Rebecca Lauren Miller
Phone: (262) 995-8347
Email: millerrl@slu.edu
Address: 1000 Washington Ave, Apt 305, St. Louis, MO 63101
St. Louis, Missouri / Central Standard Time
Saint Louis University

Background:

What are your technical skills, education, experience, etc. Especially make sure to explain with what level of mathematics you are comfortable with and on what level you would like to program.

I received my B.A. in Mathematics at Carthage College in Kenosha, WI. I am currently attending Saint Louis University as a M.A. graduate student.

I have experience executing research in mathematics with Dr. Ben Hutz, researching Moduli Spaces and Dynamical Systems as well as reviewing and executing code in Sage. I have worked with Dr. Hutz for the Spring of 2016 and Spring of 2017 semesters. I was able to work with him and Dr. Paul Fili for Sage GSOC 2016. My project was Moduli Space of Dynamical Systems. For this project I implemented two algorithms to increase the functionality of moduli space.

I have also participated in the Student Undergraduate Research Experience, at Carthage College, with Dr. Aaron Trautwein, studying Knot Theory. This study translated into my senior thesis where I wrote a program to find the harmonic equations of knots.

Before my research in mathematics I was executing research in radiochemistry. I worked for over two years at DuraColor, a UV-ink and screen-printing company in Racine, WI. I was the research and development lab assistant. While working at DuraColor, I had to be self-motivated as there were multiple areas of testing and research going on at any time. This taught me to be very organized, and keep accurate documentation. I translated these skills into my mathematics research and studies.

As a second year graduate student I feel comfortable with middle to upper level graduate mathematics.

Who are you? What makes you the best person to work on this particular project? Your personal motivation?

I am both a left and right-brained person. Aside from my love of mathematics, I have a deep passion for art and theatre. My left brained interests allow me to approach problems from many different facets.

I am the best person for working on this project because I already have experience programing in Sage in this area of mathematics. I have a rapport with Dr. Hutz and will be continuing to work with him and Sage. I worked with Dr. Paul Fili as well for GSOC 2016 and look forward to working with him again.

Over the summer I would like to continue working in my field of mathematics, instead of doing it in my spare time while working full-time elsewhere.

What platform and operating-system are you using on your computer?

I am using a Mac with OS X version 10.11.5.

Are you or have you been engaged in other open-source projects?

Sage is my only experience with open-source code. I participated in GSOC 2016.

Do you code on your own pet projects?

Yes, I do. A long-standing problem in topology has been to find differentiable parameterizations of knots. One such representation uses trigonometric polynomials, known as harmonic parameterizations. I made finding differentiable representations of knots simpler by creating a computer program to generate these harmonic parameterizations. The program was written in C++. This program takes polygonal knots in R^3, determines their edge lengths, and then converts their edges into segments from $-\pi$ to π. This allows us to then calculate the parametric equations for each of these edges piecewise in terms of π. A truncated Fourier series is then applied to approximate the linear functions representing the paths of these polygonal knots, giving us harmonic parameterizations. All of the knots found by this program are represented as 15^{th} degree polynomials, while a smaller degree could possibly maintain their equivalence class. For further research my hope is to get the knots into their lowest degree form. Then get the knots in their lowest energy form, i.e. evenly spread out in R^3. I would also like to apply known invariants to check for knot equivalences.

Are you a Sage user, how long do you know Sage?

I am a Sage user, my username is rlmiller and I have been using Sage since January 2016. I've been doing research in dynamical systems with Dr. Ben Hutz.

Closed Tickets I've worked on:

Reviewer: 22293, 22269, 22268, 21117 ,20780, 20451, 20168, 20067, 20059, 19979.

Author:

19889: Fixed style of morphism folder.
19891: Fixed style of projective products file.
20262: Adds the point transformation matrix for projective space. This function takes two sets of *n+2* points, one being our source and the other being our target. The function then determines that no *n+1* subsets are linearly dependent, and then finds the unique transformation matrix that takes our source to our target.
20650: Added function to determine whether an equation is a polynomial or not by checking if it has a totally ramified fixed point. (is_polynomial)
Then used this function to create make_look_poly, which puts polynomials in the form "x**n + a*x**(n-2) +...+c", where a and c are constants.
20820: Part of GSOC 2016: Implementing invariant set algorithm from the paper [FMV]. Given that the set of n th preimages of fixed points is invariant under conjugation find all elements of PGL that take one set to another.
21248: Part of GSOC 2016: Implementing the reduced forms algorithm from *On the reduction theory of binary forms* for both polynomials and projective morphisms. Also includes Newton's method for 2 variables, could be further expanded in the future.
22265: Fixed dynatomic polynomial. Needed to add try/except to homogenize in affine_morphism.py in order to fix a dynatomic error that returns the wrong parent in projective_morphism.py
22580: Normalize nth_iterate_map in projective_morphism.py.

Title, Project Synopsis: a short description and summary of its aim and scope.

Expanding the Functionality of Dynamical Systems.

As a member of the sage-dynamics community, researchers have compiled a wishlist for algorithms and functionality they would like added. I would like to shorten the wish list for us.For my project I will be completing some desired additions to SAGE from the Sage Dynamics Wiki. I will implement Well's Algorithm, strengthen the numerical precision in cannonical_height(), as well as implement reduced_form() for higher dimensions.

