Escape rate for the logistic map

Peter Andresén

1 Dynamical ζ-function and spectral determinant

We consider the logistic map

$$f(x) = Ax(1-x), \quad x \in [0,1]. \tag{1}$$

For $A < 4.0$ any initial point in the unit interval will remain there. Thus only for $A > 4.0$ is the escape rate $\gamma > 0$. The fixpoints x^* of eq. 1 are easily determined as well as their stability ($|f'(x^*)|$)

$$\begin{aligned} x_0^* &= 0.0, & |\Lambda_0| = A \\ x_1^* &= 1 - \frac{1}{A}, & |\Lambda_1| = A - 2. \end{aligned}$$

We now define a binary symbolic dynamics for each orbit

$$\{x_0, x_1, x_2, ..., x_n, ...\}$$

on the map. '0' is identified with x_n belonging to the left branch while '1' with the right branch of the map (Fig. 1). The irreducible periodic orbits (prime cycles) up to length 4 may then be identified as

$$\mathcal{P}_{n_p \leq 4} = \{0, 1, 01, 001, 011, 0001, 0011, 0111\}.$$

In order to determine the positions of these prime cycles the inverse of the logistic map may be utilized

$$f_{\pm}^{-1}(x) = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - \frac{4x}{A}}. \tag{2}$$

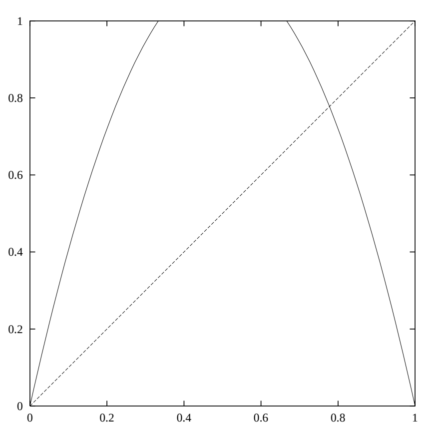


Figure 1: The logistic map $f(x)$

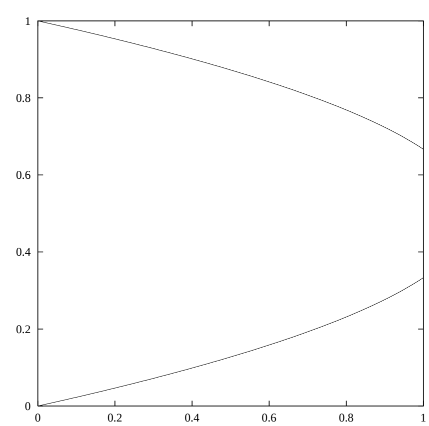


Figure 2: The inverse map $f^{-1}(x)$

The inverse f_-^{-1} and f_+^{-1} are then applied to an initial point whenever '0' resp. '1' appears in the symbolic sequence. Once satisfactory convergence of these positions have been obtained the stability of the cycle may be determined by multiplying the stability of each point. The stability of all prime cycles up to length 4 is listed below:

	$\lvert\Lambda_{cycle}\rvert$	
Cycle	$A = 4.5$	$A = 6.0$
0	4.5000	6.0000
1	2.5000	4.0000
01	7.2500	20.0000
001	26.4425	114.9545
011	19.9425	82.9545
0001	111.5689	684.4241
0011	80.6875	485.0937
0111	47.6811	328.6696

Dynamical ζ-function

The escape rate may now be determined from the leading zero of the dynamical ζ-function

$$1/\zeta(z) = \prod_p (1 - t_p), \qquad t_p = \frac{z^{n_p}}{|\Lambda_p|}. \tag{3}$$

