S2I2 Exploratory Workshop:
Open Source Software as a Foundation for Scientific Research

1 Objectives

The goal of this workshop is to discuss the viability and potential impact of a Scientific Software Innovation Institute (S2I2) supporting the use of open source software for scientific research. The PIs would use their unique experience as leaders in the Sage and SciPy projects to put together a workshop and corresponding report that would address many questions relevant to such an institute.

Stein founded the NSF-funded Sage software project in 2005 (`http://www.sagemath.org`). The main goal of the Sage project is to create a viable free open source alternative to Magma, Maple, Mathematica, and MATLAB. An important inspiration for Sage was a short workshop at NSF in 2003, run by Brian Conrey, on the future of computers in mathematical research.

Millman is active in the scientific Python community. He serves on the steering committee for both NumPy (`http://numpy.org`) and SciPy (`http://scipy.org`), which are the two fundamental libraries for numerical and scientific computing in Python. In addition to organizing numerous workshops and sprints, he has organized the last three SciPy conferences in the US as well as the first two SciPy conferences in India. He is also one of the founders and developers for the neuroimaging in Python project (`http://nipy.org`).

> "I think we need a symbolic standard to make computer manipulations easier to document and verify. And with all due respect to the free market, perhaps we should not be dependent on commercial software here. An open source project could, perhaps, find better answers to the obvious problems such as availability, bugs, backward compatibility, platform independence, standard libraries, etc. One can learn from the success of TEX and more specialized software like Macaulay2. I do hope that funding agencies are looking into this."
>
> – Andrei Okounkov (see [1]).

1.1 Intellectual Merit and Broader Impact

The **intellectual merit** of the report that comes out of this workshop is that it promises to provide a snapshot of the core issues of sustainability, peer review, and reproducibility, which are becoming a vital concern in all areas of the mathematical sciences. The report will provide a deep and unique perspective drawing on the extensive experience of the PIs.

In the long run, this workshop could lead to the creation of an institute, which would develop free open source software infrastructure, peer review models, and reproducible research methodologies. This institute has the potential to dramatically change the tools used by all students, researchers, scientists, and engineers. Thus this one workshop could potentially have a **broad impact** on nearly everybody involved in any way in the mathematical sciences and engineering.

In the short term, the main **broader impact** would be that workshop participants will be more aware of sustainability and reproducibility issues in computational mathematics. Also, several graduate students will be involved in creation of the report and preliminary surveys, which will expand their understanding of computation in mathematics. Moreover, the workshop report and

data will be made widely available, which could result in raised awareness of these issues by the mathematical sciences communities.

1.2 The Deliverable

We will write a report addressing the creation of an S2I2, identifies tools and techniques that have been successful in community software projects, and explores challenges in using supercomputing infrastructure to tackle research problems.

Work on this report would begin immediately with various background data gathering activities carried out by a few small working groups, followed by a 2-day meeting in late July. In August, the results of the meeting and data would be refined and organized, and a final report would be presented to the NSF in September 2010.

1.3 Identifying Target Communities and Foci

The workshop will include discussions about which communities and foci provide the best opportunities for an S2I2. As background, before the workshop we will form working groups that will make a public wiki containing a list of areas of mathematics (based on the AMS subject classification) and areas of scientific computing. The working groups for each area will summarize the status of relevant software, algorithms, and communities.

Example 1.1. For computation with function fields, we might find that only the proprietary computer algebra system Magma implements many of the important algorithms, and that there is a vibrant group of mathematicians working on algorithms in this area, motivated by challenge problems in arithmetic geometry and cryptography.

Example 1.2. For machine learning, we might find that there are several open source packages, such as Weka and Orange, that cater to different communities and offer slightly different features.

The Sage project has already created surveys for several areas of mathematics, and this workshop would provide an opportunity for more of these. A rough draft will be completed before the workshop, and one of the activities of the workshop will be to discuss the draft, and form recommendations about the most useful mathematical and scientific areas on which to focus our efforts.

