Math 374, Dynamical Systems, Fall 2017
Computer Project #3

Computing Feigenbaum's Constant

DUE DATE: Thursday, November 16, start of class

The goal of the project is to compute a very famous constant discovered by physicist Mitchell Feigenbaum in 1975 while working at Los Alamos National Laboratory. Although Feigenbaum received his doctorate from the Massachusetts Institute of Technology in elementary particle physics, he was also fairly gifted in mathematics. After hearing a lecture by Field's medalist Stephen Smale on period-doubling, Feigenbaum decided to investigate the rate at which the Logistic Map undergoes a sequence of period-doubling bifurcations en route to chaos. Partly inspired by ideas from physics involving Renormalization Theory, Feigenbaum, using a simple HP-65 hand-held calculator, discovered an amazing fact about systems which undergo a series of period-doubling bifurcations—they **all** do so at the same rate. This special rate is now known as **Feigenbaum's constant**. Just after predicting his special constant was universal in dynamical systems, Feigenbaum called his parents and told his mother he was going to be famous. For more information on this interesting story and to motivate yourself for the project, read the sixth chapter *Universality* in Gleick's book on chaos.

Your aim in this project is to compute as many decimal places of Feigenbaum's constant as possible. Fortunately, you get to use a computer (e.g., Maple, C++, Java, etc.) as you go through the same steps that Feigenbaum undertook himself. You will approximate Feigenbaum's constant for three families of dynamical systems, the Quadratic Family $Q_c(x) = x^2 + c$, the Logistic Map $F_\lambda(x) = \lambda x(1 - x)$ and the Sine Family $S_b(x) = b \sin x$. Whichever lab group computes the most decimal places wins a prize. **Note:** You must actually compute the constant. No credit will be given for simply looking the value up on the Internet or in a book.

It is **required** that you work in a group of two or three people. Any help you receive from a source other than your lab partner(s) should be acknowledged in your report. For example, a textbook, web site, another student, etc. should all be appropriately referenced at the end of your report. The project should be **typed** although you do not have to typeset your mathematical notation. For example, you can leave space for a graph, computations, tables, etc. and then write it in by hand later. You can also include graphs or computations in an appendix at the end of your report. Your presentation is important and I should be able to clearly read and understand what you are saying. Only **one project per group** need be submitted.

Defining Feigenbaum's Constant

Suppose that f_c is a family of differentiable dynamical systems which undergoes a series of period-doubling bifurcations. This means that as c is varied continuously, f_c has an attracting periodic point beginning with period 1, then period 2, followed by period $4, 8, 16, 32, \ldots$. This is the start of a typical orbit diagram for most of the dynamical systems we have been studying.

Let x_0 be a critical point for f_c. Define a sequence of parameter values c_n by setting c_n equal to the parameter value where x_0 is on a periodic orbit of prime period 2^n. These are values where the orbit is super-attracting since $(f_{c_n}^{2^n})'(x_0) = 0$ (by the chain rule). Specifically, define

$$\begin{aligned}
c_0 &= c\text{-value where the critical point is on a period } 2^0 = 1 \text{ orbit.} \\
c_1 &= c\text{-value where the critical point is on a period } 2^1 = 2 \text{ orbit.} \\
c_2 &= c\text{-value where the critical point is on a period } 2^2 = 4 \text{ orbit.} \\
c_3 &= c\text{-value where the critical point is on a period } 2^3 = 8 \text{ orbit, etc.}
\end{aligned}$$

Note that there may be more than one possible value of c_n for a given n. However, we are only interested in finding the values that occur in the initial series of period-doubling bifurcations. These values will be relatively close together. (Other parameter values may occur further along in the bifurcation diagram, but we shall ignore these values.)

It is not the particular values of c_n that concern us, but the ratio of the difference between successive values. This is a measure of how fast the period-doubling bifurcations occur. Specifically, Feigenbaum's constant is defined to be

$$K = \lim_{n\to\infty} k_n \quad \text{where} \quad k_n = \frac{c_{n+1} - c_n}{c_{n+2} - c_{n+1}}$$

Remarkably, this constant is the **same** for any dynamical system undergoing a cascade of period-doubling bifurcations! This was rigorously proven by Collet, Eckmann and Lanford using renormalization group analysis [1]. In a certain sense, this special constant is "universal."

Lab Exercises

The goal of this project is to approximate the value of K by computing as many values of the sequence k_n as possible. Obviously this becomes challenging since the period grows exponentially in n. For example, to find c_{10} requires finding the c-value for which f_c has a super-attracting period 1024 orbit. This is not easy and will require some patience as well as some basic programming skills.

