They can also, with a click, copy the worksheet into their notes, as well as annotate or change it.

The exercises include an interactive problem generator that creates matrices of student-specified sizes and complexity for the student to row-reduce. There is a “Solution” button which generates a step-by-step solution. A group exercise asks students to collaboratively write and test a short function to implement row-reduction using row operations. The student clicks a button to open a new Sage worksheet and types in a few lines using Sage matrices and commands from Python, the easy-to-use and industry-standard language upon which Sage is built. The student evaluates the code with a single click. Satisfied with their work, the student clicks another button, publishing the worksheet to the rest of the group for testing. Later that evening, the student checks the worksheet and sees that other students have tested the function, found a typo, and fixed it.

An advanced exercise guides a student through investigating numerical issues that arise in row reduction. By changing one parameter within a command, the matrices track numeric error bounds via interval arithmetic, and the student discovers that numerical errors can be a big issue for some matrices.

The next week, the instructor goes to the online server and looks over all of the published worksheets for the group projects. The instructor annotates each worksheet online, using an embedded word processor that supports mathematical typesetting.

The instructor writes a quiz over row-reduction. Inside of the LaTeX code, the instructor embeds a CAS command to make a simple matrix that serves as the answer to a quiz problem. The instructor then uses the CAS (from within his LaTeX document) to convert the answer matrix to a matrix for a routine quiz problem that involves only simple operations to row-reduce. The question matrix is automatically incorporated in the printed quiz, while a detailed solution is automatically incorporated in the answer key.

9.2. **Previous approaches: Why Sage?** Sage is the most natural computer algebra system to realize the vision outlined above. The open development model, the notebook and LaTeX interfaces, the underlying language and philosophy behind how Sage is built, and the free nature all contribute toward this.

- Open development model - Like mathematics, the Sage model for development is driven by collaboration and openness (is this a controversial statement about mathematics?). Just as research topics drive the direction of pure mathematics, the curriculum and needs of the users directly drives the development. As an example of how teaching has driven the development, one of the investigators on this grant was asked one day by a student if Sage had a 3D vector field plotting function. He quickly wrote one. Encouraged by the open community that is common among open-source projects such as Sage, he posted his code online in the ticket-tracking website. Another developer that had written most of the 3d graphing in Sage suggested a small change that greatly increased the efficiency of the function. Several months later, another person (tsarnold) authored a very similar function which had slightly improved options and published their code on the public Sage notebook server. In Fall 2008, the investigator needed the 3d vector field plotting function for his multivariable class. He made a few changes to the code based on suggestions, and posted it for his class to use. In a closed proprietary system, that's where things likely would have stopped. Since that time, several other deep changes to the internals of Sage have been made to simplify code in functions like the 3d plotting, some of which were directly the result of feedback about the function. The investigator recently taught the same class again and needed the function again. He incorporated the good ideas from the various sources into his function and posted a documented, efficient version of the function to the trac ticket and asked for review. Another Sage developer reviewed the function immediately, with the comment "Very nice, positive review. This is great timing since I am about to teach vector fields in a week or two." The function was incorporated into Sage and released with the next version two weeks later. Now not just the original students benefit from the time, but the collaborative effort of at least 4 people from around the world directly led to a function that will serve all students and teachers using Sage.

  Another example is the statistics functionality that (may?) be merged from the people in Europe that are using Sage to teach statistics??
- Notebook and LaTeX interfaces—we need to mention the LaTeX interface, since that is how we are going to be doing the textbooks, and what makes it so easy to incorporate Sage into traditional materials.
- The philosophy—here we mention the "genius of Sage" of building the car, plus we should mention that Python serves (literally) as the nuts-and-bolts of Sage, holding it all together, as well as the interface.
- Free (and open-source)—mention the restrictive licensing and fees associated with commercial systems. Also talk about how this lets us get into *diverse* institutions.

Computer algebra systems (CAS) have held great promise for education since their first appearance in the 1960s (Reduce, Macsyma) and accelerated with the introduction of Maple in the early 1980s and Mathematica in the late 1980s. Currently, CAS are widely found in the undergraduate classroom and a considerable amount of mathematics educations research has focused on the use of CAS in the learning of undergraduate mathematics (cite survey article by K. Heid or a more recent article). However, in many cases, computer algebra systems have driven the curriculum (cite Calculus & Mathematica project) and the result has not been the broad transformative effect on mathematics education that was envisioned. The closed development process creates a high barrier for teachers and students to alter or extend the software to meet curriculum needs. For example, external extensions to proprietary systems, such as libraries or packages, can extend proprietary systems, but must be purchased, distributed, and installed by every end-user. However an open development process, such as the open-source model that Sage follows, allows students and faculty to shape the core technology in a timely fashion to support the transformation of teaching and learning of mathematics on a broad scale. Open source software, such as Sage, allows the teaching and learning of mathematics to drive the technology, rather than the technology driving the teaching and learning.

The genius of Sage is the way it unifies over one hundred mature, best-of-breed, open source packages, each excelling at specific types of computations (e.g. MPIR for multi-precision integer and rational arithmetic, or IML for integer matrix computations) or complete libraries for specific areas of mathematics (e.g. GAP for group theory, FLINT for number theory, R for statistics) into a single open source system with a consistent interface (Mention SciPy, NumPy instead of some these packages - what are the most recognizable?). Sage is "building the car, not reinventing the wheel." This unification greatly simplifies the use of mathematical software for both students and teachers. Further, with an open-source license, Sage removes substantial barriers to classroom use of computer algebra systems such as purchase cost, restrictive licensing, expensive hardware, and limited availability on mobile devices. MENTION PYTHON HERE.[ii] ii

Sage's notebook interface is perhaps its most impressive feature, especially for use in education. Much of the power of the Sage notebook is a direct result of decisions to employ open software and open standards in its design. From within a standard web browser, a user can create a "worksheet" and interactively edit and execute new commands, with computations performed remotely on a server, and output returned to the worksheet. The interface makes it easy to view graphical output, annotate a worksheet, receive help on commands, browse the manuals, browse the source code, manage multiple worksheets, publish completed worksheets, or share and collaborate on worksheets. The Sage project hosts a public Sage server at `sagenb.org` which currently has over 20,000 accounts. So right now, anybody in the world with a network connection can productively harness the full power of Sage — even with minimal hardware or mobile devices, such as an under-powered desktop, a laptop, netbook, smartphone, or cell phone. Wide-open notebook servers continue to appear throughout the world [Hungary 2009/11/25, http://sage.math.u-szeged.hu/], while many more run behind campus firewalls for dedicated use.

