Iterated Functions Project

Math 242

due Friday, February 28

Investigate one of the following areas related to iterated functions.

1. **Collatz sequence lengths:** Let the *length* of the Collatz sequence starting with n be the number of iterates until 1 is reached. Are all positive integers possible lengths? Are some lengths more common than others? If so, why? Can you find and explain any patterns in Collatz lengths?

2. **Collatz maxima:** Let the *maximum* of the Collatz sequence starting with n be the largest number obtained in the sequence. Are all positive integers possible maxima? Are some maxima more common than others? If so, why? Can you find and explain any patterns in Collatz maxima?

3. **Collatz-like functions:** Can you find other functions whose iterates produces patterns similar to the Collatz function? In what ways are they similar? Does this make you think the Collatz function is special, or is there a large class of functions with similar behavior?

4. **Logistic map:** Try to support or refute the claim that if k is any positive integer, then there is a value of r that produces a cycle of length k. For what values of r do bifurcations occur? For what intervals of r do you notice chaotic behavior? What patterns do you notice?

5. You may also investigate your own questions. If you want to do this, talk with Prof. Wright before Wednesday about the questions you would like to investigate.

In your investigation, be creative! Try to explain your observations as best you can. You may read about the Collatz conjecture, the logistic map, or other topics. However, the purpose of this project is to make discoveries by computational experimentation, not by reading about what other people have done.

As usual, your Mathematica notebook should indicate not only what you computed, but also how well you understand what you did. A list of calculations with no reasoning will not suffice. Your goal should be to communicate your solution to another person (e.g., another student at your level who is not in this course).

Only submit code that actually runs. If you can't get something complicated to work, try something simpler. It's better to turn in an incomplete assignment that runs instead of a "complete" assignment that doesn't run.

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.