1.3.1 Survey Methodology

We will carry out surveys during June and early July 2010, which we will analyze at the workshop. The survey will consist of a list of key open- ended questions. Questions may include:

- *What workshop topics would compel you two attend?* Example responses: "A workshop on practical algorithms and software for computing normal forms of integer matrices in the context of algebraic topology, group theory, and number theory" or "A workshop on efficient algorithms and software for signal and imaging processing in the context of the biological sciences."
- *Which algorithms are crucial to your research and why?* Example responses: "Hess's algorithm for computing Riemann-Roch spaces, which is critical to understanding the arithmetic of curves and constructing optimal error correcting codes" or "the F4 algorithm for computing Groebner basis, which plays a key role in developing block ciphers in cryptography."

- *Who are the current and potential leaders in practical applications and development of software in your research area (please include graduate students)?*

1.4 Identifying Tools, Techniques, and Processes

Another goal of this proposal is to identify and describe concrete tools and techniques that can provide a foundation for sustainable, high-quality software infrastructure. There are many tools, such as source code management systems, code review systems, and automated testing systems, that have been developed to support software engineering. Given the software we hope to build, support, and sustain, some tools are more appropriate and effective than others. We will look at existing projects, including Sage, NumPy, Python, OpenOffice, Firefox, R, Macaulay2, Scilab, etc., and identify the role of:

- Programming languages,
- Code peer review processes and systems,
- Automated regression and unit testing software,
- Specific bug trackers, and
- Mailing lists, discussion groups, and chat rooms.

Background work identifying these tools will be carried out by working groups before the workshop, and the results will be discussed at the workshop. The report will contain tables examining which software engineering techniques are used by mathematical and scientific software communities and which are not. This may follow the format and methodologies that Steve Easterbrook has recently introduced in studying software engineering in the climate change community.

1.5 Personnel

An S2I2 would host a mix of short-term and long-term visitors as well as technical and administrative personnel. During the workshop we will explore possible organizational structures. Finding the right personnel will be essential to providing continuity to the software projects supported by the institute. Among the questions we will raise are:

Question 1.3. *What kind of personnel should an S2I2 have?* Example responses: "It is essential for the S2I2 to have a few full-time technical software developers employed for many years"; or "It is more important to hire short-term personnel over full-time employees (e.g., graduate students during summer session, faculty on sabbatical)."

Question 1.4. *Should the S2I2 software engineers be drawn from industry or academia?* Example response: "It is better to hire scientists and mathematicians with experience in software development over professional software engineers with some background in science and engineering."

1.6 Sustainability

We will also discuss the long-term sustainability of open source software and communities supported by an S2I2. If the institute itself is necessary for long-term success, how will the institute sustain itself in the long run?

Question 1.5. *How should the institute seek long-term sustainability?*

Question 1.6. *What are the trade-offs between a not-for-profit institute and a for-profit company?*

Question 1.7. *Should the institute sell books and training?*

Question 1.8. *Should an institute pursue fund raising (e.g., donations)?*

Question 1.9. *What are the positive and negative aspects of other revenue streams?* For example, one source of income could be maintaining Sage Notebook servers for hundreds of universities and schools.

1.7 Leveraging Cyberinfrastructure

In what ways could an institute produce software that best leverages national and international cyberinfrastructure? For example, one of the PIs (Stein) recently ran a tutorial workshop on Sage at the Scientific Software Day at the Texas Advanced Computing Center (see `http://www.tacc.utexas.edu/softwareday/`), where he learned about some of the challenges research scientists experience when using resources such as the NSF-funded TeraGrid. We will invite people from this community to identify and understand some of the problems they face:

- Running massively parallel computations on supercomputers.
- Combining code in different languages, including Fortran, MATLAB, C/C++, and Python.
- Dealing with software licensing and copyright issues in the context of supercomputing, where traditional commercial licensing may not be affordable.
- How to manage and document evolving software, which encodes sophisticated models.
- How to easily and efficiently create user interfaces for exploring large data sets.
- How to overcome the reluctance of some mathematicians and scientists to utilize the TeraGrid resources available to them to attack grand challenge problems in their research areas? Is this more an issue of awareness, lack of necessary training, technical skills, or appropriate software?

