Math 308, Sprint 2012, Project 2

Due date: Monday, 27 Feb 2012

Instructions. While there are no specific neatness requirements, your work should look professional, and you should submit it in a form that would be appropriate for submitting to a boss at a job you care about. Credit will be given for full, complete, and thoughtful analysis; however, a good report is not necessarily a long report. Do not submit fluff or filler, but only quality work that you are proud of.

Submit your project as a group with all group member's names clearly labeled on the front. Submit all of your code[1], along with any relevant graphs, mathematical calculations, and explanations. Make sure graphs are well labeled and referenced, and that they are easy to read. Please note that each person is responsible for 100% of the work, so it is your responsibility to keep all group members on task, or to finish the work yourself.

Objectives. The major objectives of this project are:

- To give you an introduction to difference equations (these can be thought of as "discrete" as opposed to "continuous" differential equations).
- To learn about "for loops" which are a major concept in programming, and are extremely useful in Matlab.

Part 1. (No need to submit this part; just try it out.)

Warm up by trying the three pieces of Matlab code below. It's easier to type these into a *.m file rather than to enter them one by one in the command line.

Pro tip: Typing code by hand will help you learn it much better than copy/paste.

```
for i = 1:5
  i^2
end
```

```
N=7;
for i = 5:N
   sprintf('The cube of %g is %g.',i,i^3)
end
```

```
a = 3; % Set the value of 'a' to 3.
for i = 1:4
  % Compute 2a+1 and replace a by this new value.
  a = 2*a+1;
end
a
```

[1] There is no need to submit code which was *not* written by your group.

Part 2.

Discrete Dynamical Systems. Differential equations are "continuous" models, but there are also "discrete" models, which are called "Discrete Dynamical Systems" or sometimes "difference equations." Consider the familiar exponential growth population model with growth rate r. If we let P_n denote the population at the n^{th} time step, then P_{n+1}, the population at the $(n+1)^{\text{st}}$ time step is given by

$$P_{n+1} = rP_n$$

More generally, consider the function $f(x)$. We can make in iterative process out of this as follows. Start with a "seed" value x_0 (you pick this), and then compute the next iterates as

$$\begin{array}{rcccl} x_1 & = & f(x_0) & = & x_0^2 - 1, \\ x_2 & = & f(x_1) & = & x_1^2 - 1, \\ \vdots & & \vdots & & \vdots \\ x_n & = & f(x_{n-1}) & = & x_{n-1}^2 - 1, \end{array}$$

and so on.

Cycles, periods, and orbits. Consider the function $f(x) = x^2 - 1$. Use a `for` loop in Matlab to test some seed values. The collection of all of the values that you can get out by this iterative process from a given seed value is known as the ***orbit*** of the cycle. Find a seed value that has an orbit of size 2.

If you get a sequence that repeats, say $a, b, c, d, a, b, c, d, a, b, c, d, \ldots$, this is called a ***periodic cycle***. The length of a period is the smallest repeated sequence. This particular example has a period of length 4 (but by definition, it is not of length 8, 12, 16, etc.). Find all the seed value with a period of 1. (There are called "fixed points" or "equilibrium solutions.") You can get an idea using Matlab, but show your results algebraically.

Sometimes, the cycle doesn't begin right away. Consider a sequence that looks like this: $a, b, c, d, c, d, c, d, \ldots$. If the "$c, d$" pattern repeats forever after the start up, we say it is ***eventually periodic***. This example is eventually periodic with period length 2. Notice that the orbit of this cycle is $\{a, b, c, d\}$. Find a seed value for $f(x)$ that is not periodic, but is eventually periodic with period 2.

Now consider the function $M(x) = x^2 + c$ for different values of c, but always with the same seed value $x_0 = 0$ (we are thinking of the process $x_{n+1} = M(x_n)$). Notice that for most values of c, $|x_n|$ gets very large (after about 20 iterations or so). However, for *some* values of c, x_n stays bounded forever, such as when $c = -0.5$. First, try to figure out all the values of c which make x_n stay bounded. Next, answers the same question when you allow c to be complex. What does the region look like? Try to map out a picture of it. (**Expect to spend the majority of your time on this question.** The previous questions should have been fairly straight-forward.)

Matlab tips: Complex numbers in Matlab are written as, for example, $2+3j$ instead of $2+3i$. Here, $j = \sqrt{-1}$. You can find the "size" of a complex number $z = a+bi$ by computing its "modulus" or absolute-value $|z| = \sqrt{a^2+b^2}$, which is written as `abs(z)` in Matlab. After you define an `x` and `y` range you want to look at, look at the full grid like this:

```
[X,Y]=meshgrid(x,y); c = X + j*Y; pcolor(abs(c)); shading flat;
```

So far, this will just look like concentric circles, since it is just plotting `|c|`, which is the distance of `c` from the origin. Next, set `z=zeros(size(c))` for you first iteration, then `z=z.∧2 + c` for the next iteration, and so on. Plot `z` with `pcolor(abs(z))`. This should be enough to get you started. Have fun!