Rewriting the above expression into a power-series

$$1/\zeta(z) = 1 - \sum_i \hat{c}_i z^i \tag{4}$$

where

$$\begin{aligned}
\hat{c}_1 &= \frac{1}{|\Lambda_0|} + \frac{1}{|\Lambda_1|} \\
\hat{c}_2 &= \frac{1}{|\Lambda_{01}|} - \frac{1}{|\Lambda_0 \Lambda_1|} \\
\hat{c}_3 &= \frac{1}{|\Lambda_{001}|} - \frac{1}{|\Lambda_0 \Lambda_{01}|} + \frac{1}{|\Lambda_{011}|} - \frac{1}{|\Lambda_1 \Lambda_{01}|} \\
\hat{c}_4 &= \frac{1}{|\Lambda_{0001}|} - \frac{1}{|\Lambda_0 \Lambda_{001}|} + \frac{1}{|\Lambda_{0011}|} - \frac{1}{|\Lambda_0 \Lambda_{011}|} - \frac{1}{|\Lambda_1 \Lambda_{001}|} + \frac{1}{|\Lambda_0 \Lambda_1 \Lambda_{01}|} \\
&+ \frac{1}{|\Lambda_{0111}|} - \frac{1}{|\Lambda_1 \Lambda_{011}|}
\end{aligned}$$

This yields the following coefficients:

	$\hat{c}_n$	
n	$A = 4.5$	$A = 6.0$
1	0.62222222	0.41666667
2	0.04904215	0.00833333
3	0.00213844	-7.94460×10^{-5}
4	-0.00014227	9.89314×10^{-7}

The escape rate γ may now be estimated by the logarithm of the leading zero z_0 of eq. 4. Including terms up to order n yields the following estimates:

	Leading zeros and escape rate			
	$A = 4.5$		$A = 6.0$	
n	z_0	γ	z_0	γ
1	1.607143	0.474458	2.400000	0.875469
2	1.443020	0.366738	2.294688	0.830597
3	1.434746	0.360988	2.296804	0.831519
4	1.435525	0.361531	2.296743	0.831492

Spectral determinant

Apparently the correct method to calculate the escape rate is to use the spectral determinant (here we use the same weights t_p as before)

$$det(1 - z\mathcal{L}) = \prod_p \prod_k \left(1 - \frac{t_p}{\Lambda_p^k}\right), \qquad t_p = \frac{z^{n_p}}{|\Lambda_p|}. \tag{5}$$

This expression may be rewritten by taking the logarithm, expanding the logarithm and summing the geometrical series;

$$\begin{aligned}
\log\left(det(1 - z\mathcal{L})\right) &= \sum_p \sum_k \log\left(1 - \frac{t_p}{\Lambda_p^k}\right) \\
&= -\sum_p \sum_k \sum_r \frac{t_p^r}{r\Lambda_p^{kr}} \\
&= -\sum_p \sum_r \frac{z^{n_p r}}{r|\Lambda_p^r|(1 - \Lambda_p^{-r})} \\
&= -\sum_p \sum_r B_p(r) z^{n_p r}, \qquad\qquad B_p(r) = \frac{1}{r|\Lambda_p^r|(1 - \Lambda_p^{-r})}
\end{aligned}$$

Writing $\log\left(det(1 - z\mathcal{L})\right)$ as a power series

$$\log\left(det(1 - z\mathcal{L})\right) = -\sum_i b_i z^i$$

we obtain to 4th order

$$\begin{aligned}
b_1 &= B_0(1) + B_1(1) \\
b_2 &= B_{01}(1) + B_0(2) + B_1(2) \\
b_3 &= B_{001}(1) + B_{011}(1) + B_0(3) + B_1(3) \\
b_4 &= B_{0001}(1) + B_{0011}(1) + B_{0111}(1) + B_{01}(2) + B_0(4) + B_1(4)
\end{aligned}$$

Now in order to determine the coefficients Q_i of the spectral determinant we must solve the following

$$det(1 - z\mathcal{L}) = 1 - \sum_i Q_i z^i = \exp(-\sum_i b_i z^i) \tag{6}$$

This yields the following relations between the Q_is and the b_is (expand the exponential)

$$\begin{aligned}
Q_1 &= b_1 \\
Q_2 &= b_2 - \frac{b_1^2}{2} \\
Q_3 &= b_3 - b_1 b_2 + \frac{b_1^3}{6} \\
Q_3 &= b_4 - b_1 b_3 - \frac{b_2^2}{2} + \frac{b_2 b_1^2}{2} - \frac{b_1^4}{24}
\end{aligned}$$

As the stability of the prime cycles has been calculated earlier, the coefficients of the spectral determinant may easily be determined to be