2 Transformative Impacts

In this section, we discuss some of the potential transformative impacts on both mathematical and computational research in science and engineering, which could come out of this workshop.

2.1 Impacts on Mathematical Research

Questions of correctness, reproducibility, and scientific value arise when building mathematical research on top of proprietary software. There are published refereed papers containing results that rely on computations performed in Magma, Maple, or Mathematica, including several by one of the PIs (Stein). In some cases, a specific version of Magma is the only software that can carry out the computation.

Question 2.1. What can an S2I2 do to ensure that computational mathematical results can be shared and reproduced [2]?

Thousands of papers rely on results computed by commercial software using unpublished algorithms available in only that software and nowhere else. For example, Magma, which has a relatively small user base, notes that "We are currently aware of approximately 3,000 publications about research in which Magma or Cayley have played a role" (see `http://magma.maths.usyd.`

edu.au/magma/citations/). To maintain their competitive advantage, some algorithms in proprietary mathematical software are not published. Some vendors go so far as to blatantly argue against exposing the internals of their software:

> "Indeed, in almost all practical uses of Mathematica, issues about how Mathematica works inside turn out to be largely irrelevant. You might think that knowing how Mathematica works inside would be necessary [...]" (See [3].)

At the workshop, we will explore to what extent the current dominant use of proprietary software to support research is viable and sustainable. Concerns may include:

- Tens of millions of lines of source code are kept secret in proprietary mathematical software. Should we be concerned that mathematical research is being built on top of this foundation? Or, are mathematical algorithms described well enough in the literature that independent implementations are practical, and the use of proprietary software will never lead to a crisis in the foundations of mathematical research?
- Older versions of software are often not available, so it may be impossible to run computations described in a paper. What problems does this present to readers?
- It can be problematic for researchers at different institutions with different software to collaborate if one institute only has access to Maple and the other has only Mathematica. What limitations on scientific research are imposed by the use of proprietary software?

Answers to the above questions will undoubtedly vary depending on subject area and researcher. In order to obtain a well-rounded perspective on the above questions, we will ensure heavy users of proprietary software are invited. For example, one of the PIs (Stein) recently had a discussion with David Farmer, a frequent Mathematica user, about his computations with Maass forms. Evidently, several researchers have developed completely independent closed code bases in different languages for doing computation with Maass forms. Due to the subtle numerical issues involved in these computations, Farmer explains that this has been good for research on Maass forms, since if several different codes produce the same answer, then one has greater confidence in the result.

We finish this section with two anecdotes.

2.1.1 MuPAD-Combinat

The MuPAD-Combinat project, which was started by Florent Hivert and Nicolas M. Thiéry in 2000, built the world's preeminent system for algebraic combinatorics on top of MuPAD. In 2008, MuPAD was purchased by MathWorks (makers of MATLAB), so MuPAD is no longer available as a separate product, and now costs $3000 (commercial) or $700 (academic). As a result, the MuPAD-Combinat group has spent several years reimplementing their code as part of Sage.

The MuPAD-Combinat group was not taken by surprise by the failure of MuPAD, but instead was concerned from the beginning by the inherent risk in building their research program on top of a commercial plaform. In fact, they decided to switch to Sage two months before the bad news hit, and have since made tremendous progress porting their code. This work has received substantial funding from the NSF as an FRG (DMS-0652641, DMS-0652652, DMS-0652668, DMS-0652648):

> "The FRG paid in particular for my 18 months of sabbatical at Davis, which was critical in the switch from MuPAD to Sage. [...] It has been such a relief during the last two years not to have this Damocles sword on our head!"
>
> – Nicolas M. Thiéry, personal communication.

