

Periodic orbit theory of chaos

About the lectures

These are notes for four lectures to be given in the school of mathematics of the university of Bristol in May 2015. I am grateful to the University of Bristol and especially Jonathan Robbins for the warm hospitality I received during my stay in Bristol, and to the Leverhulme Trust for generous financial support.

It is well known that many dynamical systems exhibit chaotic behavior, which renders their long-time dynamics inherently unpredictable. For chaotic systems it is therefore reasonable to abandon the idea of describing their dynamics exactly, and focus instead on probabilistic description, even when the dynamical laws are deterministic.

However, deriving the statistical distribution from the dynamical laws is a difficult problem that has been solved only in special cases. The goal of these lectures is to introduce *periodic orbit theory*, the strongest method that has been found so far to approach this problem.

The limited scope of the lectures allows me to introduce only the fundamental ideas of the method, along with the necessary basic ideas from dynamical systems theory (and only those). The level of rigour is dictated by my scientific background, and is similar to what one finds in theoretical physics literature.

The main source for the material presented here is the free webbook "*Chaos: Classical and Quantum*" by P. Cvitanovic et al., available at chaosbook.org. Other sources will be published together with these notes and other course material on the course website TBD.

Course plan

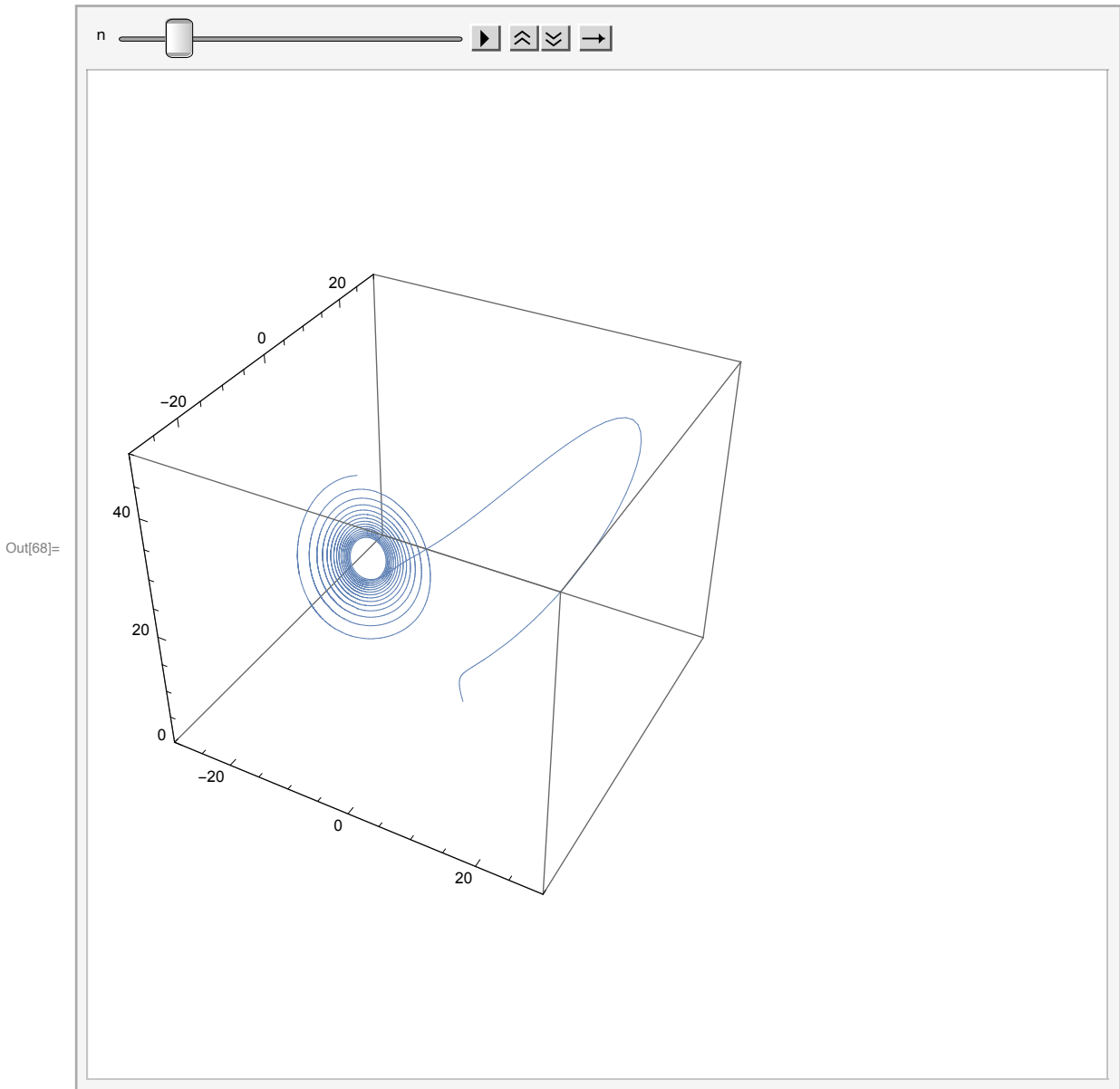
I The Lorenz system: Introduction to chaos

The Lorenz system consists of 3 ordinary differential equations for the variables x , y , z . It was put forward by Lorenz in 1963 as a truncation of climate model equations exhibiting aperiodic dynamics.

$$\begin{aligned}x'(t) &= \sigma(y(t) - x(t)) \\y'(t) &= r x(t) - x(t) z(t) - y(t) \\z'(t) &= x(t) y(t) - b z(t)\end{aligned}\tag{1}$$

σ , r , and b are real constant parameters. The values assumed by Lorenz $\sigma = 10$, $b = \frac{8}{3}$, and $r = 28$, lead to strongly chaotic dynamics, and are also the most studied. We will use these parameter values exclusively here.