Q_n		
n	$A = 4.5$	$A = 6.0$
1	0.57142857	0.40000000
2	0.07915894	0.01523810
3	0.00555000	7.59784×10^{-5}
4	- 0.00004270	-4.53109×10^{-9}

Again the escape rate may be determined by the logarithm of the leading zero to the spectral determinant, which yields

Leading zeros and escape rate				
	$A = 4.5$		$A = 6.0$	
n	z_0	γ	z_0	γ
1	1.750000	0.559616	2.500000	0.916291
2	1.456235	0.375854	2.298703	0.832345
3	1.435713	0.361661	2.296745	0.831493
4	1.435495	0.361510	2.296745	0.831493

We observe that the spectral determinant, as promised, converges faster than the dynamical ζ-function.

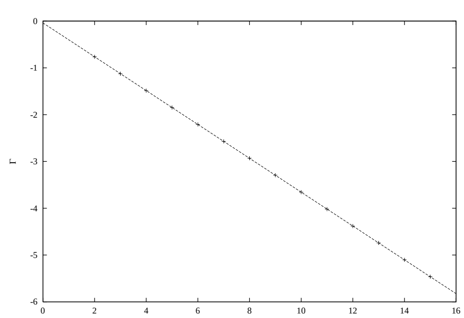
2 Numerical calculation

Numerically the escape rate may be estimated by measuring the fraction of initial conditions (here 10^7 randomly distriuted) after n iterations of the map. This fraction decays exponentially

$$\Gamma(n) \sim e^{-\gamma n}$$

and thus the escape rate γ may be determined by plotting $\ln\Gamma(n)$ vs. n. Fitting a line to the numerially obtained results (see below) for the interval $n \in [2, 10]$ yields

$$A = 4.5: \qquad \gamma = 0.36155 \pm 0.00008$$
$$A = 6.0: \qquad \gamma = 0.83188 \pm 0.00060$$



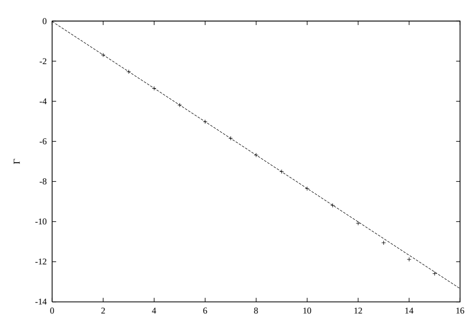


Figure 3: Numerical estimates of the escape rate for the logistic map and the fit by linear regression on the interval $[2, 10]$ ($A = 4.5$ resp. $A = 6.0$)

