Orbits and Classifications of Fixed Point Iterations via Graphical Analysis

Yutong Li

June 1, 2014

Contents

1 Abstract

In numerical analysis, fixed-point iteration is a method of computing fixed points of iterated functions. As the name suggests, a process is repeated until an answer is achieved. Iterative techniques are used to find roots of equations, solutions of linear and nonlinear systems of equations, and solutions of differential equations. In this project, we focus on iteration of regular continuous univariate functions and the classification of their respecting fixed points. This project consists of two components, one which classifies the fixed point through numerical methods and the other being a thorough interactive graphical analysis of such fixed points. The purpose of this project is to add an functionality to Sage regarding fixed point iterations.

2 Classifying Fixed Points

The process of finding fixed points and classifying them is quite simple. That said, we will still need a few definitions to serve as the basis of our coding. No proof will be provided for the following theorems as this is not the focus of the project.

Definition: Let $F : \mathbb{R} \to \mathbb{R}$ a continuous differentiable function and a point x_0 in the domain of f, the fixed point iteration is

$x_{n+1} = f(x_n), n = 0, 1, 2...$

which gives rise to the sequence $x_0, x_1, x_2, ...$ which is hoped to converge a point $x*$. We call this point the **fixed point** of f.

Theorem: A fixed point of a function $F(x)$ is a number x^* such that $x^* = F(x^*)$.

Definition: A fixed point of F is called

- *attracting* if $|F'(x^*)| < 1$.
- *repelling* if $|F'(x^*)| > 1$.
- *neutral* if $|F'(x^*)| = 1$.

The neutral fixed point is an ambiguous case. The fixed point itself could be attracting, repelling or neither, thus in order to figure out exactly how it behaves, we will require a different approach.

With the above ingredients, we can now implement them for the code of finding the fixed points and classifying their stability.

Examples: Consider the iterated functions $F_1(x) = 0.05x^2 + 0.25x + 1$ and "$F_2(x) = x - x^2$. We can find and classify the fixed points of these functions by calling the Sage function **fixed_points()** and inputting the $F(x)$ as the argument.

```
fixed_point(0.05*x**2+0.25*x+1)
1.47920271060385, eigenvalue = 0.397920271060385, attracting
13.5207972893961, eigenvalue = 1.60207972893961, repelling

fixed_point(x-x**2)
0, eigenvalue = 1, neutral
```

Through raw computation we have successfully found and classified the fixed points, but this also arises to several shortcomings. For example, we might be curious at how fast the iteration converges or diverges. We might be interested at how far away the initial point could be from the fixed point such that it still converges, i.e. *basin of attraction.* Moreover as mentioned above, the neutral fixed point is a tricky case; we do not know which side of the fixed point is converging or diverging just by looking at the eigenvalue. Thus, we require a more thorough analysis of the function, which leads us to graphical analysis.

3 Interactive Graphical Analysis

Graphical analysis is a tool to help visualize orbits of iterated functions, which provides a two-dimensional representation of orbits. It is extremely useful in analyzing univariate dynamical system whilst providing more detailed information than numerical methods in a more straightforward fashion. Below will be a detailed walkthrough of how my version of graphical analysis functions.