Wells Algorithm is important because it finds canonical height's without having to calculate the resultant. This algorithm being implemented will cut down on computational time, because factoring the resultant can be difficult. The algorithm is found in *Computing The Canonical Height of a Point in Projective Space* by Eliott Wells. It is important to note that this only works in dimension one projective space over the rationals and Euclidian Domains, this allows us to have a meaningful division algorithm to calculate the greatest common divisors over our ring. What Well's algorithm does is instead of computing the limit individually at places of bad reduction, we can now do this all at once. Now we can look at the primes of bad reduction for the evaluation of a homogeneous lift at a point with relatively prime coordinates.

Next I would change canonical heights functionality to be over the complex interval field. This would allow us to keep track of the numerical precision involved in the calculations as real interval field for floating point computations keeps track of errors. Now the cannonical_height() function lives in projective_point.py. But the function doesn't handle precision consistently. The function can return greater than 53 bits even if out output is not that accurate. We are able to calculate the error bound due to the tail of the geometric series but it can return more decimal places than is accurate.

As part of GSOC 216, I added the reduced_form algorithm from *On the reduction theory of binary forms* by Michael Stoll and John Cremona. This algorithm only works for P^1. This algorithm takes binary forms, which can have very large and hard to work with coefficients and put them into reduced form by finding an element of SL(2,Z) that gets them into reduced form; the reduced form having "nicer" coefficients to work with. I would like to implement the algorithm from Stoll's *Reduction Theory Of Point Clusters In Projective Space* which will allow us to expand reduced_form() to P^n. Much like the P^1 case, this new reduced form has us compute a covariant z of a point cluster, Z, and move it to a domain. This time instead of the fundamental domain we are finding the covariant that lies in the space $H_{n,C}$ of positive definite Hermitian forms in $n+1$ variables, where C is the complex field. A Hermitian Form is a mapping, φ, from a vector space, V, to the Complex field, C, with specific properties such as that as $\varphi : V \times V \rightarrow C$. Where for a,b in V. $\varphi(a,b)$ is the complex conjugate of $\varphi(b,a)$. These Hermitian forms can be represented as matrices and thus allow us to work in P^n. Taking this covariant z and moving it to $H_{n,C}$ will return a polynomial with a reduced form, the coefficients will be smaller and easier to work with.

If time remains I would like to assist in finishing ticket number ***#21129 arakelov-zhang pairing of rational maps***, which was opened last July. In this ticket a function has been added to compute the dynamical Arakelov-Zhang pairing of two rational maps defined over number fields. The Arakelov-Zhang pairing measures the dynamical closeness of two maps, the closer two maps are to each other the closer to zero is the number it returns. However, we still need to add the functionality to calculate the Lipschitz Constants, and a way to keep track of precision. We need to calculate Lipschitz constants, because these constants tell us imprecise we are getting.The function is returning an answer but there is no way to know how accurate it is at this point in time, this will require converting it to the Complex Interval Field.

If time remains and this ticket had closed, which from talking to Dr. Paul Fili it should not be. I would be interested in implementing some normal forms such as Milner Normal Form for quadratic rationals. Patrick Ingram's normal form for post-critically finite polynomials, is found in his paper *A Finiteness Result for Post-critically Finite Polynomials.*

What is your personal involvement or relationship with your proposed project?

I have been working with expanding the functionality in SAGE for dynamical systems for over a year under the guidance of Dr. Ben Hutz. Part of this project would also be a continuation of my GSOC 2016 project, now I will be extending reduced_form() to P^n. I also worked with Dr. Paul Fili for GSOC 2016, he has worked on canonical_heights() before, and has worked in the realm of Well's algorithm before. Dr. Fili is also one of the authors of ticket #21129.

Details: describe all the details and explain modules or parts of your whole project. Break down the whole project into individual tasks - as good as possible - and describe deliverable and quantifiable results for each of them. It also helps if you have already discussed this with a possible mentor.

This project involves implementing two algorithms, and altering some existing code, see project synopsis.

Schedule: A timetable, including special circumstances like exams or holidays, for the individual tasks.

May 4th- 30th: During our community bonding period I will read though the two papers containing the algorithms and discuss them with my mentors. This is to ensure a general knowledge and build towards a deeper understanding.

May 31st - June 20st: I will be implementing Well's algorithm. I would also be writing and testing the code.

June 21st- July 5th :Then I would take the next two weeks to change canonical height to function over the complex interval field (CIF). This will involve moving the code to perform its calculations over CIF, adding new checks to the precision, and several new examples.

July 6th- August 21st: For the final weeks, I will be working on implementing reduced_form() for P^n. Based on how complicated it was to implement reduced_form() for P^1 I want to allow a lot of time.

No holidays or days off needed, will be working from home. If time remains I will work on ticket #21129 or implement Milner and Ingram's normal forms.

Risk Management: Try to anticipate potential problems and explain, how to mitigate them. Propose alternative scenarios, if a particular milestone isn't reached, to still successfully complete the project.

My proposed schedule leaves time to overcome problems. Some potential problems could be gaining a deeper understanding of the papers containing the algorithms. To overcome this problem I would read additional sources and consult with my mentors.