A Program prime.c

The program used on my 'pocket calculator' to determine the position of the prime cycles, their stability, the coefficients of the dynamical ζ-function, the spectral determinant and the escape rate for the logistic map is listed here:

```
#include<stdlib.h>
#include<stdio.h>
#include<math.h>

/***********************************************************
   Program to calculate escape rate for the logistic map
              f(x)=A*x*(1-x)
   including prime cycles up to length 4.
   Cycles are defined in the file prime.dat via length and
   binary symbolic sequence.
 **********************************************************/

#define  sgn(Q)        ( Q<0 ? -1.0 : 1.0 )
#define  A             4.5

int main(){

 int i,j,k,n,m[4],max;

 double x[5],b[4],c[4],q[4],lambda[8],f,df,y,z;

 FILE *input;

 x[0]=0.5;
 input=fopen("prime.dat","r");
 printf("\n");

 for(k=0;k<8;k++){
   fscanf(input,"%d",&n);
   for(j=0;j<n;j++)
     fscanf(input,"%d",&m[j]);

   for(i=0;i<100;i++){
     for(j=0;j<n;j++)
       x[j+1]=0.5-(1-2*m[j])*0.5*sqrt(1.0-4*x[j]/(double)A);
     x[0]=x[n];
   }

   lambda[k]=1.0;
   for(j=0;j<n;j++)
     lambda[k]*=(A*(1-2*x[j]));

   printf("Eigenvalue for cycle ");
   for(j=0;j<n;j++) printf("%1d",m[j]);
   for(j=0;j<4-n;j++) printf(" ");
   printf(" : %6.4lf ( ",fabs(lambda[k]));
   for(j=0;j<n;j++)
```

```
    printf("%5.3lf ",x[j]);
  printf(")\n");
}

fclose(input);

/* Spectral determinant terms */

b[0] = 1/(fabs(lambda[0])-sgn(lambda[0]))
      +1/(fabs(lambda[1])-sgn(lambda[1]));
b[1] = 1/(fabs(lambda[2])-sgn(lambda[2]))
      +1/(2*(pow(fabs(lambda[0]),2)-pow(sgn(lambda[0]),2)))
      +1/(2*(pow(fabs(lambda[1]),2)-pow(sgn(lambda[1]),2)));
b[2] = 1/(fabs(lambda[3])-sgn(lambda[3]))
      +1/(fabs(lambda[4])-sgn(lambda[4]))
      +1/(3*(pow(fabs(lambda[0]),3)-pow(sgn(lambda[0]),3)))
      +1/(3*(pow(fabs(lambda[1]),3)-pow(sgn(lambda[1]),3)));
b[3] = 1/(fabs(lambda[5])-sgn(lambda[5]))
      +1/(fabs(lambda[6])-sgn(lambda[6]))
      +1/(fabs(lambda[7])-sgn(lambda[7]))
      +1/(2*(pow(fabs(lambda[2]),2)-pow(sgn(lambda[2]),2)))
      +1/(4*(pow(fabs(lambda[0]),4)-pow(sgn(lambda[0]),4)))
      +1/(4*(pow(fabs(lambda[1]),4)-pow(sgn(lambda[1]),4)));

q[0] = b[0];
q[1] = b[1]-pow(b[0],2)/2.0;
q[2] = b[2]-b[0]*b[1]+pow(b[0],3)/6.0;
q[3] = b[3]-b[0]*b[2]-pow(b[1],2)/2.0+b[1]*pow(b[0],2)/2.0-pow(b[0],4)/24.0;

/* Cycle expansion terms */

for(k=0;k<8;k++)
  lambda[k]=fabs(lambda[k]);

c[0]  = 1.0/lambda[0]+1.0/lambda[1];
c[1]  = 1.0/lambda[2]-1.0/(lambda[0]*lambda[1]);
c[2]  = 1.0/lambda[3]-1.0/(lambda[0]*lambda[2])+1.0/lambda[4]-1.0/(lambda[1]*lambda[2]);
c[3]  = 1.0/lambda[5]-1.0/(lambda[0]*lambda[3]);
c[3] += 1.0/lambda[6]-1.0/(lambda[0]*lambda[4])-1.0/(lambda[1]*lambda[3]);
c[3] += 1.0/(lambda[0]*lambda[1]*lambda[2]);
c[3] += 1.0/lambda[7]-1.0/(lambda[1]*lambda[4]);

printf("\n");
for(j=0;j<n;j++)
  printf("%1d order expansion term : %14.12lf %16.14lf\n",j,c[j],q[j]);
printf("\n ------------------------------------ \n\n");

for(j=0;j<4;j++){
  f=1.0;
  y=1/c[0];
  while(fabs(f)>0.00001){
    f=0.0;
    df=0.0;
    for(i=1;i<j+1;i++)
```

```
      df-=(i+1)*c[i]*exp(log(y)*i);
    df-=c[0];
    for(i=0;i<j+1;i++)
      f-=c[i]*exp(log(y)*(i+1));
    f+=1.0;
    y=y-f/df;
  }

  f=1.0;
  z=1/q[0];
  while(fabs(f)>0.00001){
    f=0.0;
    df=0.0;
    for(i=1;i<j+1;i++)
      df-=(i+1)*q[i]*exp(log(z)*i);
    df-=q[0];
    for(i=0;i<j+1;i++)
      f-=q[i]*exp(log(z)*(i+1));
    f+=1.0;
    z=z-f/df;
  }

  printf(" Escape rate to %1d order: %lf (%lf)    %lf (%lf)\n",j,log(y),y,log(z),z);
 }
 printf("\n ------------------------------------ \n");
 return 0;
}
```