2.1.2 F4: Fast Computation of Groebner Basis

Consider the F4 algorithm for computing Groebner basis. Magma and Maple each have their own closed "secret" implementations of this algorithm, and their implementations do extremely well on practical problems. For the last 10 years, many people have tried valiantly to produce open source implementations that are competitive in general and nobody has yet succeeded.

There are a number of other similar algorithms, whose implementation takes substantial focused effort and research, since the actual algorithms themselves are not published. Could an institute, with several-month long programs for creating sustainable first-rate implementations of these algorithms, lay the foundations for future generations of researchers, and also educate more people about how these algorithms work?

2.1.3 Transformation?

Browsing a listing of new papers at `http://arxiv.org`, one is struck by how many rely on mathematical software; this reliance is only likely to grow. Could an institute that fosters open source mathematical software transform how mathematical research is conducted? Could it refine, popularize, and teach the tools, structures, and development processes needed so that we, our graduate students, and their students, can create sustainable software infrastructure to support mathematical research? Or is such education and transformation better done in other ways?

2.2 Impacts on Science and Engineering

We would discuss during the workshop the extent to which an S2I2 would impact the use of computational methods and software in science and engineering. Over the last several years, the PIs have witnessed an increasing trend for research scientists and engineers to work in a high-level interpreted environment (such as Python or MATLAB) to do exploratory interactive work. The institute could provide a coordinating facility to ensure that the latest algorithms and methods are available in a system that everyone can utilize and benefit from. Statisticians have done something similar using the open source statistical package R as the de facto standard for publishing statistical algorithms and methods.

Question 2.2. What is the best way to train scientists in software engineering practices and methodologies? What kind of training activities should an institute promote to ensure the long-term viability of community developed software?

Question 2.3. What role should an S2I2 play in promoting reproducible research? What kind of software tools or methods have the greatest impact and promise?

In summary, as scientific research grows increasingly dependent on computing, it may be critical that our computational resources are developed with the same rigor, open review, and access as the results they support. An S2I2 institute has the potential to help promote:

- Sharing of scientific software, data, and knowledge necessary for reproducible research,
- Unrestricted access to research outcomes and educational tools,
- Open source software,
- Academic recognition of computational work on equal footing with the publication of results,
- Tested, validated, and documented software as the basis for reliable scientific outcomes, and

- High standards of computational literacy in the education of mathematicians, scientists, and engineers.

This workshop will provide feedback on the above suggested goals, along with others suggested by participants. For example, which of the above goals is most important to researchers and why?

3 Statement of Need

There is a strong interest in high quality, open source, community developed software for mathematical and scientific research. Computation is playing a significant role in many areas of mathematical research, science, and engineering. The long-term vitality of this research may be enhanced by open source community developed software. An institute would provide a home base for some of these projects, and this workshop will provide a way to assess such needs.

4 Related Events

In this section, we list Sage and SciPy workshops during the last few years. The PIs have been heavily involved in these workshops and will draw on relevant connections in recruiting people for the proposed S2I2 workshop.

4.1 Sage Days Workshops

Sage Days workshops are gatherings in which a few dozen undergraduates, graduate students, postdocs, professors, and others come together for about 5 days and passionately design and code algorithms that improve Sage. These events are highly relevant to the present proposal, because they provide a tried and tested model for the sort of workshops that could take place as part of future S2I2-funded activities. The workshops involve notable research and educational activities. For example, at Sage Days 19 at the Clay Mathematics Institute in Dec. 2009 (see `http://wiki.sagemath.org/dayscambridge2`), Jennifer Balakrishnan completed the first ever verification of something called Kolyvagin's conjecture for a rank 3 elliptic curve, whilst Barry Mazur and Kiran Kedlaya made some initial forays into anabelian geometry using Sage, and Karl-Dieter Crisman organized a day of talks devoted to the use of Sage in college teaching.