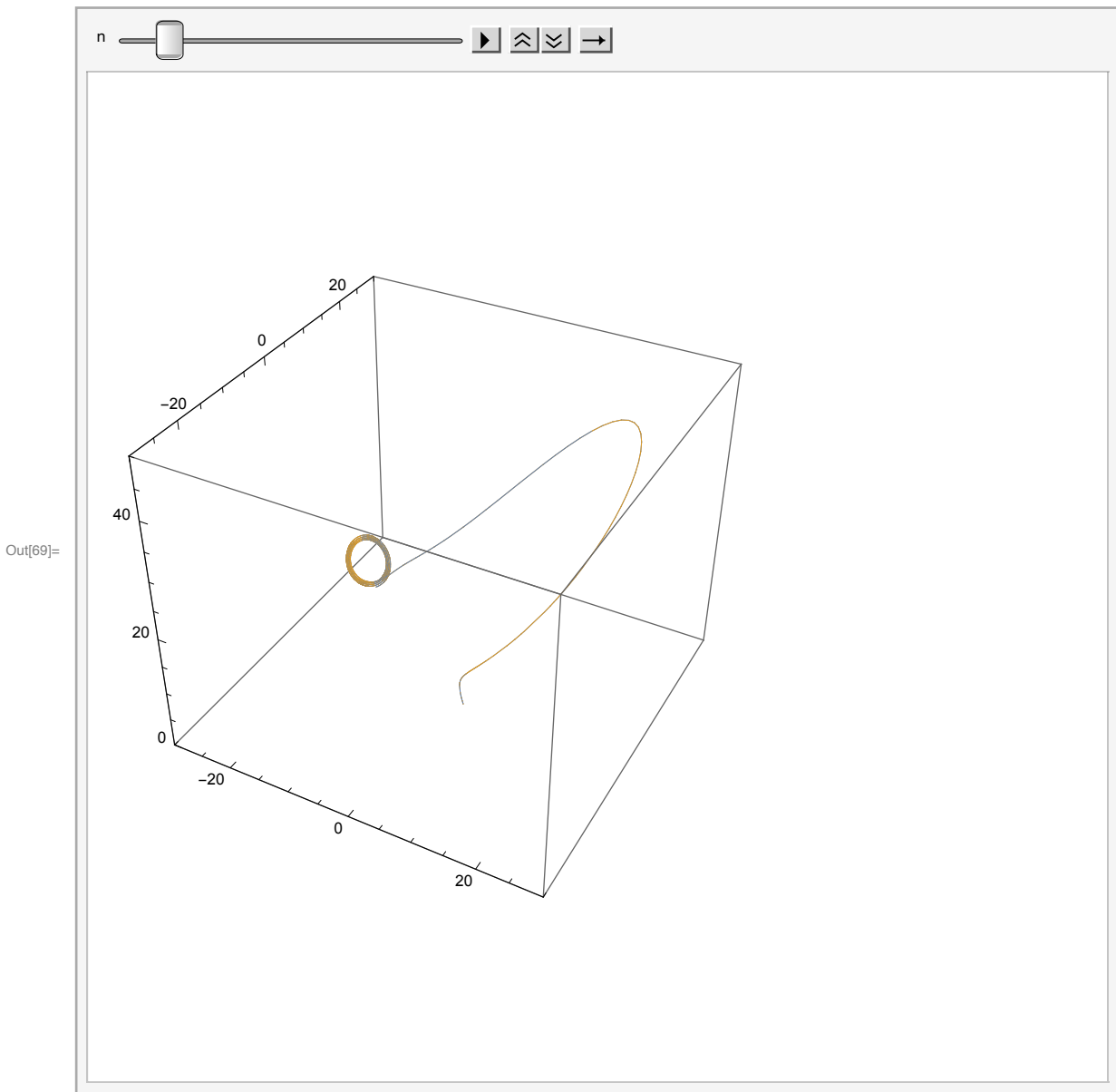
A typical orbit, that is, a solution $(x(t), y(t), z(t))$ of the Lorenz equations looks like this:



The dynamics can be roughly described as follows: After an initial transient, the orbit is attracted to a butterfly shaped region in the three-dimensional state space. Each wing of the butterfly looks like a slightly curved membrane, but it has a zero-measure transverse component—the attractor is a fractal with dimension slightly larger than 2.

The dynamics in each wing follows an outward spiral—clockwise in the left wing and counter-clockwise in the right, when viewed from above—and leaves the wing when the spiral radius increases beyond a certain radius, landing at some point in the other wing. The process then repeats itself. The wings are related by the exact symmetry $(x, y, z) \rightarrow (-x, -y, z)$ of the Lorenz equations.

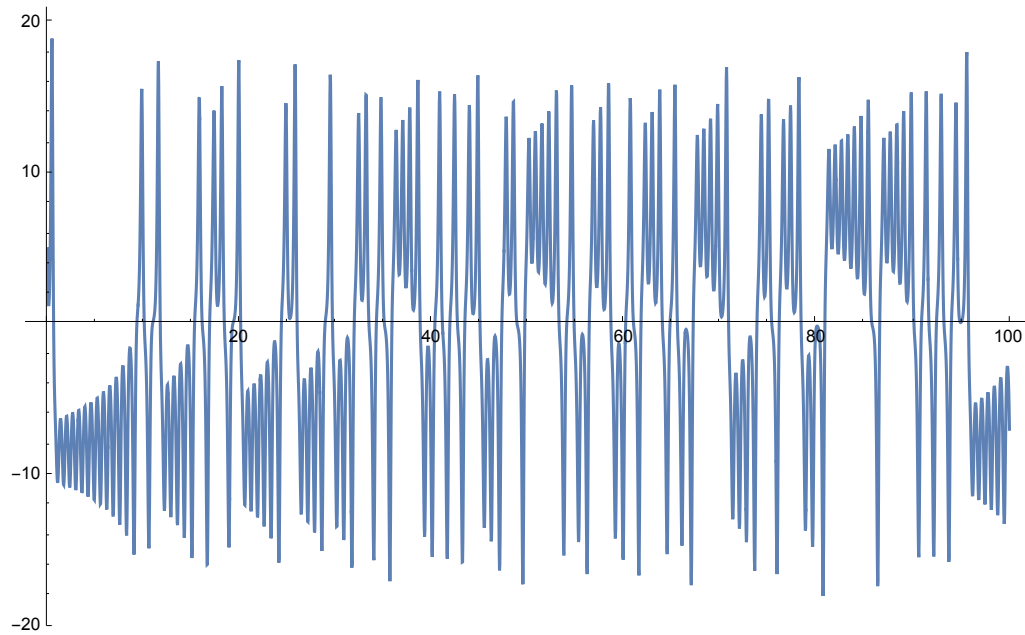
Although the orbits converge in directions transverse to the attractor, they *diverge* tangent to it, a phenomenon called sensitive dependence on initial conditions, or ‘the butterfly effect.’