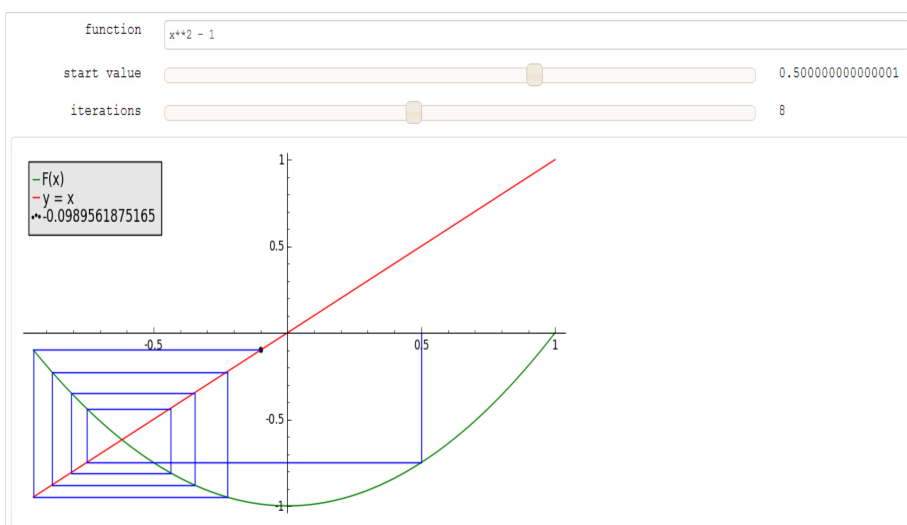

Figure 1(Initial Layout)

When the code is first executed, this will be the default appearance. This is what we call a **Cobweb graph** due to it's shape. The function $x^2 - 1$, starting value of 0.5 and 8 iterations are the default arguments whereas the green line is the graph of $F(x)$, red line indicating the graph of $y = x$, the blue line drawing the orbit, and the black point at the tip of the orbit represents the current value of the iteration. There is an interaction box where we can change the function, starting value, and the number of intended iterations

From the theorem of fixed points mentioned above, we know that x^* is a fixed point if $x^* = F(x^*)$. Therefore, the intersections of the two functions $y = F(x)$ and $y = x$ will be a fixed point. From the above graphical analysis, we see that with a staring of 0.5, the orbit spirals around the fixed point but is gradually trailing away from it. Thus, it is safe to say that this is a repelling fixed point.

Example: An orbit of an attracting fixed point at 1 of the function $F(x) = x^2 - 2x + 2$ with starting value 1.9 and 10 iterations.

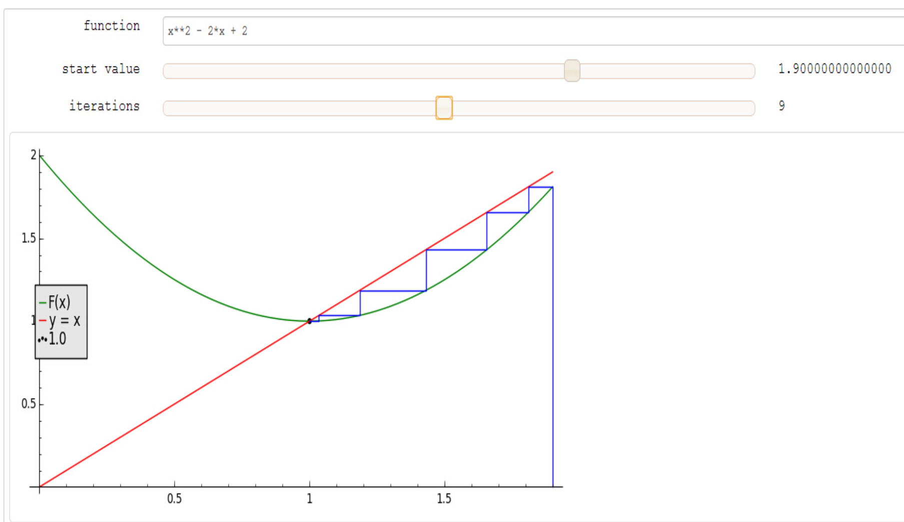

Figure 2(start value = 1.9, iterations = 10)

Example: How to determine the stability of a neutral fixed point. We use the same function $F(x) = x - x^2$ from the above section.