Materials enhanced or created in our project will of course be distributed with open licenses and made freely available as part of Sage's website (XX visitors a month), or as part of Sage itself (over 6,000 downloads a month). But more importantly, the process for creating these materials will be made as easy as possible for authors, and workshops will be organized to train new authors and further refine the process of creating Sage-enhanced textbooks, guides and demonstrations.

10. Implementation

One of our main goals is to provide a broad, positive influence on the teaching and learning of mathematics in the classroom with CAS-enhanced materials. The implementation of Project Sage will consist of three coordinated efforts: (1) creating Sage-enhanced curricular materials, (2) providing the Sage infrastructure to support these materials, and (3) working with and providing classroom support for teachers using these materials in the classroom for the first time.

10.1. **Creating curricular materials.**

- Convert selected open source mathematics textbooks to Sage-enhanced electronic versions, modeling how incorporating live code and interactive demonstrations can bring the full power of a computer algebra systems to the student *from within the book.*

[ii] An open source approach to software for educational use, such as the approach Sage developers have taken, has many natural advantages. Sage has many impressive capabilities, a large number of which are a direct result of its adherence to the central philosophies of open source. For example, consider the choice of the open source language Python, one of the most popular scripting languages today, to unify the various parts of Sage. This choice then allows Sage to easily integrate the many impressive scientific packages written in Python. But more importantly, Python becomes the programming language for users within Sage, rather than a one-off language more typical of proprietary systems. A user with knowledge of Python is ready to be productive in Sage immediately, while a student new to programming receives a basic education in Python as a by-product of learning Sage — a skill that is readily transferable to a wide variety of applications in mathematics, science and engineering.

William William's number theory book.

Rob Rob's Linear algebra textbook. Do this first because (a) already has some Sage, (b) already has been designed to include "computational notes," (c) linear algebra in Sage is fairly mature, (d) I'm most familiar with the source for this one.

Rob Augment Tom's abstract algebra book, with Tom's assistance/involvement. I have a pile of stuff now on group theory, and ideas for new Sage functionality here. I expect to generate a lot for rings and fields Spring 2010 when I teach that material from Tom's book.

Rob Add Sage hints/experiments to Bogart's combinatorics problem book. Despite a GFDL license the source is not posted, and my one request has seen no answer. So I don't know how much work it will be to "clean-up" the source, etc. A former student, now on the faculty at Seattle U may be interested in being involved with this (I'm not suggesting adding him to the grant).

- Create new subject-specific guides in traditional undergraduate subject areas to assist teachers with integrating the Sage into existing courses.
- Create an organized and searchable library of interactive online demonstrations for classroom use

10.1.1. *Infrastructure for textbooks.* Describe the system for converting textbooks to Sage; everyone will be able to do it!

Many mathematicians and scientists use the LaTeX language to create technical articles and books. As a result, there is an impressive array of extensions to LaTeX. Sage also includes extensive support for LaTeX, through every mathematical object having a LaTeX representation, and the use of jsMath to render mathematics beautifully in a Sage worksheet (within a standard web browser). Furthermore, the Sage notebook interface allows a user to add new text, including LaTeX code for mathematics.

This project proposes to convert open-source textbooks into folders of interactive Sage worksheets, or to create Sage-enhanced supplements for existing open-source textbooks. A Sage worksheet is a combination of input and output cells, optionally with text (HTML) inbetween. The tex4ht translator (an NSF funded project) can convert extremely complex LaTeX into jsMath, which can be converted to the Sage worsheet format with extremely minimal modifications. It is possible to insert Sage code into a LaTeX source file, and have it migrate to the eventual Sage worksheet as an input cell. A pilot project, a fourteen page primer on group theory designed to accompany Judson's abstract algebra text, clearly shows the feasibility of this process and clearly identifies the technical changes needed to make the process routine for an author.

The end result is a folder of Sage worksheets, each a mixture of text, high-quality typeset mathematics, Sage input cells and Sage interactive demonstrations.

Also, the plan for bundling a collection of worksheets into a more high-level document (basically a packaged version of the Sphinx documentation).

10.1.2. *Converting specific textbooks.* A list of specific textbooks we will do.

Each of the textbooks below is licensed with a GNU Free Documentation Licenses (GFDL) or a Creative Commons License that allows anyone to distribute modified versions, typically only requiring attribution of the original author's work and the use of the same license.

- *A First Course in Linear Algebra*, Robert Beezer, `http://linear.pugetsound.edu/`. Second-year university level text, concentrating on understanding how to understand and formulate proofs. Used at thirteen schools since Fall 2007. Contains some Sage commands.
- *Abstract Algebra: Theory and Applications*, Tom Judson, `http://abstract.ups.edu/`. Standard upper-division treatment of groups, rings, domains, fields, Galois Theory. Published commercially in 1992, released as open source in 2007. Used by seven schools in its first year available. Supplement describes group theory in Sage.
- *Combinatorics Through Guided Discovery*, Kenneth Bogart, `http://www.math.dartmouth.edu/news-resources/electronic/kpbogart/`. Problem book in combinatorics with roughly 400 problems designed to teach the subject. NSF funded, released as open source.
- *Elementary Number Theory: Primes, Congruences, and Secrets*, William Stein, `http://wstein.org/ent/`. Published by Springer-Verlag, will be available with open license in Month Year????.