The Sage community has grown dramatically over the last four years as a result of over two dozen workshops in which the development and use of Sage has played a central role. The workshops listed below have primarily involved topics in number theory, arithmetic geometry, computer algebra, large-scale fixing of bugs, and algebraic combinatorics. Stein organized or co-organized most of these workshops. The frequency of these workshops has grown: *we anticipate that there will be at least 12 such workshops during 2010.* Considering that the number of Sage Days per year is nearly half the number of workshops of an institute such as AIM or MSRI, the Sage Days workshop series may be thought of as a **virtual institute** that has grown up around the Sage project.

- *Sage Days 1:* Feb. 2006 at UC San Diego.
- *Summer Graduate Workshop on Computing with Modular Forms:* July 2006 at MSRI.
- *Sage Days 2:* Oct. 2006 in Seattle, WA.
- *Interactive Parallel Computation in Support of Research in Algebra, Geometry and Number Theory:* Feb. 2007, MSRI.

- *Sage Days 3:* Feb. 2007 at IPAM (UCLA).
- *Sage Days 4:* June 2007 in Seattle, WA.
- *Sage Days 5: Computational Arithmetic Geometry,* Oct. 2007, CMI, Boston.
- *Sage Days 6:* Nov. 2007, Heilbronn Institute, Bristol, UK.
- *Sage Days 7:* Feb. 2008, IPAM (UCLA).
- *Sage Days 8: Number Theory and High Performance Computation*, Mar. 2008, Austin.
- *Sage Days 8.5: Developer Coding Days*, June 2008 in Seattle, WA.
- *Sage Days 9: Mathematical graphics and visualization*, Aug. 2009, SFU, Vancouver.
- *Sage Days 11: Special functions and computational number theory meet scientific computing*, Nov. 2008 in Austin, TX.
- *Sage Days 12: Bug Smash*, Jan. 2009 in San Diego, CA.
- *Sage Days 13: Quadratic Forms and Lattices*, March 2009 in Athens, Georgia.
- *Sage Days 14: Sage and Macaulay2 for Algebraic Geometry*, Mar. 2009, MSRI.
- *Sage Days 15: Developer Days*, May 2009 in Seattle, WA.
- *Sage Days 16: Computational Number Theory*, June 2009 in Barcelona, Spain.
- *Sage Days 17: Computing with Modular forms and L-functions*, Sep. 2009, Lopez Island.
- *Sage Days 18: Computations related to the BSD Conjecture*, Dec. 2009, CMI, Boston.
- *Sage Days 19: Second Sage Bug Smash*, January 2010 in Seattle, WA.
- *Sage Days 20: Combinatorics*, Feb. 2010, Marseille, France.
- *Sage Days 20.5: Algebraic Combinatorics, Rep. Theory*, May 2010, Fields Institute.
- *Sage Days 21: Function fields*, May 2010 in Seattle, WA.
- *Sage Days 22: MSRI Graduate Student Workshop on Elliptic Curves*, June 2010, MSRI.
- *Sage Days 23: Number theory and computer algebra*, July, 2010, Leiden, Netherlands.
- *Sage Days 23.5: Singular and Sage*, July, 2010, Kaiserslautern, Germany.
- *Sage Days 24: Symbolic computation*, July, 2010 at RISC in Linz, Austria.
- *Sage Days 25: Numerical computation*, August, 2010 in Bombay, India.

4.2 SciPy Events

The SciPy community has had numerous conferences, coding sprints, and developer meetings over the last several years. There are annual conferences in the US, Europe, and India, which bring together scientists, mathematicians, and programmers from academia and industry. Three years ago, Millman founded and edited the peer-reviewed conference proceedings for the US conference.

