MAT 581 MIDTERM PROJECT TOPICS

Introduction: Projects are mostly computational (programming) assignments that are larger in scope than a homework problem. Your work should consist of a Matlab file with a program and a separate file that explains your method. The topics are listed below.

Project 1. *The probability that a random 3rd degree polynomial has an attracting fixed point.*

Description. Let A, B be the first two digits of your SUID number.

Let $p(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3$ where a_0, a_1, a_2, a_3 are random numbers taken from the uniform distrubition on the interval $[-A, B]$. Estimate the probability that p has an attracting fixed point on the real line. That is, the probability that there is a real number x^* such that $p(x^*) = x^*$ and $|p'(x^*)| < 1$.

To estimate the probability of something happening, one can use a `for` loop with a large number of runs (more is better for accuracy of the estimate, but you also want your code to finish in reasonable time). In each run of the loop, choose numbers a_0, a_1, a_2, a_3 using `rand` and they try to decide whether this cubic polynomial has an attracting fixed point.

The probability of an event is estimated by the frequency of its occurrence after a large number of tries; that is, $P \approx S/T$ where T is the number of tries and S is the number of successes (in this context, the number of polynomials which have an attracting fixed point). The output of this project is a single number.

How to decide the existence of a fixed point? Some possible approaches are:

- Use Matlab's `roots` command to find all roots of $p(x) - x$, and evaluate $|p'|$ at those roots. Note that `roots` includes complex (non-real) roots, which should be ignored. A conditional statement with the command `isreal` can help.
- Try to iterate p to see if it converges. If it converges, the limit is probably an attracting fixed point (the probability of encountering a neutral/indifferent fixed point is zero). However, convergence of iteration depends on the initial point, so it will fail to converge for some points even if an attracting fixed point exists. Maybe try several starting points.

Project 2. *The distribution of the number of real roots of a random 5th degree polynomial.*

Description. Let A, B be the first two digits of your SUID number.

Let $p(x) = a_0 + a_1 x^1 + \cdots + a_5 x^5$ where $a_0, a_1, \ldots, a_5$ are random numbers taken from the uniform distrubition on the interval $[-A, B]$. Let N be the number of its real roots; this is a random variable. Algebra tells us that $1 \leq N \leq 5$: a polynomial cannot have more roots than its degree; and a polynomial of odd degree with real coefficients has at least 1 real root.

The goal is to estimate the probabilities $P[N = k]$ for $k = 1, 2, 3, 4, 5$: that is, the probability of having 1 real root, of having 2 real roots, etc. The output of this project is five numbers.

Matlab's command `roots` will be helpful. See Project 1 for its description (and Matlab's built-in help).

Project 3. *Lyapunov exponents of periodic points.*

Description. We have encountered fixed points, that is solutions of $f(x) = x$. A periodic point is a more general concept. To defined it, first introduce notation f_k for $f \circ \cdots \circ f$ where f is repeated k times. For example, $f_1(x) = x$, $f_2(x) = f(f(x))$ and $f_3(x) = f(f(f(x)))$.

By definition, a number x is a **periodic point** of order k of function f if $f_k(x) = x$ and k is the smallest number with this property. For example, a periodic point of order 3 satisfies $f_3(x) = x$ but $f_2(x) \neq x$ and $f_1(x) \neq x$. The **orbit** of this point consists of the numbers $x, f(x), f(f(x))$.

We know that for a fixed point x, it is important to distinguish between $|f'(x)| < 1$ and $|f'(x)| > 1$. The **Lyapunov exponent** of a fixed point x is simply the number $h = \ln|f'(x)|$, so a negative exponent indicates an attracting point and a positive exponent indicates a repelling point. For a periodic point, the Lyapunov exponent is the average value of $\ln|f'|$ on the orbit. That is, if the orbit is $x_1, x_2, \ldots, x_k$, then the Lyapunov exponent is

$$h = \frac{1}{k}(\ln|f'(x_1)| + \ln|f'(x_2)| + \cdots + \ln|f'(x_k)|)$$

This number is important because it determines whether the periodic orbit is stable ($h < 0$) or unstable ($h > 0$).

Let A be the first digit of your SUID. Define

$$f(x) = (3.9 + A/100)x(1 - x)$$

so that, for example, $f(x) = 3.92x(1 - x)$ if $A = 2$. Write a script that asks the user for a number k, and outputs some periodic point of order k and its Lyapunov exponent.

Possible approaches:

- Since f_k is a polynomial, one can use `roots` to find all solutions of $f_k(x) = x$. This approach requires finding the coefficients of f_k first.
- The behaviour of f tells us to look for periodic points in the interval $[0, 1]$. One could try bisection based on this interval or a smaller interval, to find some root of the function $g(x) = f_k(x) - x$.
- One can also use `fzero` with some initial point in $[0, 1]$.
- Regardless of which approach is used, it is important to check that the solution of $f_k(x) = x$ is indeed a periodic poitn of order k, that is, $f_j(x) \neq x$ for $1 \leq j \leq k - 1$. When checking this, keep in mind that computer arithmetics is not perfectly accurate.

Project 4. *Self-intersections of a closed curve*

Description. Given random positive numbers A_1, A_2, A_3, A_4 (called amplitudes) and random numbers t_1, t_2, t_3, t_4 in the interval $[0, 2\pi]$ (called phases), consider the closed curve defined by parametric equation

$$x = \sum_{k=1}^{4} A_k \cos k(t - t_k),$$
$$y = \sum_{k=1}^{4} A_k \sin k(t - t_k)$$

where $0 \le t \le 2\pi$. This curve is closed because $t = 0$ and $t = 2\pi$ correspond to the same point (x, y).

The goal is to find whether this curve intersects itself, that is, whether it passes through some point more than once. Mathematically, self-intersection means there are two numbers t, s such that $x(t) = x(s)$ and $y(t) = y(s)$ but $t - s$ is not an integer multiple of 2π.

Since $x(t) = x(s)$ and $y(t) = y(s)$ is a system of two equations with two unknowns, it should be possible to use multivariate Newton's method to find some roots (t, s) of the system. But one should be careful to avoid the roots where $t - s$ is a multiple of 2π. Other approaches may be possible, for example using complex numbers to rewrite the curve as a polynomial with complex coefficients, and doing more algebra with it.

In addition to displaying the decision (does the curve intersect itself or not), the program should plot the curve.