Add capsule summary, Add extent of Sage usage (massive?)Open license consistent with paragraph above?

- *Vector Calculus*, Michael Corral, `http://www.mecmath.net/`. Standard treatment of multivariate calculus: vector calculus, partial derivatives, multiple integrals, theorems of Green and Stokes. Impressive graphics.
- *Trigonometry*, Michael Corral, `http://mecmath.net/trig/`. An in-depth, comprehensive and unified treatment of the typical high-school topics. Impressive graphics.
- *Cryptography*, David Kohel, `http://echidna.maths.usyd.edu.au/~kohel/tch/Crypto/crypto.pdf`. Classical ciphers and their cryptanalysis, modern stream ciphers and public-key cryptography. Significant appendices on Sage.
- Could we add my Riemann Hypothesis book with Barry Mazur? http://wstein.org/rh/ It's aimed at undergrads, but isn't a traditional textbook by any stretch.
- *Numerical Analysis*, Steven Pav, `http://scicomp.ucsd.edu/~spav/pub/numas.pdf` (course notes; small book length)
- (A possibility:) http://www.math.duke.edu/education/calculustext/

10.1.3. *Other curricula material.* Other curricula material that we will convert; subject-specific tutorials, course notes, etc.

We propose to create subject specific tutorials to answer questions such as "What are the Sage commands you need to know in order to do multivariate calculus?" We propose to create a systematic and organized collection of subject specific tutorials and interacts to include with Sage. Some work has been put forth in this (library of interacts, Georgia Sage Days Primers). But there needs to be a lot more work and infrastructure in place to distribute these with Sage. Maybe reference the Demonstrations project?

- [Jason and William] Iron out the infrastructure issues with Rob's book, and whatever writing we do on our own subject-specific tutorials. o Adapt notebook to collections of worksheets + Grouping worksheets (folders?) + Format for posting/moving (ie zipped, tar'ed) o Cross-worksheet links + Must be portable, ie stable/functional after a move + Then needs support in tex4ht o Define new Latex environments for... + Inline sage code + Live Sage code blocks (w/, w/out corresponding output) + Interacts + Coordinate this with Dan Drake's SageTeX o Create a tex4ht mode to support above + Goal: run tex4ht on Latex souce, output is zipped-up worksheets + tex4ht now has a project page, Karl Berry is a TeX expert +
  http://gna.org/projects/tex4ht/ o Make a tex4ht spkg for others to use easily
- Subject-specific tutorials
  *
  Have a Sage EduDays (maybe at the very beginning of summer) where we invite several people to do projects over the summer to be class-tested after summer 2, before summer 3.
  * Have another Sage EduDays where we invite some people back from the summer 2 EduDays to help teach a larger group about writing subject-specific curricula using Sage. In the classroom * Survey people about barriers to classroom use
  * Sign up a few people to use Sage in the classroom. List a few names, and have a mechanism to advertise for more people. Give people a stipend to write a report of using Sage in a class for a semester. We will offer them technical support.

10.2. **Sage in the Classroom.** The heart of Project Sage is getting Sage-enhanced materials in the hands of teachers and students to assist them in the teaching and learning of mathematics. We will work with selected faculty at test sites to provide support for using Sage in their classroom, for the Sage-enhanced materials that we have created, and teach them how to write their own materials. We will recruit two cohorts for our test sites with the first cohort beginning Summer 2011 and the second beginning the following summer. Several institutions have already expressed interest. [Note: We need to list some of these places or even attach letters of support expressing interest.] The teachers at our test sites will receive the following support.

- Teachers must commit to using Sage-enhanced materials in their classrooms for two years.
- Teachers must write and test a Sage-enhanced model for the class that their are teaching.

- Teachers must write and file a report at the end of each year of their two-year commitment.
- After the first year, teachers must train and conduct departmental workshops on Sage and how to use Sage-enhanced materials in the classroom. Teachers in the first cohort will also mentor teachers in the second cohort.

In return, Project Sage will offer the following support for our teacher-participants.

- Teachers will receive a $5000 per year stipend for participating in Project Sage.
- Teachers will receive Sage support as well as support for Sage- enhanced materials and access to a Sage server.

10.3. **Sage infrastructure.**

- Extend and improve Sage's capabilities in relevant subject areas to facilitate student learning and ease-of-use.
- Improve Sage's infrastructure in areas directly related to undergraduate education, such as teacher-student interaction, collaboration and organization of work, and the framework for quickly and easily building interactive demonstrations.

```
* Target subject areas: linear algebra, undergraduate abstract algebra, discrete math, graphic
      o

        Full-scale use of nthiery's TestSuite system to make sure that things are consistent a
      o Evaluate the intuitiveness and discoverability of the interfaces. Make sure the docume
      o Go through a book in each of these areas and make sure that you can do exercises throu
* Notebook features:
      o Ways for students and professors to organize their work (like tags on the the workshee
      o Ways for students and professors to interact with each other (e.g., submitting and gra
            + A grading and annotating system for grading student worksheets.
      o interactive javascript input widgets (hopefully we can do something with mathjax!)
      o flexible layout of interact controls and output
* Have a Sage Days for working on all of this sometime during the summer.

* Make any needed changes to the code/goals from Summer 1, based on feedback from classroom us
* Make it easy to set up a campus/personal sage server
      o Document the hardware requirements for various scenarios
      o Make the virtualbox image very polished, including graphical buttons to create and sta
      o Carefully document any things that people need to understand about setting up the netw
      o

        Explore the option of a cloud instance (like an Amazon EC2 instance) that instructors
      o Build the infrastructure for a library of curricula materials. This includes a library
      o

        Have another Sage Days focused on these goals (separate from the EduDays focused on wr


* Make any needed changes to the code/goals from Summer 1 and Summer 2, based on feedback from
* Have another Sage Days focused on these goals
*

  Have a Sage Days for writing more CCLI grants or other education grants! Maybe a
  Sage Days with the Moodle and WebWork people to talk about Sage integration
  plans and grants for that.
```

