Computational Physics
Homework 3

Yi-Hsuan Hsu

10/03/2014

1 Problem 1 to 4

$$x_{n+1} = f(x) = a \times x_n(1 - x_n) \tag{1}$$

Plot one-dimension map of $x_n]$, show the convergence behavior of a in interval $[0, 1]$, $[1, 3]$, $[3, 3.449]$. Give analytical derivation of convergence rule in each interval.

1.1 Theory

Prove the global attravting stable fixed point $X*$ in each interval.
Assume that the series converges to a fixed point x^*, rewrite equation (1)

$$n \to inf, x_{n+1} = x^* = f(x_n) = ax_n(1 - x_n) \tag{2}$$
$$= ax^*(1 - x^*) \tag{3}$$
$$x^* = 1 - \frac{1}{a}, or x^* = 0 \tag{4}$$

1.1.1 a = [0,1] , X* = 0

When $n \to \infty$, Taylor expansion of equation (1) X^* in first order

$$x^* + \epsilon_{n+1} = x_{n+1} = f(x^* + \epsilon_n) = f(x^*) + f'(x^*) * \epsilon_n + O(\epsilon^2) \tag{5}$$
$$\epsilon_{n+1} = f'(x^*)\epsilon_n = \lambda\epsilon_n \tag{6}$$

Where X^* is assuming convergent to 0, ϵ is little displacement in every iteration. To show the stability of attracting point. If the $multiplier |\lambda| < 1$, then ϵ would converge to 0.

$$0 < |\lambda| = a < 1 \tag{7}$$

Therefore, when $0 \leq a < 1$, the iteration converged stably to $X* = 0$.

When $a = 1$, we expand Taylor series in second order, that is

$$x^* + \epsilon_{n+1} = f(x^*) + f'(x^*) * \epsilon_n + f'(x^*) \tag{8}$$

$$\epsilon_{n+1} = \epsilon_n(1 - \epsilon_n) \tag{9}$$

Since ϵ is a small displacement, RHS alway smaller the displacement. Therefore, we may conclude that $a = 1$ converge to global stable fixed point $x^* = 0$.

1.1.2 a = [1,3] , X* = 1-1/a

The multiplier λ

$$|\lambda| = |f'(x^*)| = |a(1 - 2(1 - \frac{1}{a}))| < 1 \tag{10}$$

$$1 < a < 3 \tag{11}$$

1.1.3 a = [3.0,3.448489.....]

In this section, the problem is nothing solving quartic equation

$$x_{n+1}g(x_n) = f(f(x_n)) = af(x_n)(1 - f(x_n)) \tag{12}$$

when $n \to \infty$ expand the equation and take the long division of x and $x - 1 + \frac{1}{a}$ we can find out that

$$x^* = \frac{1+a}{2a} \pm \frac{\sqrt{a^2 + 2a - 3}}{2a^2} \tag{13}$$

plug the result into multiplier, and by chain rule

$$\lambda = |g(x)| = |f'(x_1)f'(x_2)| = |4 + 2a - a^2| < 1 \tag{14}$$

solve a, we have$3 < a < 1 + \sqrt{6} \sim 3.448498....$, prove series is stable in 2-circle.

1.1.4 find bifurcaiton point experimentally

See Table 1.

1.2 Algorithms

Mathematica build-in RecurrenceTable function which generates a list of values x for successive n based on solving the recurrence equations equation.

