Math 466/666: Programming Project 1

This project explores the computation of the Foias constant which has a natural relationship to the prime number theorem.

For this project the class has been randomly grouped into teams consisting of 3 or 4 students. Each team should present their work in the form of a typed report using clear and properly punctuated English. Pencil and paper calculations may be typed or hand written. Where appropriate include full program listings and output.

Every team member should participate in the work and be prepared to independently answer questions concerning the material. One report per team must be submitted through WebCampus to complete this project. Make sure the report addresses each of the items listed below and then upload the report as a single pdf file.

1. List the members on your team and explain how the work for this project was conducted. Please provide details concerning how many meetings were held, what was discussed at each meeting and what work was done between meetings.

Further include a written statement attesting that the submitted report represents the original efforts of the team members listed and, in particular, does contain the work of other students in the class.

(i) It is fine to consult books, published papers and online resources while working on this project. If you do so, every outside source of information should be cited with a proper bibliographic reference at the point it is used.

(ii) There is no need to cite any additional help provided by the instructor during class or individually during office hours.

2. Given $\alpha \in [1,2]$ consider the recurrence relation

$$x_{n+1} = \Big(1 + \frac{1}{x_n}\Big)^n \qquad \text{where} \qquad x_1 = \alpha.$$

Set $\alpha = 1$ and write a computer program in Julia that computes x_n for $n = 2, \ldots, 50$. Please include the full program listing and output. For reference,

$$x_{15} \approx 1.2634346914781789 \qquad \text{and} \qquad x_{16} \approx 6283.87526674291.$$

3. Based on the numerical evidence in the previous step, make a conjecture regarding the value of the limits

$$\lim_{n\to\infty} x_{2n} \qquad \text{and} \qquad \lim_{n\to\infty} x_{2n+1} \qquad \text{when} \qquad x_1 = 1.$$

Explain your reasoning in as much mathematical detail as possible.

4. Change the program in the previous step to compute the values of x_n when $\alpha = 2$. Look at the output and now make a conjecture regarding the value of the limits

$$\lim_{n\to\infty} x_{2n} \qquad \text{and} \qquad \lim_{n\to\infty} x_{2n+1} \qquad \text{when} \qquad x_1 = 2.$$

Explain the reasoning behind your conjecture.

5. [Extra Credit] Use rigorous mathematical analysis to prove the conjectures made in the previous two steps.

6. Define

$$\alpha_* = \sup\Big\{\alpha : |x_{2n+1}| \text{ is bounded as } n \to \infty \Big\}.$$

Intuitively α_* is the largest value of α such that $|x_{2n+1}|$ is bounded. Explain in details how the interval bisection method could be used to approximate α_*.

7. Write a computer program that bisects the interval $[1, 2]$ to find an approximation of α_* good to at least 4 significant digits. Include the full program listing and output.

8. Explain theoretically how many times the interval $[1, 2]$ needs to be bisected to ensure the resulting approximation is good to at least 4 significant digits. What if 6 significant digits are desired? How about 8 significant digits?

9. Define

$$\alpha^* = \inf\Big\{\alpha : |x_{2n}| \text{ is bounded as } n \to \infty \Big\}.$$

Intuitively α^* is the smallest value of α such that $|x_{2n}|$ is bounded. Modify the program from the previous step to obtain an approximation of α^*. Include the output and describe what modifications were made to the code.

10. Based on the numerical evidence obtained in the previous two problems conjecture whether the values of α_* and α^* are equal or different.

11. [Extra Credit] Use rigorous mathematical analyisis to prove the conjecture stated in the previous step. Alternatively, further support your conjecture by computing additional digits of α_* and α^* using the `BigFloat` arbitrary precision arithmetic built into the Julia programming language.