11. Evaluation

The evaluation plan will measure the impact of Sage on undergraduate students, faculty, and the curriculum. More specifically, the plan will include methodologies for measuring

(1) the effectiveness of the project in developing Sage integrated curriculum materials such as Sage integrated with open source textbooks,
(2) the effectiveness of the use of Sage integrated materials within the secondary and undergraduate classroom.

The evaluation plan will also include a mechanism for tracking the implementation of Sage materials into the undergraduate and high school classroom by working with selected test sites. We will create an online tracking system to ensure that timely and accurate qualitative and quantitative data are collected for all test sites, including demographic data and statistics as well as service obligations.

Experience with previous, similar projects guides the evaluation plan for this effort. [Note: Add in the qualifications and experience of the evaluators.]

11.1. Formative Evaluation.

The formative component of the evaluation will focus on monitoring the quality of project activities, enabling the project to make mid-course corrections where needed and will address (1) the effectiveness of the project in developing Sage integrated materials. The summative component will focus on the impact of the project on the students and teachers at the test sites and will address (2) the effectiveness of the Sage-integrated materials in the undergraduate and high school classroom. We plan to gather information on the contributions of the project to the fields knowledge about future development of CAS integrated curricular materials.

The formative evaluation will be guided by the following questions:

(1) To what extent are Sage integrated materials and the support for these materials designed to the needs of the test sites.
(2) To what extent are the Sage integrated materials transportable/scalable to other undergraduate and high school classrooms?

The formative aspects of the evaluation will collect information on key aspects of the project to assess and provide feedback on the extent to which the project is designed and implemented to address important needs effectively. The key aspects of the project on which the evaluation will focus are: the need for, and resources to support, Sage enhanced materials in the classroom; and creation of materials that can be used more broadly in high school and university mathematics classrooms. Classroom needs and resource commitments will be documented through teacher surveys and interviews. The basis for selection of participating teachers and classrooms will be based on interest in the project and support from the teacher's institution.

11.2. Summative Evaluation.

The summative evaluation will be guided by the following questions:

(1) What is the impact of CAS integrated materials on teachers content knowledge, pedagogical content knowledge and classroom instructional practice?
(2) What is the impact of CAS integrated materials on in their own classrooms, campuses/districts on their students mathematics learning and participation in advanced-level mathematics courses?
(3) What is the ease of implementing CAS integrated materials on a wide scale basis?

To measure these changes Dr. Judson will collaborate with the project director and other members of the project and will consult with Marja-Liisa Hassi and Sandra Laursen of Ethnography & Evaluation Research (E&ER) at the Center to Advance Research and Teaching in the Social Sciences (CARTSS) at the University of Colorado in the selection of appropriate instruments to identify measures of for specific areas of the project. Drs. Hassi and Laursen have extensive experience evaluating other large mathematics and science education projects (NSF Awards # 0920126, # 0723600 # 0450088). [Note: I am not sure that these are the appropriate projects that we should mention, but I am assuming that Marja-Liisa will rewrite this part of the proposal.] To insure alignment to project goals, measures for the classroom will not be selected until the Sage-enhanced materials have been developed. The measures will be administered to participants at appropriate intervals throughout the project period.

Drs. Hassi and Judson and the project PIs will conduct site visits to partner institutions to observe and document the use of Sage-enhanced materials as the project progresses. Additional data will be collected through periodic surveys of all participants, and interviews of a sample. All surveys and interviews will be prepared and conducted by Drs. Hassi and Judson.

Drs. Hassi and Judson will collaborate with the appropriate institutions to collect demographic data on schools, teachers, and students within participating schools. They will analyze all data using appropriate statistical measures and will prepare one annual report per year documenting all evaluation activities and results and will provide formative evaluation feedback to the project PIs in presentations and correspondence timed to inform design decisions and mid-course corrections.

12. Dissemination

We will broadly disseminate the results of Project Sage through a variety of forums. The Project Sage curricular materials will be accessible on the Sage website, `http://www.sagemath.org/`, where users can read documentation, make contributions to Sage, download Sage for free, and keep abreast of new developments in Sage. Sage Days, series of conferences and workshops devoted to the development of Sage, are an ideal venue for dissemination. To date there have been eighteen Sage Days conferences, and there is now time dedicated to using Sage in the classroom. We are now plan to apply for special sessions and workshops on Sage in the classroom at the Joint Mathematics Meetings and MathFest, where we will be able to share the results of Project Sage with the undergraduate teaching community. We will also publish articles about the impact of using CAS-integrated materials on the teaching and learning of undergraduate mathematics. Drs. Beezer and Grout are part of a team delivering a workshop on how to use Sage in the classroom under the MAA's Professional Enhancement Program (PREP). We plan to apply for future PREP workshops that will incorporate the results of Project Sage.

13. Experience and results from prior NSF support

Here we need to make the point that this is not an exploratory project (i.e., type I). Sage has been classroom-tested, NSF has given prior support for developing class. Rob has been using an open-source textbook with sage examples. Tom is an open-source author. Jason has used Sage in several different courses and shaped Sage capabilities as a result. We are at the institutionalizing stage now. Also mention the MAA PREP grant as evidence of adoption and institutionalization. William has had several NSF grants.

