Department of Mathematics September 28, 2020

Three Math Students Describe Their Summer Research

*This past summer, three students, **Philip Huang, Celine Nguyen,** and **Jason Stack** worked with **Professors Jeff Hatley, Brenda Johnson,** and **Leila Khatami** in a six-week research program. At the beginning of the program, the three students worked together to learn SageMath (Sage), a powerful, free, open-source computer algebra system that contains many mathematical packages, including some for Algebra, Algebraic Geometry, Calculus, Combinatorics, Calculus, Linear Algebra, Graph Theory, Group Theory, Number Theory, Statistics, etc.! After ramping up, the students worked in pairs on two different projects: Philip and Jason worked on an elliptic curves project with Professor Hatley, Celine and Jason worked on a project in commutative algebra with Professor Khatami, and Philip and Celine worked on a project in group homology with Professor Johnson. Philip, Celine, and Jason wrote the following about their summer experience.*

In this project, we learned how to utilize Sage, a software package designed for doing advanced mathematical computations, such as collecting data, forming and testing conjectures, and providing concrete examples to illustrate general theorems. We used Sage to explore multiplication maps in an Artinian algebra, elliptic curves, and homology groups.

Commutative Algebra Project (Celine Nguyen and Jason Stack)

Over the period of 6-week, we studied algebraic geometry, commutative algebra and some linear algebra in order to answer the question of finding non-similar multiplication maps in an Artinian algebra. In order to do this, we studied the Hilbert functions and associated Jordan types of different families and looked for patterns of each of the two and three-variable ideals. We built a Sage function for each family of ideals to collect a large set of data. With each function, one to three parameters, depending on the ideal, would be looped over to generate new specific ideals from the family. We proceeded to generate all possible Jordan types for a given Hilbert function. We created functions for each family of ideals we were studying and looped over one to three variables, depending on the family of ideals being studied, to generate new examples from that family. For each family, we used previous functions that we built in Sage to fill in each entry in the dictionary for each ideal example in the family being studied. This data collection helped us view Hilbert functions alongside Jordan types in order to give a better picture of their relationship to search for certain Jordan types that occur for specific Hilbert functions. While we cannot generate an infinite amount of data to find all possible Jordan types, building these dictionaries has given an insight into the relationship between the Hilbert functions and Jordan types that gives us a new method: instead of finding Jordan types, we can build the possible Jordan types for a given Hilbert function.

Elliptic Curve Project (Philip Huang and Jason Stack)

Over the course of the 6-week program, we studied elliptic curves focused on congruences between two elliptic curves modulo a prime. For each elliptic curve, there is a sequence of values denoted a_l, indexed by prime values l. Two elliptic curves are congruent modulo a prime p if for any l value, a_l = b_l modulo p, where a_l is a value from the sequence defining the first elliptic curve and b_l is a value from the sequence defining the second elliptic curve. There is an important result, namely the Sturm bound, that states if the a_l values of two curves are congruent modulo p for l ≤ M, where M is the Sturm bound, then the two curves are congruent modulo p. It is also important to note that the so-called Weil bound gives a range in which each a_l value has to fall, and this is proportional to the square root of l.

(Continued on next page)

The motivating question in our research project was whether or not a relationship between a_l congruences and the corresponding l values existed. We anticipated that as l values increased, the number of non-congruences would increase. As the Weil bound was increasing as l increased, we anticipated that it would become less and less likely that “accidental congruences” would occur, thus increasing the number of non-congruences witnessed at higher l values.

To test this, we created a code using SageMath that ran through elliptic curves, compared their congruences at given l values and recorded whether a congruence or non-congruence occurred. The results indicated that as l values increased, the number of non-congruences witnessed at each l value remained relatively constant. We explain this result by the fact that although the Weil bound is increasing as l increases, the a_l values are also bounded by the prime p for which congruence modulo p is being tested. Thus, once the Weil bound exceeds this bound, which is noted to happen at small l values, the non-congruences witnessed do not increase. We then compared these results to a probabilistic test estimating how many non-congruences would be witnessed at each l value. The results also indicated that the actual amount of non-congruences as a given l value was always less than the probabilistic estimation. This relates to that fact that elliptic curves are specific kinds of modular forms. The a_l values for elliptic curves are not just random integers within the Weil bound, they correspond to the q-expansion for the modular forms defining those elliptic curves. Due to the conditions and restrictions of the q-expansion, these a_l values are not determined randomly, and thus probability cannot tell us everything that is happening. The underestimation confirms the specificity of the conditions on the q-expansion and justifies why probability is not the only determinant in non-congruence witnesses for l values. In future projects, we hope these results can be used to find more general results about modular forms.

Graphs and Homology Project (Philip Huang and Celine Nguyen)

This project studied the homology groups of the clique complexes to answer the following question: for the clique complex of a graph, does the homology distinguish between graphs that are intrinsically linked (knotted) and graphs that are not, or are there other complexes that we can construct that will do so?

Sage has some really helpful built-in functions which help us compute a great number of the homology groups for clique complexes for graphs that are known to be intrinsically knotted and graphs that are known to not. As long as we find out if there is any homology pattern, we can use the information to formulate conjectures about intrinsically knotted graphs and their clique complexes.

Unfortunately, there were no clear patterns between the homology groups, facets, simplicial homology or the degree of vertices of intrinsically knotted graphs and graphs that are not. We had the answer to the aforementioned questions; it is just not the answer we expected. Other complexes, and/or the proofs of intrinsically knotted graphs might be studied in the continuation of this project.

Problem of the Newsletter – September 28, 2020

Last week’s problem: Unfortunately, there were no correct solutions submitted to last week’s problem. A solution, however, is posted at math newsletter sites around Bailey Hall.

This week’s problem: Let Z^2 be the integer-coordinate points *(x, y)* in the plane. For each integer $n \geq 0$, let P_n be the subset of Z^2 consisting of (0,0) together with all points *(x, y)* such that $x^2+y^2=2^k$ for some $k \leq n$. Determine, as a function of *n,* the number of four-point subsets of P_n whose elements are the vertices of a square.

Professor Friedman (friedmap@union.edu) will accept solutions until noon on Friday, October 2. (As usual, when a solution is posted next week, the source of the problem will be revealed.)