COMPUTATIONAL NUMBER THEORY

WINTERSEMESTER 21/22

DR. BARINDER S. BANWAIT

barinder.s.banwait@gmail.com

1. Overview of the course

Computation has been a key part of number theory since our earliest recorded histories of the subject, with examples ranging from modular arithmetic computations in ancient Chinese mathematics (*Chinese remainder theorem*, 3^{th} century CE), to enumeration of Pythagorean triples by ancient Babylonian mathematicians (*Plimpton 322*, 1800 BCE). Computation of number theoretic quantitites were how Euler came upon the statement of Quadratic Reciprocity [Cox11, §1 Part C] and how Gauß formulated higher reciprocity laws [Cox11, §4 Part C]. In modern times, computers have been used for proving theorems [Boo19], disproving old conjectures [Elk88], and formulating new conjectures, most notably the *Birch and Swinnerton-Dyer conjecture*, one of the Clay Millenium problems. Today researchers in this area are aided by several open-source software projects, including the computer algebra systems Sage [The20] and PARI/GP [The21], as well as the *L-functions and modular forms database* (LMFDB) [LMF21], an extensive database of mathematical objects arising in Number Theory.

Part I of the course will discuss several topics in the realm of elliptic curves, modular forms, and higher dimensional varieties which have an algorithmic or computational aspect. During the lectures the essential mathematical ideas will be explained, but mostly students will be guided to discover these topics for themselves using the computer-algebra system Sage. As such the format of the lectures will be practical and hands-on.

Part II of the course will consist in the student working on their own independent project to develop an algorithm, dataset or data visualisation of interest to the computational number theory community. A topic will be chosen in consultation with the course instructor, who will oversee the project and be available for queries and discussion. The student will be evaluated on the basis of this independent project. It is expected that Part II will occupy the last four lectures.

2. Goals of the course

(1) To obtain knowledge of how computers can be used in the study of elliptic curves, modular forms, and higher dimensional varieties.

(2) Ability to use the Sage computer algebra system for number-theoretic investigation.

(3) To create an independent project utilising Sage and/or the LMFDB of benefit to the wider community. The student will be encouraged to make this

project publicly available on GitHub for others to appreciate. If relevant, the outcomes of the project could be incorporated into Sage or the LMFDB.

(4) Working knowledge of Git version control.

3. Topics in Part I of the course

Disclaimer: The following topics are for indication only; depending on how the course progresses some of these topics may end up being dropped.

(1) **Elliptic curves**: Computing rational torsion subgroups and Mordell-Weil ranks. Finding elliptic curves with large rank. Birch and Swinnerton-Dyer's original experiments. The Elliptic Curve Diffie-Hellman (ECDH) protocol.

(2) **Modular forms**: Modular symbols. Computing coefficients of different kinds of modular forms (classical, Hilbert, Bianchi). Congruences between coefficients of modular forms. Level lowering and raising in the LMFDB.

(3) **Higher dimensional varieties**: The Sato-Tate conjecture for higher genus curves. Integers representable as the sum of three cubes. Elkies' disproof of Euler's conjecture on solutions of $A^4 + B^4 + C^4 = D^4$.

4. Possible Student projects

The following possible projects are for indication only; this is not meant as an exhaustive list.

(1) Implement an algorithm in Sage. Examples could be ranks of elliptic curves over number fields, Atkin-Lehner pseudoeigenvalues, or conductors of higher genus curves.

(2) Add some dataset to the LMFDB. This could include bad primes for the Jacobian of genus 2 curves, building blocks of GL_2-type abelian varieties, or visualisations of Hilbert modular forms.

(3) Visualise isogeny volcanoes.

5. Practicalities

1. **Prerequisites.** Students are expected to have a reasonable understanding of Elliptic Curves, Modular Forms, and Algebraic Geometry.

No prior experience of Sage, Python, or the LMFDB will be assumed.

2. **Format of the lectures.** As explained above each lecture will consist in a high-level explanation of a topic, followed by the student working hands-on with those ideas in Sage.

3. **Student Evaluation.** Students will be evaluated based on the clarity and quality of the independent project.

4. **Access to Sage.** Since this course is more in the spirit of *learning by doing*, access to Sage during the lectures is essential; therefore the seminar will meet in a Computer Lab and may use a pool computer during the sessions. Students are also welcome to use their own personal laptops if they have one.

5. **Time and place.** The course will meet in one of the computer labs on the 3rd floor of Mathematikon, at a time to be decided. It will start in the second week of the Semester, i.e., the week beginning Monday 25.10.21.

6. **Registering interest via email.** Students interested in participating in the course are requested to send an email to Dr. Banwait (barinder.s.banwait@gmail.com) with some information about their mathematical background, as well as experience with programming, particularly in Sage or Python.

References

[Boo19] Andrew R Booker. Cracking the problem with 33. *Research in Number Theory*, 5(3):1–6, 2019.

[Cox11] David A Cox. *Primes of the form x2+ ny2: Fermat, class field theory, and complex multiplication*, volume 34. John Wiley & Sons, 2011.

[Elk88] Noam D. Elkies. On $A^4 + B^4 + C^4 = D^4$. *Mathematics of Computation*, pages 825–835, 1988.

[LMF21] The LMFDB Collaboration. The L-functions and modular forms database. http://www.lmfdb.org, 2021. [Online; accessed 5 February 2021].

[The20] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.2)*, 2020. https://www.sagemath.org.

[The21] The PARI Group, Univ. Bordeaux. *PARI/GP version 2.14.0*, 2021. available from http://pari.math.u-bordeaux.fr/.