14. Project Management/Governance Plan

The Project Sage leadership team will consist of the project PI, Co-PIs, Co-PI/Project Director, and selected faculty from the five universities.

Dr. William A. Stein, Associate Professor of Mathematics at the University of Washington, is

Dr. Kiran S. Kedlaya, Associate Professor of Mathematics at the Massachusetts Institute of Technology, is

Dr. Jason Grout, Assistant Professor of Mathematics at Drake University, is

Dr. Robert Beezer, Professor of Mathematics at the University of Puget Sound, is

Dr. Thomas Judson, Associate Professor of Mathematics at Stephen F. Austin Sate University, is an active researcher in both mathematics and mathematics education. He is the author of an open-source undergraduate abstract algebra textbook, and will will work with Dr. Beezer to Sage-enhanced materials for abstract algebra. Dr. Judson has worked extensively with undergraduate mathematics teachers and has mentored graduate students in the teaching of mathematics. In addition to his teaching and research accomplishments, Dr. Judson brings added experience in working with diverse groups both in the U.S. and abroad. As such, he will work with Dr. Hassi to guide research and evaluation efforts for Project Sage. With experience in working with undergraduate faculty and graduate student mentoring, Dr. Judson will also provide insight and assistance in the implementation of Sage-enhanced materials in the undergraduate classroom. Project Sage efforts connected to research in mathematics education will be under the direction of Dr. Judson.

15. Future directions

16. End of new grant

End of where we are at.

17. How sage allows for wide adoption while maintaining the benefits of previous approaches

Sage is a cohesive system that includes nearly one hundred pieces of open-source mathematics software. Creating this distribution has required the combined work of over two hundred developers, including faculty, industry professionals and students at the graduate, undergraduate, and high school level.

William Stein started the Sage project at Harvard in 2005. Now there are over 200 Sage developers. In late 2009, there were well over 5,000 downloads of Sage and over 75,000 visits to the `http://sagemath.org` website. The free `http://sagenb.org` online Sage server has about 18,000 accounts from faculty, students, the general public all over the world.

The goal of the Sage project is to create viable easy-to-use free open source mathematical software to support both education and research in mathematics, engineering and the sciences. Success would have a genuinely transformational impact on the mathematical sciences, broadly impacting people involved with mathematics at all levels. It would have the potential to have a profound impact on how mathematics is taught and done for decades to come.

Sage can be used to study general and advanced, pure and applied mathematics. This includes a huge range of mathematics, including algebra, calculus, elementary to very advanced number theory, cryptography, numerical computation, commutative algebra, group theory, combinatorics, graph theory, exact linear algebra and much more. Sage combines and integrates numerous open-source software packages seamlessly into a common experience. It is well suited for education, studying and research. The notebook interface to Sage runs in a web-browser. Inside a Sage notebook, you can create embedded graphics, beautifully typeset mathematical expressions, add and delete input, and share your work across the network. A command-line interface is available for yet another style of interaction with the program.

Do we need something prior suppo other appli we are maki year?

17.1. **Student driven interface.** The Sage notebook interface has attracted many student developers. It originated at UCLA (is this right??, more detail, Boothby, who else?). Intense development continues today, most recently a major upgrade was designed and constructed Fall 2009. As an example of student involvement, this recent upgrade relied on significant contributions from Tim Dumol, a high school student in the Phillipines. With so much student input to its design, the resulting interface is one very familiar to today's student. It is an AJAX/Web-2.0/cloud-computing (pick a buzzword) style application, taking advantage of numerous Javascript libraries to turn a web browser into a powerful application, similar to GMail or Google Wave (better comparisons?). This makes it possible for thne notebook to be platform-independent and run well on minimal hardware like netbooks and smartphones. Presumably it would run unchanged (and with impressive performance) in Google's upcoming Chrome OS. (Is this a place to talk about similar projects like CodeNode, Code Mirror, Python SPD, etc?)

17.2. **Technology independent.** Sage's fundamental adherence to open software and open standards, combined with a modular approach, makes it highly independent of specific choices of hardware and operating systems (need to mention Windows support here as project goal?). The software compiles and runs on a wide variety of hardware and software (many flavors of Linux, say top 3,4,5 distros, Mac, Sun, 32-bit/64-bit, Atom chip for netbooks). (Mention commercial systems that have dropped Sun support?) High standards for code review and demanding test suites ensure that Sage installs easily in such a wide variety of situations. Indeed many users install all of Sage as a way to guarantee that a single desired open source package will install on their system. At ahigher level, a modular approach to combining mathematical packages means that as new, faster or more capable packages appear, it is relatively easy for Sage to transition to these new packages with no obvious negative changes to the typical user. (Example: Did mpmath replace something?)

17.3. **Features and how we are addressing the disadvantages.**

- Mathematical deficiences. Sage has a solid foundation of mathematical objects based on category theory. For example, when asking for the nullspace of a matrix, you actually get back a vector space object. You can then do things like intersect that vector space with another vector space, ask if a vector is in the vector space, look at the homomorphisms from the vector space into other objects, etc. [[category theory = sounds scary, abstract, and like "Axiom"? instead could we just say that our core design builds on a more modern (i.e. last 60 years – bourbaki) views of mathematical objects instead of ignoring them. it's striking that mathematics in the world of maple and mathematica is really 19th century, whereas sage is 20th century. We could acknowledge Magma and Scratchpad/Axiom and MuPAD.]]