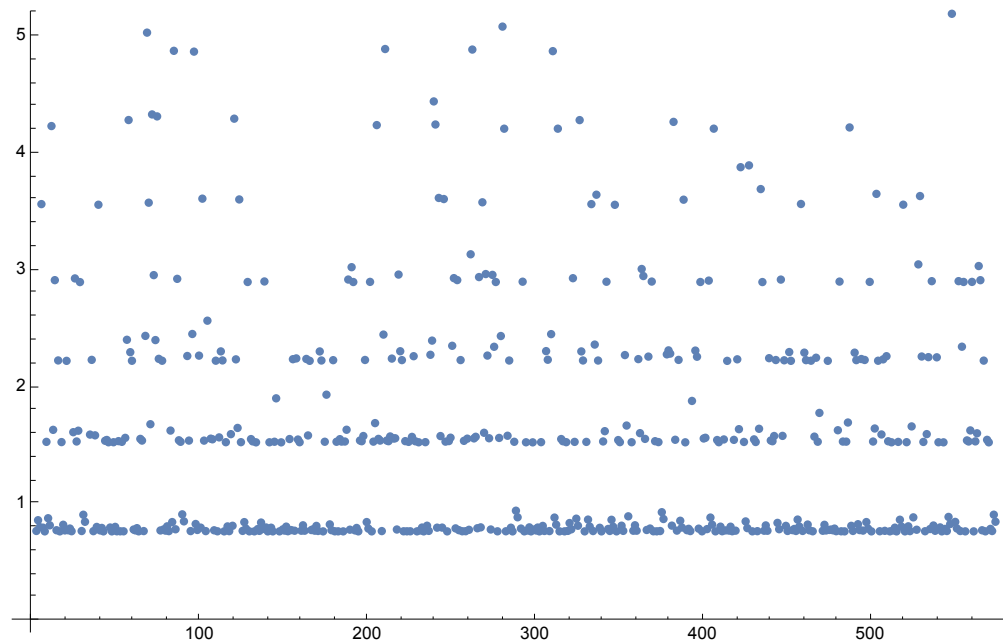
This means that two orbits with small initial separation $\delta(0)$ (in the example above $\delta(0) = 0.01$ in the z direction), grows exponentially $\delta(t) = e^{\lambda t} \delta(0)$ as long as δ remains small, with a *Lyapunov exponent* $\lambda > 0$. At some point, depending logarithmically on the smallness of the initial perturbation, the orbit separation becomes large enough that the orbits split into different wings, and then any remaining correlation between them is lost.

The simultaneous stretching and compression mixes the states on the attractor very efficiently, so that a small piece of it is quickly spread over the entire butterfly, although its volume shrinks very fast, the dynamics being dissipative. The stretching and folding, together with sensitive dependence on initial values are the fundamental properties of chaotic dynamics.

The chaotic properties of the dynamics have an important consequence. In a coarse-grained description, the Lorenz dynamics consists of transition between two states, left, and right which occur after a sequence of switching times t_1, t_2, \dots . The state of the system can be labelled for example by the sign of $x + y$.



A typical sequence of switching times looks like this



Since the underlying dynamics is deterministic, one could have guessed that the switching time sequence obeys well-defined rules, but this is not the case. Sensitive dependence on initial values implies that there is no way to predict the future t_n given the past beyond a few steps because the initial value is only known to a finite precision. The inevitable initial ignorance makes the dynamics effectively random. The purpose of these lectures is to understand this random process.

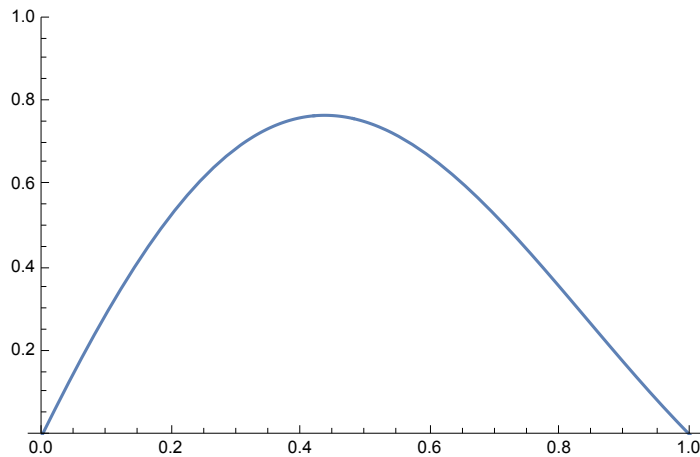
2 Unimodal maps

2.1 Discrete dynamical systems, the logistic map

The essence of the Lorenz dynamics is captured by the *return map*, which describes the discrete steps of climbing on the outward spirals and wing switching. Since the Lorenz attractor is almost confined to a two-dimensional manifold, the discrete dynamics takes place on a one-dimensional

manifold. The stretch and fold action implies that the mapping has at least one turning point.