For **each** of the systems, $Q_c(x) = x^2 + c$, $F_\lambda(x) = \lambda x(1-x)$ and $S_b(x) = b \sin x$, do the following:

1. Compute (at least) the first seven parameter values $c_0, c_1, \dots c_6$. These should be found to a high degree of accuracy (at least 10 decimal places.) **List these values in a table.**
2. Compute (at least) the first five values k_0, k_1, k_2, k_3, k_4. If you have computed the c_n-values correctly and accurately, you should notice some convergence in your sequence. **List these values in a table.**
3. Estimate Feigenbaum's constant K. The more values of c_n, and consequently k_n, you are able to compute, the better your estimate will be.

Of the three dynamical systems, Q_c, F_λ, and S_b, which has a sequence k_n which seems to be converging the fastest to Feigenbaum's constant?

Some Hints: First, notice that two of the maps you are investigating are topologically conjugate, as proved on HW #4. This should help significantly with the computations for one family, as period-doubling bifurcations are preserved under conjugacy. (Why?)

Second, you should write a `while` loop in Maple (or some other program) to find the values of c_n to a high degree of accuracy. Recall that to obtain 20 decimal places of accuracy, type

`Digits := 20` . The suggested numerical algorithm to use is called the **Bisection Method**, although you may know of a different method to apply. The Bisection Method relies on some common mathematical sense and the Intermediate Value Theorem. To begin, use the *Orbit Diagram* java applet available from the Dynamical Systems and Technology Project website at Boston University to find a range for the particular c-value in question. This allows you to find an upper bound c_u and lower bound c_l for the pertinent c-value, $c_l < c < c_u$.

The algorithm proceeds as follows: First check whether the nth iterate of the critical point x_0 comes back below or above itself for your bounds. For example, suppose that $f^n_{c_u}(x_0) < x_0$ but $f^n_{c_l}(x_0) > x_0$. Set $c = (c_l + c_u)/2$ and compute $f^n_c(x_0)$. There are two possible cases, either $f^n_c(x_0) < x_0$ or $f^n_c(x_0) > x_0$. We explain the case $f^n_c(x_0) < x_0$. The other case is similar.

If $f^n_c(x_0) < x_0$, then the true value of c_n must lie between c_l (where the critical point returned above itself) and c (where the critical point returned below itself). The next choice of c, call it $\hat{c}$, should be $\hat{c} = (c + c_l)/2$. Keep c_l the same, but change the value of c_u to c. You have now zoomed in on the true value by cutting the target range in half (hence the "bisection method.") The old range was $[c_l, c_u]$ and the new one is half that size $[c_l, c]$.

Now compute $f^n_{\hat{c}}(x_0)$ and proceed as before, chopping your target range in half by determining whether the true value is in $[c_l, \hat{c}]$ or $[\hat{c}, c]$. Adjust your bounds accordingly, take the average of your new bounds and compute the n-th iterate of the critical point again. The nice thing about this algorithm is that it is easy to obtain error estimates for the true value of c_n. We can iterate this algorithm as many times as needed to obtain the largest degree of accuracy required. Be sure to choose your initial bounds carefully (and close together), otherwise you may re-discover a parameter value already on your list.

The above algorithm can be implemented as a `while` loop in Maple. You will need to insert a `for` loop inside the `while` loop to compute the iteration (see Computer Project #1). Recall that to skip to the next line in Maple without executing the command use the `shift` and `return` keys together. You will also need to insert an `if then` statement to distinguish between the two possible cases of being above or below the critical point. The syntax for these commands is fairly straight-forward. For example, your while loop might look like

```
while flag = 0 do

 . . .  commands in here . . .

end do;
```

You can exit the loop by setting `flag := 0` after you have executed the bisection algorithm sufficiently long. You don't have to use a "flag." This is just a suggestion. The syntax for an if-then statement looks something like

```
if (x0 < 0) then
... commands in here ...
else
... commands in here ...
end if;
```

For more information on these commands, you can type `?while` or `?if` .

Important: Please turn in a print out of the `while` loop you use in Maple (or some other program) to compute the special parameter values c_n.

References

[1] P. Collet, J.-P. Eckmann and O. E. Lanford. Universal properties of maps on an interval. *Comm. Math. Phys.* **76**, no. 3, 211-254, 1980.

[2] J. Gleick. *Chaos: Making a New Science*, Penguin Books, New York, NY, 1987.