- Costs Sage is freely available, and people are encouraged to make copies and share Sage with other people. There are multiple public Sage servers, on which anyone can register an account and use Sage, including `http://sagenb.org`, run by William Stein, and supported by the NSF (SCREMS grant DMS-).
- In addition to not costing any money, the Sage source code is freely available and people are encouraged to look at it and improve it, if they wish.
- Sage uses the mainstream Python programming language. This language is one of the most popular programming languages, and has been used in industry in key applications (e.g., Google, Star Wars, the hubble telescope, etc.). Many universities are shifting beginning programming courses to using python. Python is lauded for being a very easy-to-learn, powerful language. Many students have exposure to python before using Sage (meaning that the learning barrier is drastically reduced), and the python skills that are developed in learning Sage are then applicable in areas far beyond mathematics or their course work. Since python is an industry standard, there is a huge amount of existing code readily available to students for non-mathematical tasks that come up when analyzing data and exploring mathematical ideas. For example, ... [[Also see the TIOBE rankings of popularity which make Python the number 2 or 3 interpreted language: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html]]

Mention something about the huge number of lightning talks or the scipy conferences, talking about applications of python in science.

18. The Vision

18.1. **Vignettes.** Vigenttes from the point of view of students (in class and out of class), instructor (wanting to add something small to a course or wanting to go big time, or teaching a more advanced course, oradapting existing material, or....?

Ideas:

- (To replace the Taylor polynomial example below. Too long?) Imagine the section of a linear algebra textbook teaching row operations and the procedure for bringing a matrix to reduced row-echelon form. An interactive demonstration in the text would generate a small matrix requiring fairly simple row operations and a "next" button would step through the row-reducing procedure. After each step, the original matrix and the modified matrix could be displayed with the affected entries color-coded and text would describe the row operation employed (using properly formatted mathematical symbols). Once a student was comfortable with the procedure, inputs could allow the student to bypass the simplistic "next" button and decide at each step exactly which row operation to use. Once comfortable with the method, a short code stanza would illustrate how to use Sage's command to convert a matrix to its echelon form. As live Sage code, the student could edit this to examine many other examples.

  In the exercises, a student could be asked to demonstrate their knowledge of the procedure by writing an actual Python procedure to row-reduce a Sage matrix over the rational numbers. It would be a simple matter of clicking just below the problem to open up a Sage input cell to enter the code, and a click on the evaluate link to test executing it. The student could concentrate on the mechanics of the procedure since a Sage matrix type with rational entries would avoid the choice of a data structure, while Sage's exact arithmetic for rational numbers would avoid numerical problems like overflow and rounding. A problem generator in the exercises could generate a seemingly infinite supply of "random" matrices for row-reduction, with varying sizes and varying complexity (as controlled by the student), yet perhaps with fairly "easy" row operations. More applied exercises could lead to nontrivial systems of equations that could be simply solved in two lines by constructing the right augmented matrix and then calling Sage's command for row-reduction.

It is important to realize that everything just described can be created and executed using only Sage and is available to the student in their text. No Java applets are needed for the interactive demonstration or problem generator, no compiler or programming environment is needed for the programming exercise. The seemingly random problems would be generated by a Sage command (which is an example of code we would add to Sage to support these types of educational applications). Everything happens in the electronic copy of the book, in a standard web browser, and what the student creates can remain there as a permanent part of their copy of the book.

- Taylor polynomials – (But we aren't converting a calculus textbook! Should we use an example from a textbook we are actually converting?) – Imagine a calculus lesson on Taylor polynomials as a Sage worksheet in a folder that comprises a calculus textbook. Sage input cells would instruct the reader on the relevant Sage syntax, and provide the reader the opportunity to edit the input to experiment with different functions and different degree polynomials. An included interactive Sage demonstration could use a slider to control the degree of the polynomial, and an input box would allow the student to specify an input where the function and the poynomial could be compared numerically. On each change to the two inputs the demonstration would automatically respond with a new plot of the function and the polynomial, indicating visually the location of the input value on the two curves, along with the two numerical values at the input, plus the numerical difference between the values. By opening up the mini-word-processor built into Sage, a student can annotate a copy of their text alongside the demonstration, recording their observations or questions, using LaTeX code to create the mathematical expressions accurately in their notes.
- High school teacher accessing an interact from within Sage for their classroom – Some sort of vignette talking about high school is important because CCLI emphasizes preparing K-12 teachers. If we show we are empowering K-12 teachers, that is a great thing we should point out. Make sure to say because we are free, high schools have an extra incentive to use it.
- Students collaborating, working from home and school together, maybe in an advanced undergraduate course – Maybe on an algebra project that requires stuff in Sage that only Sage and Magma have.
- Teacher converting notes or modifying the textbook (maybe at a small liberal arts school)

18.2. **Implementation of proven methods.**

18.3. **Measurable outcomes.** list of specific, measurable outcomes

- A system for easily producing online textbooks with live Sage examples from standard Latex source files
- Online textbooks with live Sage examples covering a variety of undergraduate subjects, including linear algebra, abstract algebra, and number theory
- A number of smaller tutorials and course notes for specific topics, such as calculus, discrete math, differential equations, etc.
- Classroom-ready Sage Appliance Virtual machine, so instructors can easily set up a class or campus Sage server
- Translated versions of various appendices of books that contain code for commercial math software systems.
- Improvement of Sage for undergraduate classroom instruction

18.4. **Textbooks and other curricula material (Before the classroom).**

18.4.1. *Infrastructure for textbooks.* Describe the system for converting textbooks to Sage; everyone will be able to do it!

Many mathematicians and scientists use the LaTeX language to create technical articles and books. As a result, there is an impressive array of extensions to LaTeX. Sage also includes extensive support for LaTeX, through every mathematical object having a LaTeX representation, and the use of jsMath to render mathematics beautifully in a Sage worksheet (within a standard web browser). Furthermore, the Sage notebook interface allows a user to add new text, including LaTeX code for mathematics.