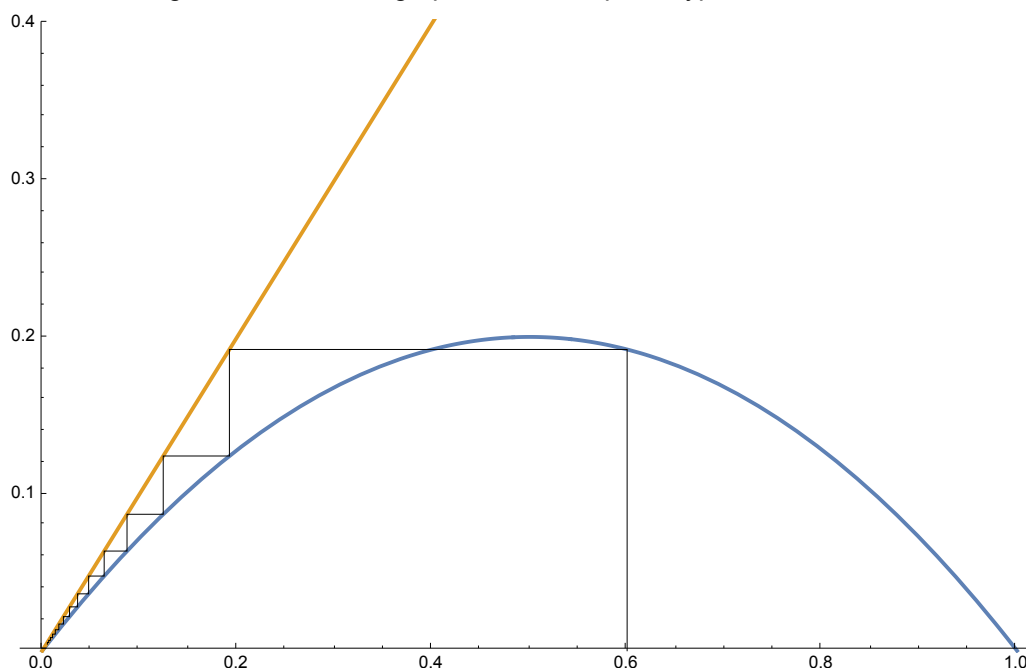
Rather than attempt a direct description of the Lorenz dynamics, we will look at the simplest type of dynamical systems with these properties: *unimodal* maps, that is, piecewise smooth maps $f(x)$ of the unit interval to itself with a single extremum, such as



The dynamics of unimodal maps will be mostly illustrated by the logistic map family $I_r(x) = r x(1 - x)$, parameterized by r . For $0 < r \leq 4$ it maps the unit interval into itself.

2.2 Fixed points, bifurcations, periodic orbits

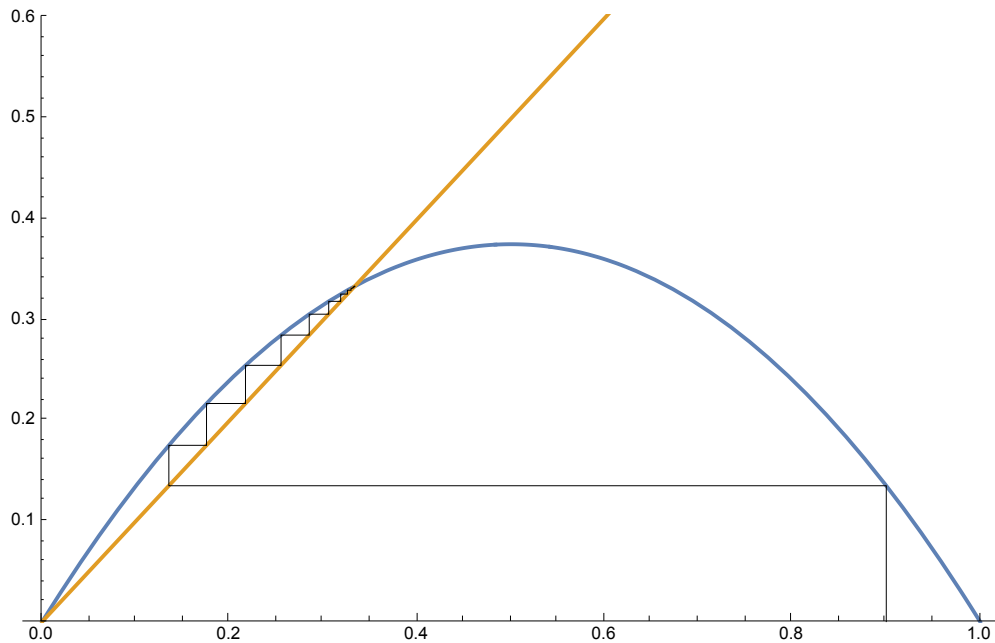
A simple way to visualize the dynamics of one-dimensional maps is to draw the graph of the map overlaid with the graph of the identity function. An orbit is then obtained by vertical and horizontal lines bouncing between the two graphs. For example, a typical orbit of $I_{0.8}$ looks like



Evidently, the orbit tends to zero, $\lim_{n \rightarrow \infty} I_r^n(x) = 0$, I_r^n being the n th iterate of I_r , for any initial x . In other words, $x = 0$ is a *globally attracting* fixed point. It follows that it is a (linearly) *stable* fixed point, $|I_r'(0)| < 1$. Fixed points are visible in orbit plots as intersections between the map and identity graphs, and when f is increasing they are linearly stable iff the map slope is smaller than the unit slope.

$x = 0$ is the only fixed point, and is globally attracting for all $0 < r < 1$, but when $r = 1$ there is a

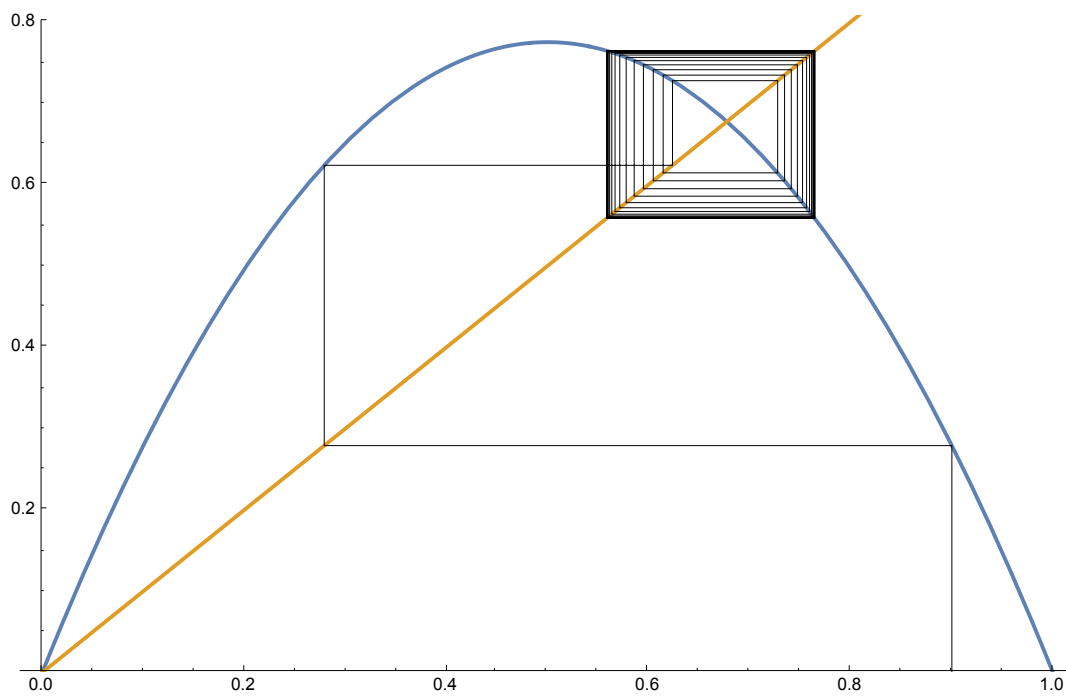
(transcritical) *bifurcation*, where it becomes unstable, and a new fixed point is created at $x_1 = 1 - \frac{1}{r}$. The new fixed point is stable and globally attracting for $1 < r < 3$. A typical orbit for $r = 1.5$ is