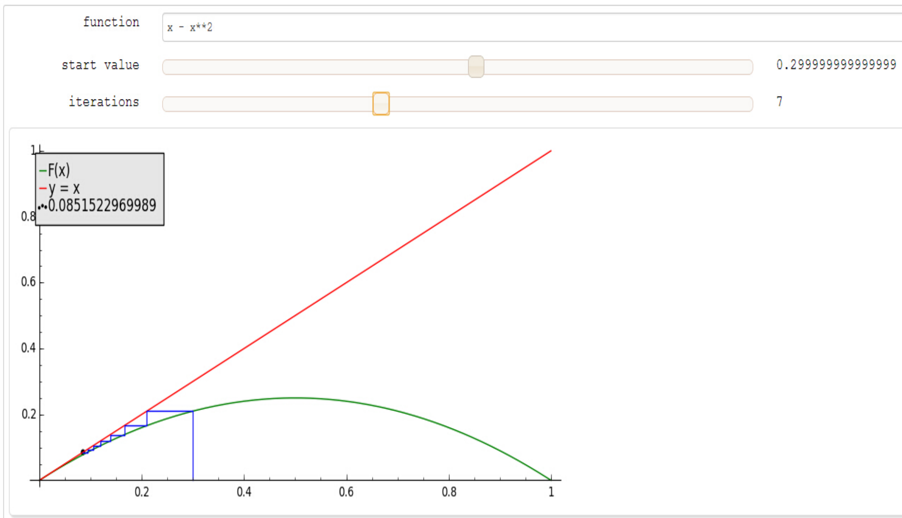

Figure 3 (start value = 0.3, iteration = 7)

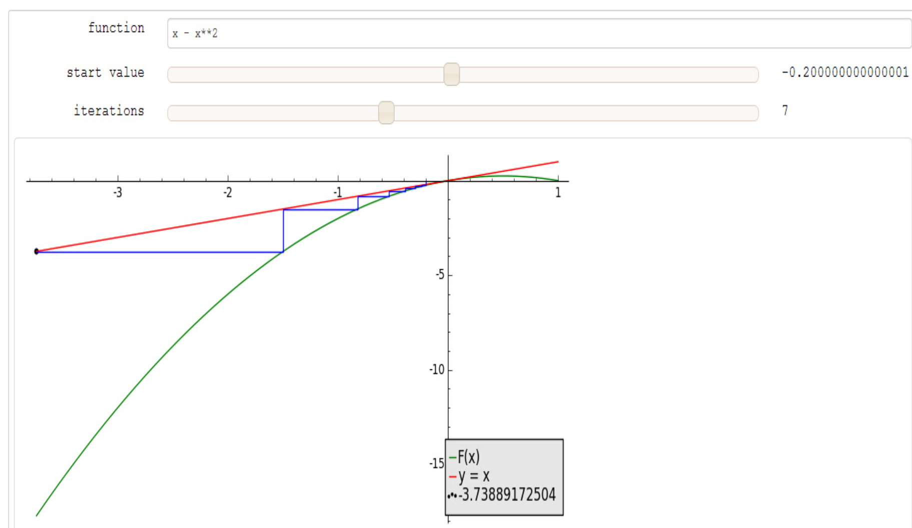

Figure 4 (start value = -0.2, iteration = 7)

From figure 3, we see that the orbit is converging to the fixed point of 0 with a starting value of 0.3. However, if we slide the initial point to a negative value of -0.2, the orbit zigzags away from 0. Hence, this neutral fixed point is in fact attracting on the right but repelling on the left.

Example: A fixed point may be attracting in a certain interval only, we call this the basin of attraction. Below is an illustration of how graphical analysis can help in determining the interval.