Manipulate function creat interactive manipulation plot for user to adjust favored parameter a and initial number X_0. This is especially useful to visualize stability of series given different initial value.

period	bifurcation point a
2	3.000 2
3.449	
4	3.544
8	3.564
16	3.568

Table 1: Logistic output after 10000 iteration, step-in algorithm

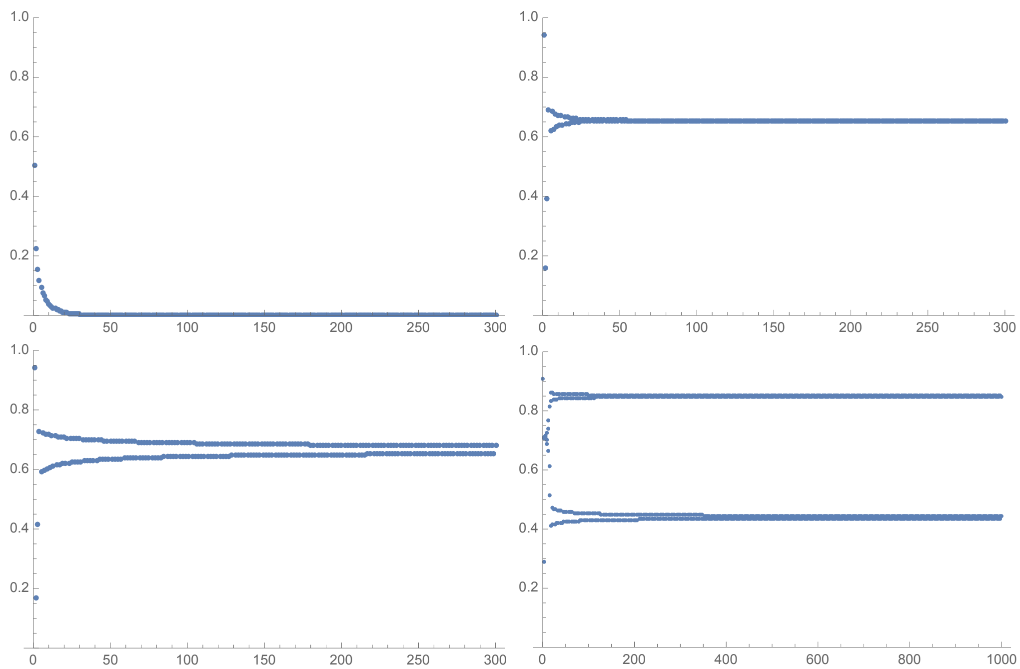

Figure 1: logistic map. Upper, left a=0.99, right a=2.9. Lower, left a=3.0, right a=3.4494.

1.3 Sample output

2 Problem 5 to 7

Find out bifurcation point for 2^m-circle.