This project proposes to convert open-source textbooks into folders of interactive Sage worksheets, or to create Sage-enhanced supplements for existing open-source textbooks. A Sage worksheet is a combination of input and output cells, optionally with text (HTML) inbetween. The tex4ht translator (an NSF funded project) can convert extremely complex LaTeX into jsMath, which can be converted to the Sage worsheet format with extremely minimal modifications. It is possible to insert Sage code into a LaTeX source file, and have it migrate to the eventual Sage worksheet as an input cell. A pilot project, a fourteen page primer on group theory designed to accompany Judson's abstract algebra text, clearly shows the feasibility of this process and clearly identifies the technical changes needed to make the process routine for an author.

The end result is a folder of Sage worksheets, each a mixture of text, high-quality typeset mathematics, Sage input cells and Sage interactive demonstrations.

Also, the plan for bundling a collection of worksheets into a more high-level document (basically a packaged version of the Sphinx documentation).

18.4.2. *Converting specific textbooks.* A list of specific textbooks we will do.

Each of the textbooks below is licensed with a GNU Free Documentation Licenses (GFDL) or a Creative Commons License that allows anyone to distribute modified versions, typically only requiring attribution of the original author's work and the use of the same license.

- *A First Course in Linear Algebra, Robert Beezer,* `http://linear.pugetsound.edu/`*. Second-year university level text, concentrating on understanding how to understand and formulate proofs. Used at thirteen schools since Fall 2007. Contains some Sage commands.*
- Abstract Algebra: Theory and Applications, Tom Judson, `http://abstract.ups.edu/`. Standard upper-division treatment of groups, rings, domains, fields, Galois Theory. Published commercially in 1992, released as open source in 2007. Used by seven schools in its first year available. Supplement describes group theory in Sage.
- *Combinatorics Through Guided Discovery, Kenneth Bogart,* `http://www.math.dartmouth.edu/news-resources/electronic/kpbogart/`*. Problem book in combinatorics with roughly 400 problems designed to teach the subject. NSF funded, released as open source.*
- Elementary Number Theory: Primes, Congruences, and Secrets, William Stein, `http://wstein.org/ent/`. Published by Springer-Verlag, will be available with open license in Month Year????. Add capsule summary, Add extent of Sage usage (massive?)Open license consistent with paragraph above?
- *Vector Calculus, Michael Corral,* `http://www.mecmath.net/`*. Standard treatment of multivariate calculus: vector calculus, partial derivatives, multiple integrals, theorems of Green and Stokes. Impressive graphics.*
- Trigonometry, Michael Corral, `http://mecmath.net/trig/`. An in-depth, comprehensive and unified treatment of the typical high-school topics. Impressive graphics.
- *Cryptography, David Kohel,* `http://echidna.maths.usyd.edu.au/~kohel/tch/Crypto/crypto.pdf`*. Classical ciphers and their cryptanalysis, modern stream ciphers and public-key cryptography. Significant appendices on Sage.*
- *Could we add my Riemann Hypothesis book with Barry Mazur? http://wstein.org/rh/ It's aimed at undergrads, but isn't a traditional textbook by any stretch.*
- Numerical Analysis, Steven Pav, `http://scicomp.ucsd.edu/~spav/pub/numas.pdf` (course notes; small book length)
- (A possibility:) http://www.math.duke.edu/education/calculustext/

18.4.3. *Other curricula material.* Other curricula material that we will convert; subject-specific tutorials, course notes, etc.

We propose to create subject specific tutorials to answer questions such as "What are the Sage commands you need to know in order to do multivariate calculus?" We propose to create a systematic and organized collection of subject specific tutorials and interacts to include with Sage. Some work has been put forth in this (library of interacts, Georgia Sage Days Primers). But there needs to be a lot more work and infrastructure in place to distribute these with Sage. Maybe reference the Demonstrations project?

18.4.4. *Faculty workshops for converting and developing new "learning materials and strategies".* We will get lots of people to convert stuff and teach with Sage.

18.5. **Using Sage in the classroom - "Developing faculty expertise".** Mini-grants for faculty to write curriculum materials (textbooks/course notes/tutorials/interacts) using Sage. Maybe a few thousand dollars each? A summer salary, or a one-course buyout for a semester?

Several successful conversions or supplements will demonstrate the range of possibilities and allow the necessary experience to design an automated conversion and production system. This will allow other authors to easily parlay just a working knowledge of LaTeX to the ability to create folders of interactive Sage enhanced worksheets. Workshops and grants can help and encourage other authors to learn the technical skills, and the possibilities afforded by this new approach, while creating or converting their own textbooks or supplements.

This could be accomplished through

- A workshop sponsored by AIM.
- A SageDays devoted to textbook conversion and supplement creation.
- A mini-course at the Joint Meetings or MathFest.
- Mini-grants for people to convert textbooks and course notes to Sage
- Mini-grants for people to write course materials for Sage, such as interactive demonstrations ("interacts").

18.5.1. *Classroom trial program (formative evaluation) (feedback from faculty //and// students).*

18.6. **Research on how faculty adopt this aproach.**

19. Dissemination

19.1. **Ease of adoption.**

19.2. **Archive of refereed modules(?)** (I'm not sure if this paragraph should go above or here. for now, it's in two places.)

We propose to create subject specific tutorials to answer questions such as "What are the Sage commands you need to know in order to do multivariate calculus?" We propose to create a systematic and organized collection of subject specific tutorials and interacts to include with Sage. Some work has been put forth in this (library of interacts, Georgia Sage Days Primers). But there needs to be a lot more work and infrastructure in place to distribute these with Sage. Maybe reference the Demonstrations project?