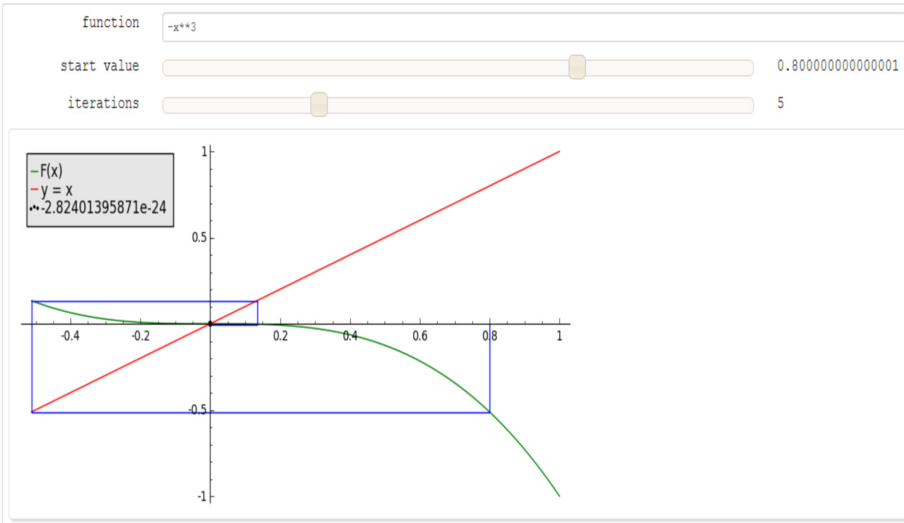

Figure 5 (start value = 0.8, iteration = 5)

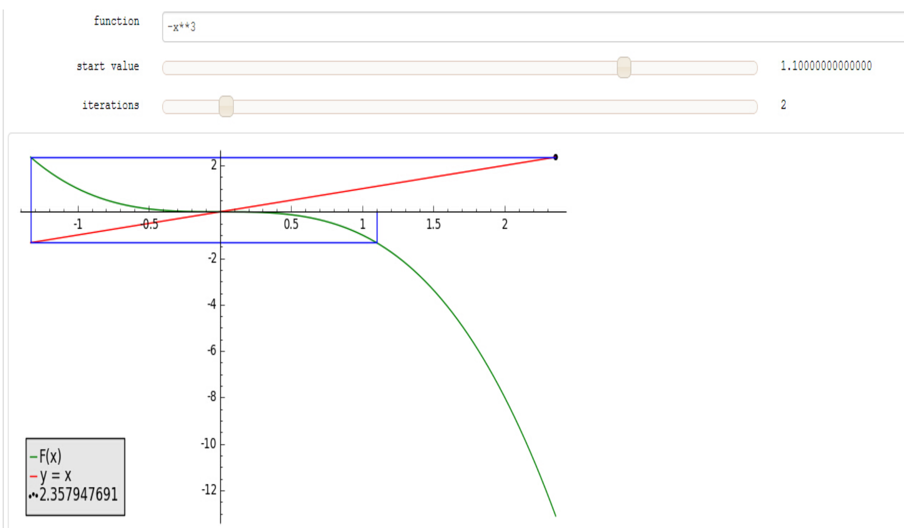

Figure 4 (start value = 1.1, iteration = 2)

The above function $F(x) = -x^3$ has one attracting fixed point at 0. However, this is only true if $|x_0| < 1$. If the starting point is larger than 1, then the iteration diverges. Thus, the classification for this fixed point would be an attracting fixed point with a basin of attraction [-1,1].

4 Conclusion

As you can see, graphical analysis easily comes into play when dealing with analyzing fixed point iterations. It is simple and straightforward to and provides detailed information about the system.

One improvement that can be done to this graph is to implement matplot.lib to adjust the legend and the white space to the right. Preferably, the legend could be taking up the white space on the right so that the whole appearance feels more full.

This tool can come in handy when doing studies in Numerical Analysis or Complex Analysis. For students that might be taking Math 464 or Math 428 in the future, this may help when you are checking your work.

5 Cited Sources

Prof. Kuennene, "GRAPHICAL ANALYSIS, AND ATTRACTING AND REPELLING FIXED POINTS6." (n.d.): n. pag. 23 Nov. 2012. Web. 1 June 2014.

Francois Monard, "Lecture 18, Notes on One-dimensional Dynamics", 12 Feb. 2014. Web. 1 June 2014

Marshall Hampton. "Interactive Dynamical System", http://wiki.sagemath.org/interact/dynsys, 18 April 2012.