Final Project

Math 242A • Spring 2024

The final project is your opportunity to use computation to investigate a mathematical topic that we didn't study this semester. The goal of this project is to demonstrate a complete process involving reading mathematical ideas, performing computational exploration, collecting observations, and formulating conjectures. This will be a group project that will result in a computational notebook and a short presentation on the final exam day. Each group will consist of 2 to 4 students.

The following list describes possible topics that serve as starting points for computational investigation. You may also come up with your own topic—if you do, talk with the professor about your topic idea. *Your project must focus on one or more questions that can be answered (at least partially) by computation.*

- **Polygonal Numbers:** Richard Guy considers sums of polygonal numbers in this paper. Can you provide computational evidence that every positive integer is the sum of three triangular numbers? How about sums of squares? Or pentagonal numbers? The paper states some results of these types. Do your own investigation and see what patterns you discover and what conjectures you can make!

- **Happy Numbers:** Start with a number N. Add up the squares of its digits. Repeat. If you eventually reach 1, then N is a *happy number*. Happy numbers can be considered in different bases, and properties such as their density can be explored, as in this paper by Gilmer. What patterns do you find in happy numbers? What related questions and generalizations can you explore?

- **Mean-Median Map:** Explore the mean-median map, as defined in this paper by Chamberland and Martelli. Do you find computational evidence that supports the conjectures in their paper? How could you generalize the mean-median map?

- **Unbounded Ducci Sequences:** Investigate the sequences that arise from iterating Ducci's function and provide evidence for Theorem 1.1 in this paper by Chamberland. Then investigate other results from the paper and try to make your own conjectures. The paper ends with some open questions. Can you come up with your own questions as well?

- **Moessner's Theorem:** Can you provide computational evidence in support of Moessner's Theorem and the related theorems in this paper by Kozen and Silva? Then come up with your own variant of Moessner's construction and look for numerical patterns. What do you observe? What questions and conjectures do you have?

- **Sums of Powers:** Ho, Mellblom, and Frodyma establish some theorems about sums of powers of consecutive integers in this paper. Verify their results computationally and then generalize. For example, you might consider sums of powers of integers in an arithmetic sequence.

- **Monte Carlo Estimation:** Monte Carlo simulations are probabilistic methods that can find approximate solutions to difficult problems. For example, Diniz, Lopes, Polpo, and Salasar use Monte Carlo simulation to solve the "sticker collector's problem." Implement their methods and explore how the number of total distinct stickers and the number of stickers per package affect the results. Then explore other generalizations.

- **Square-Sum Pair Partitions:** Hamilton, Kedlaya, and Picciotto discuss some simple number problems and various generalizations in this paper. Write code to solve some of these problems, and look for patterns in the results. What properties can you discover? What conjectures can you make?

- **Subprime Fibonacci Sequences:** See this paper by Guy, Khovanova, and Salazar for an introduction. What cycles occur in subprime Fibonacci sequences? What patterns do you notice among the even and odd terms of these sequences? What other questions do you find to investigate regarding subprime Fibonacci sequences?

- **Simulated Annealing:** Simulated annealing is a probabilistic technique that can approximate solutions to nonlinear optimization problems. See Section 6.5 in our *Computational Mathematics* text. Implement a simulated annealing algorithm and use it to solve problems such as the bin-packing problem or the traveling salesperson problem. Explore how various assumptions or parameter choices affect the solutions.

Timeline

- **May 3:** Complete the Project Planning Survey. This will ask you for possible topics and who you do (or don't) want to work with.

- **May 6:** Project topics and teams finalized.

- **May 8–13:** Class time to work on final projects. Check in with the professor regarding progress and questions.

- **Week of May 13:** Finish your project and prepare your presentation. Scheduling a practice presentation with the professor is recommended.

- **May 16, 1:00pm**: Project due. Group presentations about what you investigated and discovered.

Deliverables

- **Notebook:** Prepare either a Mathematica notebook or a Google Colab notebook. As usual, submit code that runs, and throughly explain your methodology, observations, and conclusions. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course).

- **Presentation:** During the final exam period, your group will give a 10 minute presentation explaining what you did and what you discovered.

- **Self and peer evaluation:** This brief survey will ask you to reflect on your own contributions and the contributions of your group members to your project.

Grading

This project will be graded on the EMRN scale, as described in the syllabus. To receive a grade of *Meets Expectations*, your project should exhibit the following characteristics:

- You use computation to investigate mathematical questions.
- Your code is appropriate for the given tasks and produces reasonable output.

- Your reasoning is explained using sentences, and your notebook is well-formatted and easy to read.
- Your presentation summarizes your work and your observations.
- No significant gaps or errors are present.

To receive a grade of *Excellent*, your project should further exhibit the following:

- The project involves mathematical depth, demonstrating your ability to apply your mathematical knowledge and computational observations to new questions, thus discovering mathematical ideas for yourselves.
- Computational methodology demonstrates mastery of the computational techniques that we have studied in this course.
- Exposition is clear and precise, thoroughly explaining your methodology and reasoning. Relevant definitions and assumptions necessary for the for your work are clearly stated and discussed.
- You state at least one original conjecture. This must be a precise mathematical statement based on what you have observed. In your discussion, you clearly distinguish between what is known (i.e., what has been proved) and what is conjectured. Optionally, you may prove something related to your topic.
- Mathematica/Python code is of high quality, demonstrating skillful use of programming constructs (e.g., variables, lists, functions, modules).
- Your presentation demonstrates that you have carefully thought and practiced communicating your work.
- The work exhibits creativity and insight.

Note: It is possible that members of the same team may receive different grades, according to their contributions to the project.