19.3. **AIM open source textbook initiative.**

20. Enhancing Sage for specific courses

In this section we describe Sage development projects that will enhance the effectiveness of Sage in the classroom. This involves one-time work to unify the systax for those functions that are commonly used in a particular course. During this project we will make these changes for linear algebra, abstract algebra, calculus, differential equations, number theory, and combinatorics. (say a bit more about how this will benefit all subsequent use of Sage in the classroom)

Also, we emphasize that undergraduates will contribute directly to these projects, and this has a double impact, since we are bringing as many students in as possible to do development *while* we are carrying out our project, instead of waiting until the end.

20.0.1. *Linear Algebra.* Go through linear algebra; make interfaces consistent (left/right dichotomy, matrix decomposition syntax) and implement anything else an undergrad course in linear algebra would need

RAB: I have a list of syntax fixups I noted when I built the linear algebra quick reference. The quickref has been translated to Korean, should that be a nice example of the worldwide reach of Sage.

Finally fix echelon/hermite form over ZZ/QQ

20.0.2. *Abstract Algebra.*

- additions to "named" groups, to round out list of all small groups
- quotient groups as actual sets of cosets
- all subgroups, all normal subgroups (normal may be implemented already)
- groups defined by presentations (wrap GAP support?)
- perhaps massive optimizations with libGAP, which greatly decreases latency (by a factor of 2000).
- RAB: I'll teach rings, fields, etc with Sage in the spring for the first time. I'm assuming irreducible polynomials over QQ and field extensions (towers) are well-implemented? Galois groups too? I'll try to look this over.

20.0.3. *Graph Theory.*

- Rob Miller's fast C graph backends
- the linear programming patches and code that depends on them
- traveling salesman problem

20.0.4. *plotting.*

- Make 2d graphics have transformations
- Expose matplotlib better
- improve matplotlib (for example, see how great the mma contour plot function is)
- html5 canvas backend for matplotlib
- uniform color interface
- explore options for interactive 3d plotting again
- uniform mesh and region-plotting interfaces

20.0.5. *Calculus.*

- differentiation notation
- clean up symbolics
- make a maxima library and switch to using the library interface
- get rid of maxima asking questions during integration

20.0.6. *Number theory.*

20.0.7. *Differential Equations.*

20.1. **Notebook development, allowing easy grading and organizing, etc. ("Student-focused work", even being done by students!)** The Sage notebook is an AJAX application, like Gmail or Google Maps. It provides an interactive web-based worksheet in which one can enter arbitrary Sage commands, see beautifully typeset output, create 2-D and 3-D graphics, publish worksheets, and collaborate with other users.

screenshot

"Professors at dozens of universities around the world are getting excited about how they can leverage the Sage notebook in their teaching.

"With some colleagues in our University (Lyon, France) we have built a project around Sage for undergraduate students… *And the University has decided to support this project.* Good news.

"Now we will be facing the problem to build a Sage configuration which will work for say 200 students at the same time (students will use the notebook), and prepare professors for Sage teaching. There are 'some' technical problems to solve…" (T. Dumont, the Sage mailing list.)

Stein has been working fulltime during Fall 2009 funded by University of Washington on vastly improving the robustness and scalability of the notebook. This works is coming at the right time, because use of the Sage notebook is growing rapidly. Using hardware purchased using an NSF SCREMS grant (DMS-...), Stein created a free public Sage notebook server at `http://sagenb.org`.

(check this) Typically, in Fall 2009, we have seen several hundred new accounts created on the public Sage server every day.

Several specific goals for notebook development include:

- Improve the notebook so that it will robustly handle up to 1000 simultaneous users when running on a single high-end server, as demonstrated by a robust automated test suite
- Implement management tools so administrators can manage the notebook load and better balance resources
- Create easy-to-use research tools for educators, so they can understand how their students use Sage

20.2. **Making Sage more classroom-ready.**

20.3. **Making Sage servers easy to set up and administer.** Create a nice Virtual Box Sage notebook server appliance with a graphical interface, two virtual machines that provides a rock solid secure Sage notebook server setup. Users will be able to very easily install this appliance on Windows, Linux, OS X, and Solaris (x86) servers. The management interface will provide clear and easy documentation about setting up this server, creating new notebook servers for specific classes or instructors, starting and stopping notebook servers, seeing resource usage, adding users and authentication frameworks (such as LDAP), and upgrading Sage with minimal user downtime.

We intend to provide support to get Sage to be used on several new campuses (this helps fulfill the "project is institutionalized" goal of type II proposals). We will create a new mailing list for server administration issues, and in addition to the help we provide would like to hire a knowledgeable employee for a certain number of hours per day to answer support questions.

In summary, we propose to provide high-quality enterprise level software infrastructure and support to educators. What we propose is far from free or easy it is in fact expensive. But if NSF bears the cost of a few hundred thousand dollars now, then literally millions of students will benefit with no cost to them. This is an excellent investment by NSF.

20.4. **Growing (an already started!) community ("sustainability").**

20.4.1. *Infrastructure and community already in place.* Mention sage days, workshops, joint sessions, vibrant mailing list, chat, lots of talks around the world, etc. Maybe mention a growing enthusiasm in education and Sage, but most activities center on the research side, which is why we need this grant.

3 pages!

The is *critical* to spend lots of time thinking and planning about. Most people don't, and it makes a huge impact on whether their grant gets funded.

In relationship to broader impact, include a plan about your dissemination plan. Specifically, where are you going to go, what are you talking about, why they'll accept you, so:

- put website into math dl
- PREP proposals
- workshops to help authors retrofit open source textbooks with Sage, or help open textbook authors initiate new books, with the goal being similar interactive Sage-enhanced textbooks
- workshop proposals that may be included in the budget.
- Get seven people that commit to using sage as we go along and are paid honorariums to write tutorials telling how to integrate Sage into their courses.

21. References

The list of references here.