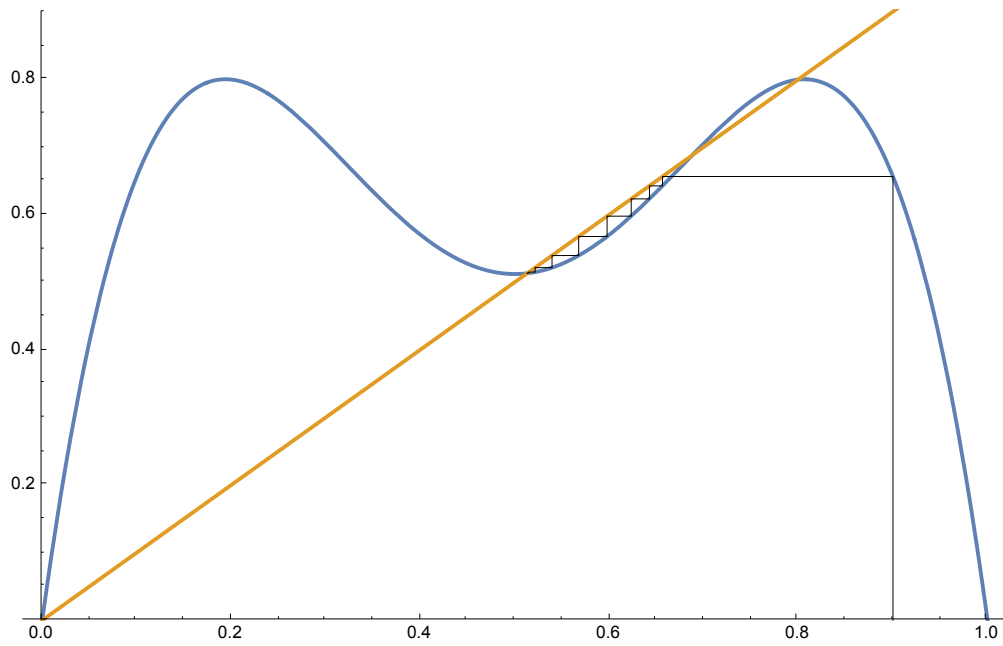
When $r = r_1 = 3$ another bifurcation occurs when the x_1 fixed point becomes unstable, but this time instead of a new fixed point, a period-2 *periodic orbit* is created $x_2^{(1,2)} = \frac{r+1 \pm \sqrt{(r-3)(r+1)}}{2r}$. The orbit is stable if for small δ

$$|f_r(f_r(\delta + x_2)) - x_2| < \delta \iff |f_r'(x_2^{(1)})f_r'(x_2^{(2)})| < 1$$

that is, for $3 < r < 1 + \sqrt{6} \approx 3.45$. A typical orbit for $r = 3.1$ is

