Final Project

Math 242

Use computation to investigate some mathematical topic. *Your project must focus on one or more questions that can be answered (at least partially) by computation.* You may work alone or with a partner. You will turn in a Mathematica or Python notebook and give a short presentation on what you investigated and discovered.

The following list contains links to math papers or articles that explain a topic and serve as a starting point for computational investigation. You may also come up with your own topic—if you do, talk with Prof. Wright about your topic idea.

- Conway's Subprime Fibonacci Sequences: What cycles occur in the *subprime Fibonacci sequences* defined in this paper? 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?

- Every Number is Expressible as the Sum of How Many Polygonal Numbers? 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!

- How do computers compute values such as $\sin(\theta)$ and $\cos(\theta)$? Taylor series provide one possible way, and the CORDIC algorithm provides another. Implement both and compare! How many correct digits are produced by each term or iteration of these methods? How much time does it take for each algorithm to compute a desired number of correct digits? Does this depend on the specific value of θ? (For this topic, Mathematica is recommended due to its high-precision numerical capabilities.)

- Bin-packing by Simulated Annealing: What is a bin-packing problem? How can simulated annealing solve such problems? Is simulated annealing efficient and does it produce optimal solutions for these problems?

  See also: A Morph-Based Simulated Annealing Heuristic for a Modified Bin-Packing Problem

- Unbounded Ducci Sequences: Investigate the sequences that arise from iterating Ducci's function and provide evidence for Theorem 1.1 in the paper. 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?

- On Moessner's Theorem: Can you provide computational evidence in support of Moessner's Theorem and the related theorems in this paper? 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?

Deadlines

- **Wednesday, May 12**: Choose your topic by this day.
- **Monday, May 17**: Turn in 1–2 paragraphs describing what you have accomplished, what remains to do, and what questions you have.
- **Final exam period**: Project due. Give a brief presentation (4–5 minutes per person) about what you investigated and discovered.

Deliverables

- **Notebook:** Submit either a Mathematica notebook or a Google Colab notebook. As usual, submit code that runs, and explain what your code does. Include the items mentioned in the rubric below. 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, each person/group will give a short presentation explaining what they did and what they discovered. Presentation length should be not more than 5 minutes per person.

Grading Rubric

Your notebook will be graded on a scale of 0 to 16 points. The following rubric gives characteristics of notebooks that will merit sample point totals. (Interpolate the following for point totals that are not divisible by 4.)

16 points. Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and conjectures are discussed. Notebook is well-formatted and easy to read.

12 points. Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and conjectures.

8 points. Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

4 points. Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

0 points. Notebook is not turned in.