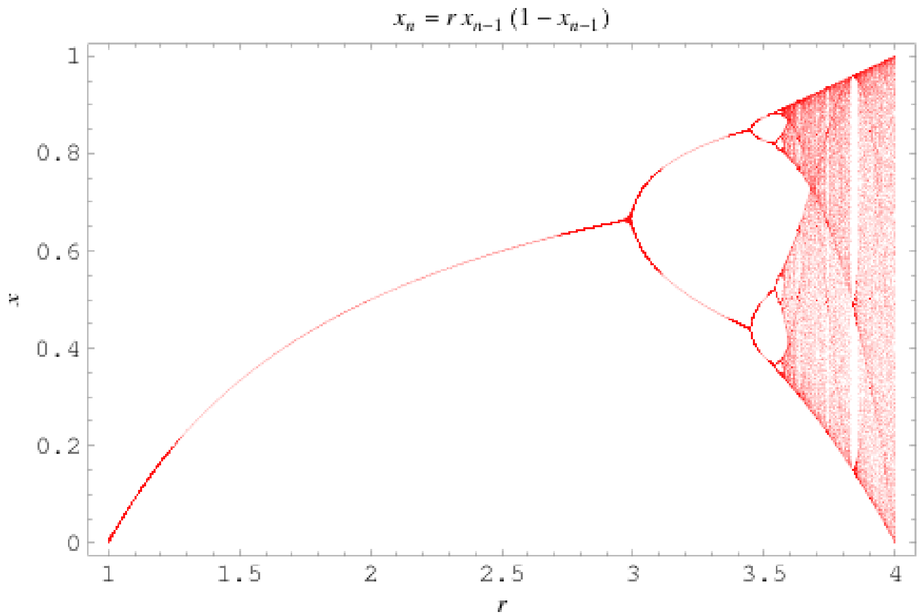

Figure 2: bifurcation diagram of the logistic map. Pic. from Mahtworld

2.1 Theory

From experiment above, we know that when r goes to certain point, x iterate to 2 number. By introducing self-iterated function $x = f^m(x) = f(f(f(......(x))))$ we are able to find out analytical solution for 2^m-order polynomial. In the other hand, we can use numerical approach to find the solution. By testing the multiplier

$$\lambda = |\frac{x}{dx} f^m(x)| = \Pi_{n=0}^{m} f'(f^m(x)) f'(f^{m-1}(x))...f'(x) < 1 \tag{15}$$

determined stability of x given certain a. More, let $\lambda \to -1$ we can find the bifurcation point accurately.

2.2 Algorithm

Giving m, random initial value x_0 and starting a, *logistic* returns $x_n(n = 2000)$ of 2^m-order iterated function. Use x_n as initial guess plug into $Newton's method$

$$x_{n+1} = x_n - \frac{f^m(x) - x}{f'^m(x) - 1} \tag{16}$$

stop at $x_{n+1} - x_n < 10^-13$.

Parameters	
Logistic step	1000
Newton's accuracy	10^{-12}
Total accuracy	$\epsilon < 10^{-13}$

Table 2: key parameters of algorithm

Multiplier calculated by equation demonstrated in (15). If $\lambda > -1$ then increasing a little step in parameter a, keep going on finding next λ and determined its stability. Once $\lambda < -1$, I adjust step size to find more digit. Repeat it on the next period, which $m = m+1$

More, I test automatically survey of a by bisection method. Giving a interval $[a_0, a_1]$, if $\lambda > -1$, then make the right end to bisect point keep search a to limit of computer digit.

2.3 Sample output

2.3.1 2-perid bifurcation

Table 2. shows the key parameter of algorithm. Bisection method are applied to find optimized bracket. The procedure do not take too much steps to find the root on the limit of computer digits. Figure 2 shows the program output. 47 steps are taken and make the result as close as possible to analytical answer 3.0.

2.3.2 4-period bifurcation

The approximation is great without changing the key parameters. The accuracy is exactly reach to computer's limit. That is, $\epsilon = |a_{exp} - (1 + \sqrt{6})| < 10^{-13}$. Compare to 2-period, 4-period only take 2 more step converging to this precession.

2.3.3 8-period bifurcation

Here comes challenge. For some reason the program did not converge to $\lambda = -1$. Sometime Newton's method is fail to find the approximate root. I let the program change initial value when Newton's fail, and raising the iteration of logistic map in order to plug in better initial guess for root-finding. It turns out bisection always going out of the expected range and fail to converge. Although I give a very small initial bracket, λ can not returns within 10^{-4} accuracy. Table 3 is key parameter in this exercise.

In addition, I try to use step-in algorithm instead of bisection. It seems that my λ is not smooth and continuous around fixed point. Figure 5 shows an example of step-in result. λ's