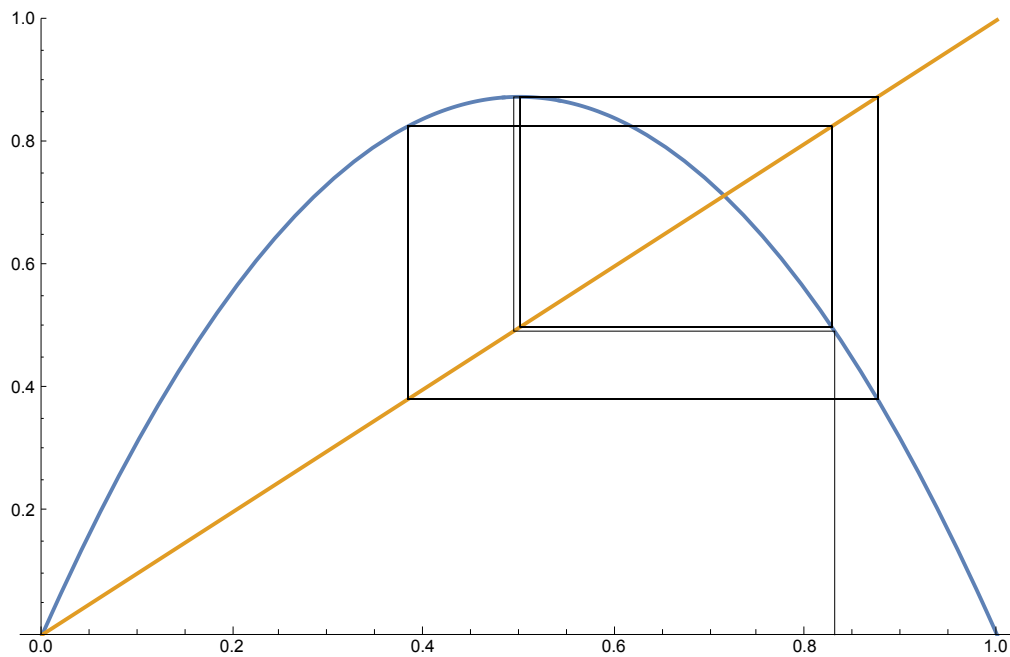


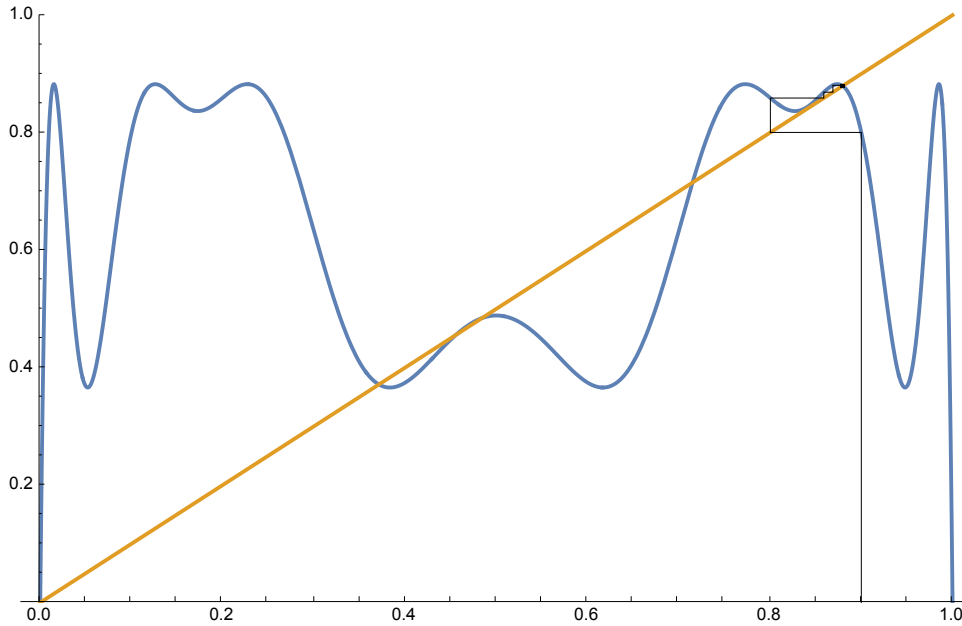
The period doubling bifurcation can be visualised by plotting the orbit of f_r^2 , the second iterate of f_r ,



The fixed points of f_r^2 consist of the unstable fixed points $0, x_1$ of f_r and the points of the period-2 orbit $x_2^{(1,2)}$, which are stable as fixed points of f_r^2 .

When $r = r_2 = 1 + \sqrt{6}$ a second period doubling bifurcation takes place: The period 2 orbit becomes unstable, and a stable period 4 orbit is created.



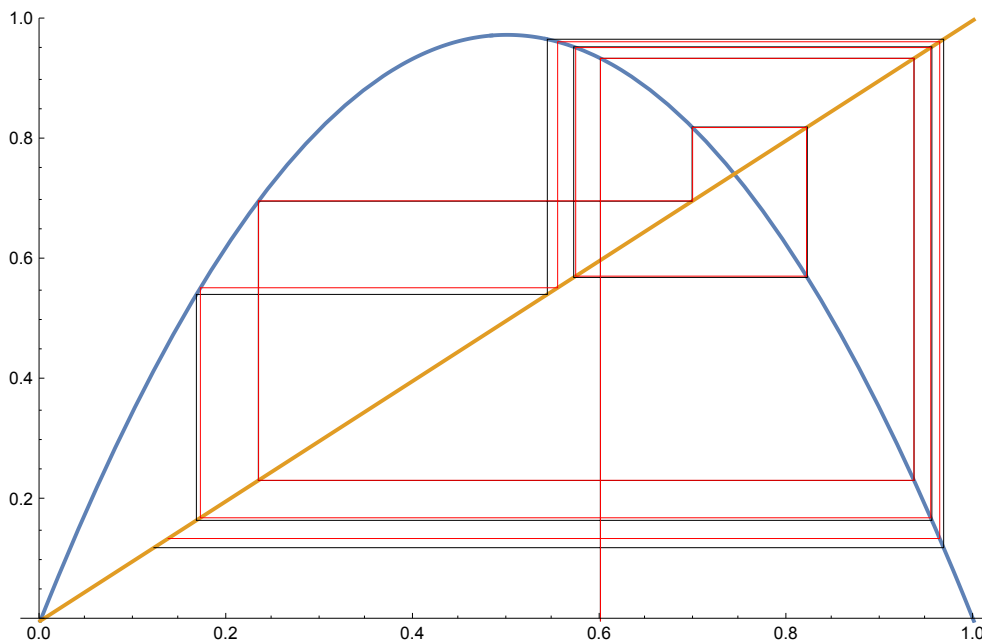


The period-4 orbit then becomes unstable at $r_3 \approx 3.544$ with a third period doubling bifurcation, and the process continues indefinitely with decreasing spacings between bifurcations. After each bifurcation, in which a stable 2^n -periodic orbit is created, the previous 2^k -periodic orbits, $k < n$, remain and are all unstable. It can be shown that unimodal maps have at most one stable periodic orbit.