While the conferences typically have 100–200 attendees, the sprints and developer meetings typically range from 10–20 participants. The sprints are similar to Sage Days (described above), but typically focus on the NumPy and SciPy libraries. The developer meetings are rarer, but are useful when to decide on a major strategic change in code. Millman has organized or co-organized most of the SciPy events:

- *2002 SciPy Conference*, August 2002 at Caltech in Pasadena, CA
- *2003 SciPy Conference*, August 2003 at Caltech in Pasadena, CA
- *2004 SciPy Conference*, August 2004 at Caltech in Pasadena, CA

- *Future directions for SciPy meeting*, March 2005 at UC Berkeley
- *2005 SciPy Conference*, August 2005 at Caltech in Pasadena, CA
- *2006 SciPy Conference*, August 2006 at Caltech in Pasadena, CA
- *2007 SciPy Conference*, August 2007 at Caltech in Pasadena, CA
- *2007 SciPy Sprint*, August 2007 at Caltech in Pasadena, CA
- *2007 European SciPy Conference*,
- *SciPy Sprint*, December 2007 at UC Berkeley
- *SciPy Sprint*, March 2008 in Austin, TX (joint with Sage Days 8).
- *SciPy Sprint*, March 2008 in Paris, France (joint with NIPY and IPython sprint).
- *SciPy Sprint*, April 2008 at UC Berkeley
- *SciPy Sprint*, July 2008 in Austin, TX (focus on MayaVi)
- *2008 SciPy Conference*, August 2008 at Caltech in Pasadena, CA
- *2008 European SciPy Conference*,
- *2009 SciPy Conference*, August 2009 at Caltech in Pasadena, CA
- *2009 European SciPy Conference*, 2009
- *2009 SciPy India Conference*, December 2009 in Kerala, India
- *2010 SciPy Conference*, June 2010 in Austin, TX
- *2010 European SciPy Conference*, July 2010 in Paris, France
- *2010 SciPy India Conference*, December 2009 in Hyderabad, India

5 Organizers

In this section, we list the names of the chairperson and members of the organizing committee and their organizational affiliation.

1. **William Stein (chair)**, Department of Mathematics, University of Washington
2. **Fernando Perez**, Neuroscience Institute, University of California, Berkeley
3. **Jarrod Millman**, Neuroscience Institute, University of California, Berkeley
4. **Victoria Stodden**, Statistics, Stanford University

6 Location and Announcements

6.1 Location

The workshop will take place on Friday, July 30th and Saturday, July 31st. Given the dates, it would be optimal to have the workshop in Berkeley or Seattle. If it has to be in the DC area, then we will have it in the DC area.

6.2 Announcements

The meeting will be announced on the main Sage mailing list, on several SciPy-related mailing lists, the R mailing list, and any other relevant lists. We will also announce the workshop (and

get feedback on the proposed questions) at several conferences the PIs are already organizing this summer, including: Sage Days 22: Computing with Elliptic Curves at the MSRI in UC Berkeley; the 9th annual SciPy Conference in Austin; Sage Days 23: Number Theory in Leiden, the Netherlands; the 3rd annual Euro SciPy Conference in Paris, France; Sage Days 23.5: Singular and Sage in Kaiserslautern, Germany (July 14-16, 2010); and Sage Days 24: Differential Algebra, Special Functions in Linz, Austria (July 17-22, 2010).

7 Organizational Plan

The meeting will be conducted over two days. The first day will be mostly focused on brainstorming and structured around a series of short presentations followed by discussion. The second day will be focused directly around drafting the workshop report. We will expect full participation by all workshop attendees on the first day. A smaller set of attendees will be asked to participate in the morning session of the second day and only the organizers will be expected to stay through the that afternoon, though everyone will be encouraged to stay for the entire workshop.