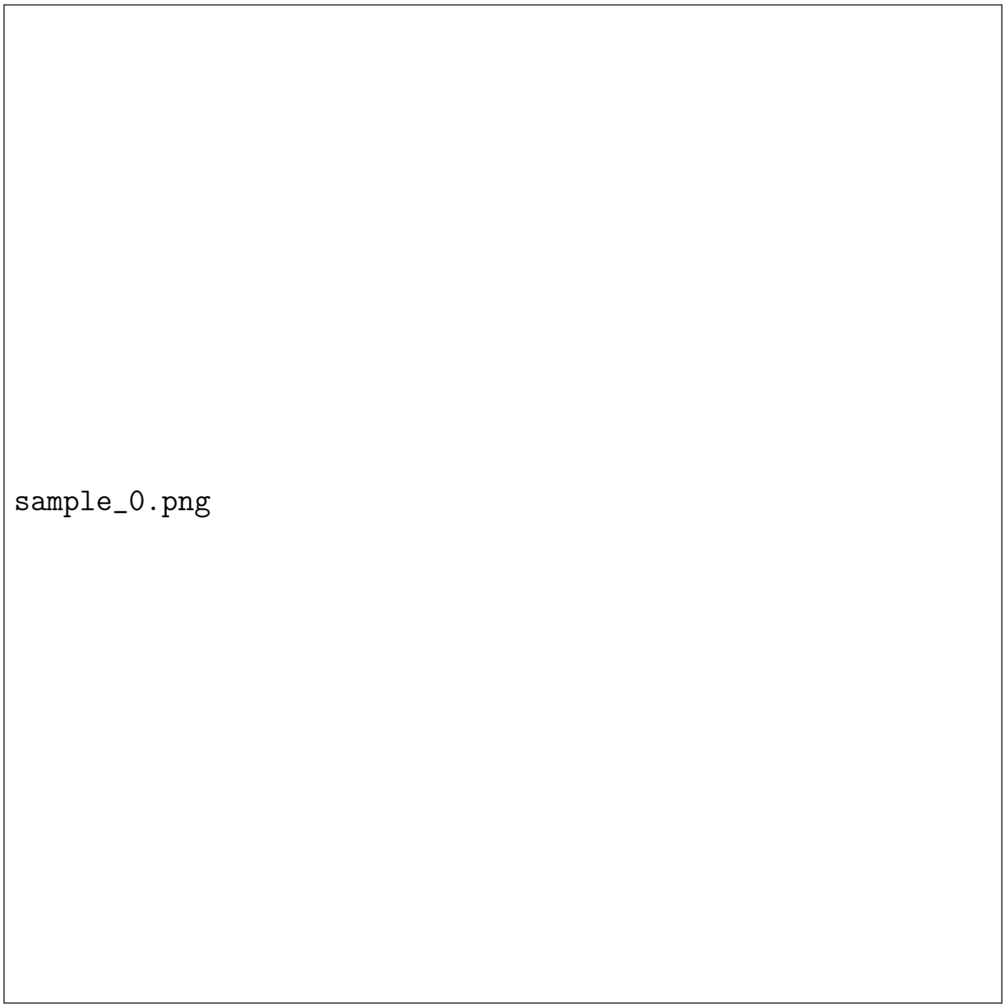

Figure 3: Numerical approach to 2-period-bifurcation point

behavior is very sensitive to initial value. It make difficulties to get a consistent result of bifurcation point.

In the further period, the program behave like this and I am not able to find any point. Therefore, I am fail to show Feigenbaum constant and r_∞.

2.3.4 f(x)=Sin(pi*X)

Simply change the definition of function and its derivative. I find the first four period bifurcation easily. Calculating *Feigenbaum* constant, I get sample_1.png

Figure 4: Numerical approach to 4-period-bifurcation point

$$\delta = \frac{0.8332663532356 - 0.719961841972}{0.8591690091416 - 0.8332663532346} = 4.37 \tag{17}$$

sample_2_2.png

Figure 5: Numerical approach to 8-period-bifurcation point

Parameters	
Logistic step	10000
Newton's accuracy	10^{-12}
Total accuracy	$\epsilon < 10^{-4}$

Table 3: key parameters of algorithm

period	bifurcation point a
2	0.7199616841972
4	0.8332663532346
8	0.8591690091416
16	0.8640801075000*

Table 4: Numerical approach to recursion sin function. First three are searching by bisection method, $\epsilon < 10^{-10}$. *Last one is using step-in method, accuracy$\epsilon < 10^{-5}$, although steps are much smaller than that.