2.3 Chaos, strange attractors, repellers

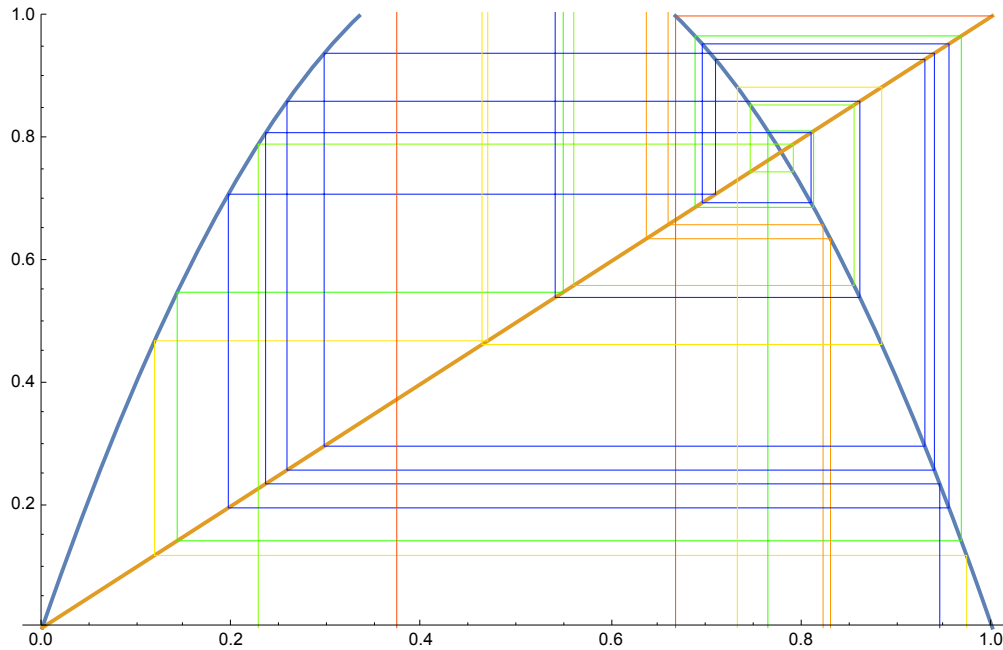
In the limit of the sequence of bifurcation $r = r_\infty = \lim_n r_n \approx 3.5699456$ the system has infinitely many periodic orbits but none of them is stable. Instead, a typical orbit is aperiodic and approaches a "strange attractor" \mathcal{A} , a (fractal) set of zero measure on which the dynamics. The orbit of a typical attractor point $x_{\mathcal{A}}$ and the set of periodic orbits are dense in the attractor, $\mathcal{A} = \overline{\bigcup_{n=k}^{\infty} I_r^n(x_{\mathcal{A}})} = \overline{\bigcup x_n}$.

When $r > r_\infty$ (for most r) the attractor has positive measure, and the dynamics on it has sensitive dependence on initial conditions. The sensitivity can be quantified by the Lyapunov exponent $\lambda = \lim_n \frac{1}{n} \sum_{i=1}^n \log(I_r'(I_r^{i-1}(x))) > 0$. It follows from this and the density properties that the dynamics is *chaotic*. λ is independent of initial x except for a zero-measure set, namely the periodic points.



When $r = 4$ the logistic map is conjugate to the piecewise linear symmetric tent map: $l_4 \circ g = g \circ t_2$, $t_r = \frac{r}{2}(1 - 2|x|)$ with the bijection $g(x) = \sin^2\left(\frac{\pi x}{2}\right)$. For both maps the image of the map and the attractor are the entire interval, and the dynamics is maximally chaotic, being conjugate to the Bernoulli shift map (see below).

When $r > 4$ the image of the $l_r([0, 1]) \not\subset [0, 1]$. The closure \mathcal{R} of the set of (unstable) periodic orbits is still an invariant set, but is not an attractor: Points arbitrarily close to the invariant set are mapped outside the unit interval in a finite number of iterations. Rather \mathcal{R} is a repeller, which means that it is stabilised by the inverse map l_r^{-1} (as a map on sets). The probability p_n that a randomly chosen initial point survives the dynamics after n steps decays exponentially: $p_n \underset{n \rightarrow \infty}{\sim} e^{-\gamma n}$. γ is the *escape rate* from the repeller.



(this is also true for some $r_\infty < r < 4$, except that repelled points are attracted to a stable periodic orbit with a period which is not a power of 2 instead of leaving the interval).

2.4 Invariant measures

In the chaotic regime the finite precision of initial conditions makes it impossible to calculate actual future orbits indefinitely with any fixed degree of precision. Instead one has to adopt a probabilistic approach. Starting from a probability measure μ_0 of initial conditions concentrated near an initial value x_i (say) we would like to know the later probability measure μ_n such that $\mu_n(A)$ is the probability that $f^n(x) \in A$.

It follows from the definition that $\mu_{n+1}(A) = \mu_n(f^{-1}(A))$. If we write the μ s in terms of densities

$\mu_n(A) = \int_A \rho_n(x) dx$ then

$$\rho_{n+1}(x) = \int \rho_n(\xi) \delta(x - f(\xi)) d\xi = \sum_{k, f(\xi_k)=x} \frac{\rho_n(\xi_k)}{|f'(\xi_k)|} \equiv \mathcal{L}\rho_n(x) \quad (2)$$

where \mathcal{L} is the evolution operator, also called Perron-Frobenius operator, and $\delta(x)$ is the Dirac delta 'function' (or distribution), defined by $\int h(x) \delta(x) dx = h(0)$ for sufficiently nice h .

For dynamics approaching an attractor one may expect the measure to tend asymptotically to a limit measure that is invariant under \mathcal{L} . Actually, there are many invariant measures. For example, any fixed point and periodic orbit carries a singular invariant measure $\frac{1}{p} \sum_{k=1}^p \delta(x - x_p^{(k)})$. We are interested in the *natural measure* that is the one arrived from smooth initial measures. For systems with escap-

ing orbits the natural measure is only conditionally invariant, that is $\mathcal{L}\rho = e^{-\gamma} \rho$ where γ is the escape rate as before.

3 Cycle expansion for the escape rate

The basic idea of cycle expansions is simple: Since periodic orbits are dense in the chaotic attractor/repeller, mean quantities can be calculated by averaging them on cycles with weights given by the relative time a typical orbit spends near each periodic orbit. In this section we apply this idea in the simplest setting of calculating escape rates, proceeding first formally, and then giving a heuristic justification.

The classical trace formula for maps

The survival ratio $e^{-\gamma}$ is the largest eigenvalue of \mathcal{L} . To get access to the spectrum of \mathcal{L} we first express $\text{tr } \mathcal{L}^n$ in terms of cycles

trIn1d

$$\text{tr } \mathcal{L}^n = \int \delta(x - f^n(x)) dx = \sum_{k, f^n(\xi_k) = x} \frac{1}{|1 - \Lambda_p|} = \sum_{n\text{-cycles } p} \frac{n_p}{|1 - \Lambda_p|} \tag{3}$$

where $\Lambda_p = \left. \frac{d}{dx} f^n(x) \right|_p = f'(x_p^{(1)}) \cdots f'(x_p^{(n)})$, and n_p is the number of *distinct* points in the cycle p . When the cycle contains no periodic sub-orbit, the cycle is *prime* and $n_p = n$; otherwise the cycle is composite and $n = n_p r$ with integer $r \geq 2$.