7.1 Day 1: Morning Session

The meeting will begin with an hour of introductory remarks from Stein and Millman setting out the general objectives of the workshop. These remarks will be immediately followed by an hour of discussion focused around these objectives. After a short break, we will have four speakers who will each deliver a brief 10 minute presentation followed immediately by a related discussion. These presentations will be solicited from our invited attendees prior to the meeting and will focus on *Tools, Techniques, and Processes* (see Section 1.4 and 1.7).

7.2 Day 1: Afternoon Session

The afternoon session will start with four speakers who will each deliver a brief 10 minute presentation followed immediately by discussion. These presentations will be solicited from our invited attendees prior to the meeting and will focus on *Reproducible Research* and target communities where this is most important (see Section 1.3). After a short break, we will have a presentation and discussion on the results of our survey. We will end with a general discussion and critique led by Stein and Millman.

7.3 Day 1: Evening

We expect the discussion to continue over dinner. After dinner, the workshop organizers will synthesize the day's discussion and prepare a presentation for the morning session.

7.4 Day 2: Morning Session

The second day will begin with an hour long presentation and discussion led by Stein and Millman that summarizes the previous day's meeting. Following this discussion, participants will break into small writing groups to draft some parts of the workshop report. One hour before lunch we will reconvene to present and briefly discuss the rough drafts.

7.5 Day 2: Afternoon Session

After lunch, most attendees will leave. Anyone who remains can join the workshop organizers in working on the report.

7.6 Post Workshop

The organizers will flesh out the draft. The draft will be stored on a collaborative wiki that can be edited by any participant and workshop participants will be encouraged to provide feedback. A final typeset version of the report will be delivered to the NSF by September 2010.

8 Recruitment Plan

8.1 Academia

In addition to soliciting applications to attend the workshop from several mailing lists, we will contact developers directly. We will target the 85 components of Sage and related open mathematical software projects, including PARI, Singular, GAP, Macaulay2, CVXOPT, and Maxima. Millman will contact developers involved in scientific computing in Python, including NumPy, SciPy, matplotlib, pymvpa, mdp, FiPy, and PyDSTool. People interested in participating will apply to the organizers, and we will select applications based on available funding and the potential of applicants to contribute.

8.2 Government and Industry

The PIs have extensive contacts in industry and will recruit participants from there. This includes contacts with Google, Microsoft, Boeing, and Enthought, who have all funded work on mathematical research software, much in conjunctions with the PIs. The PIs also have contacts in the financial sector (e.g., Lisa Goldberg at MSCI Bara `http://www.mscibarra.com/`), government and, and will solicit their input on the workshop and the institute.

8.3 Underrepresented Minorities, Women, and Persons with Disabilities

We will ensure that some underrepresented minorities and persons with disability are invited to participate in the workshop, and ask them for ideas about other people to invite. The PIs have long-term collaborations that they can draw upon to ensure this.

One of the Sage developers recently compiled a list of women who have contributed code to Sage, and we will contact some of them about applying to participate in the workshop and their ideas for people to invite. Moreover, the PI (Stein) obtained funding from Kristin Lauter (head of cryptography at Microsoft Research) to run a "Women in Sage Days" workshop soon, and will encourage attendees from the "Women in Sage Days" to apply for the workshop.

8.4 Students for Pre-workshop Data Gathering

After making a list of areas to cover, we will start by emailing our local departments, then the Sage and SciPy/NumPy lists. We will also contact S. Easterbrook at Toronto and see if he can recommend a student to help us, since some of our background projects are inspired by his research.

9 Budget

We are requesting $48K. The majority of these funds (about $42K) will be used to cover travel ($950), accommodation ($350), and food ($150) for 30 workshop attendees. Most of the workshop attendees will be staying two nights at the hotel (about $175/night). We will also have a catered working breakfast, lunch, and dinner on the first day of the workshop. For the second day we will provide a working breakfast and lunch. We also request 6 stipends ($1K/each) for students in the working groups to create the surveys described elsewhere in this proposal.