When the map acts on d -dimensional space this is generalised to

trIn

$$\text{tr } \mathcal{L}^n = \sum_{n\text{-cycles } p} \frac{n_p}{|\det(1 - M_p)|} = \sum_{n\text{-cycles } p} \frac{n_p}{|1 - \Lambda_{p,1}| \cdots |1 - \Lambda_{p,d}|} \tag{4}$$

where $M_p = \left. \frac{\partial^n}{\partial x^n} \right|_p = \left. \frac{\partial f}{\partial x} \right|_{x_p^{(n)}} \cdots \left. \frac{\partial f}{\partial x} \right|_{x_p^{(1)}}$ is the *monodromy* matrix, and $\Lambda_{p,k}$ are its eigenvalues.

The trace formula is a generating function for (4)

trFormula

$$\sum_{n=1}^{\infty} z^n \text{tr } \mathcal{L}^n = \text{tr} \frac{z\mathcal{L}}{1 - z\mathcal{L}} = \sum_n \sum_{n\text{-cycles } p} \frac{z^n n_p}{|\det(1 - M_p)|} = \sum_{\text{prime cycles } p} \sum_r \frac{n_p z^{rn_p}}{|\det(1 - M_p^r)|} \tag{5}$$

Given that our expression for $\text{tr } \mathcal{L}^n$ involves sums over all n -cycles, and that the number of such cycles grows exponentially with n , the size of term in the sum must be exponentially small for the sum to converge. This will happen if the map is *uniformly hyperbolic*, which means that

1. There is at least one expanding monodromy eigenvalue $\Lambda_{p,k} > 1$ for some k ,
2. There are no neutral eigenvalues $|\Lambda_{p,k}| \neq 1$ for all k , and
3. The expanding eigenvalues $\Lambda_{p,e}$ grow exponentially and contracting eigenvalues $\Lambda_{p,c}$ (if any) diminish exponentially, that is, there is a positive λ such that $|\Lambda_{p,e}| > e^{\lambda n_p}$, and $|\Lambda_{p,c}| < e^{-\lambda n_p}$.

This guarantees that individual terms remain small. Neutrally stable eigenvalues $|\Lambda_{p,k}| = 1$ are not good because even if $\Lambda_{p,k} \neq 1$, because arbitrary positive integral powers of $\Lambda_{p,k}$ will appear for repeats of p , and these will eventually come close to 1.

The spectral determinant for maps

Equation (5) expresses the spectrum of the evolution operator in terms of periodic orbits, but is not convenient to use as is, because the eigenvalues appear in it as poles. Instead we will use it to derive a cycle expansion for the *spectral determinant*, $\det(1 - z\mathcal{L})$, which has zeros at inverse eigenvalues. Starting from the identity $\log(\det(1 - z\mathcal{L})) = \text{tr}(\log(1 - z\mathcal{L}))$ we get

delcyc

$$\Delta(z) = \det(1 - z\mathcal{L}) = \exp\left(-\sum_{n=1}^{\infty} \frac{z^n \operatorname{tr}\mathcal{L}^n}{n}\right) = \exp\left(-\sum_{\text{prime cycles } p} \sum_r \frac{z^{r n_p}}{r |\det(1 - M_p^r)|}\right) \quad (6)$$

where r as before is the number of repetitions of the cycle ($= 1$ for prime cycles). The Taylor expansion of Δ

$$\Delta(z) = 1 - z \operatorname{tr} \mathcal{L} - \frac{z^2}{2} (\operatorname{tr} \mathcal{L}^2 - \operatorname{tr}^2 \mathcal{L}) - \dots \quad (7)$$

is most easily calculated from

$$\frac{-z \frac{d}{dz} \det(1 - z\mathcal{L})}{\det(1 - z\mathcal{L})} = -z \frac{d}{dz} \log(\det(1 - z\mathcal{L})) = \operatorname{tr} \frac{z\mathcal{L}}{1 - z\mathcal{L}} \quad (8)$$

so that if Δ_n is the n th Taylor of $\Delta(z)$ then

$$(1 + \Delta_1 z + \Delta_2 z^2 + \dots)(z \operatorname{tr} \mathcal{L} + z^2 \operatorname{tr}^2 \mathcal{L} + \dots) = -\Delta_1 z - 2 \Delta_2 z^2 - 3 \Delta_3 z^3 - \dots \quad (9)$$

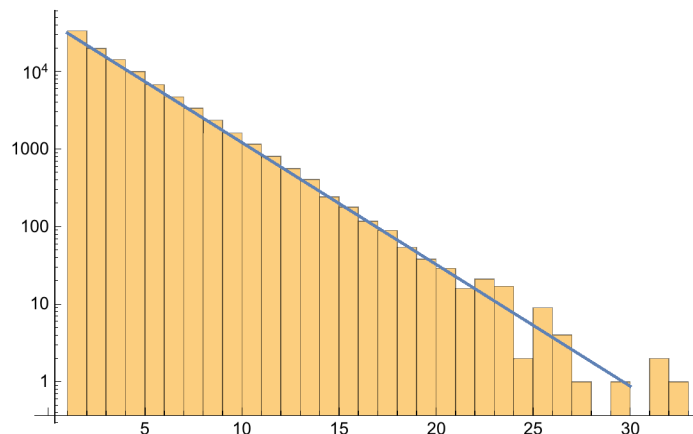
and the resulting hierarchy of equations is solved recursively by

$$\Delta_n = -\frac{1}{n} (\operatorname{tr} \mathcal{L}^n + \Delta_1 \operatorname{tr} \mathcal{L}^{n-1} + \dots + \Delta_{n-1} \operatorname{tr} \mathcal{L}) \quad (10)$$

If Δ_n decay fast enough for large n , when can use polynomial truncations of the spectral determinant to obtain approximations for γ , whose accuracy increases as a function of n . If $\Delta^{(n)}(z)$ is the n th order Taylor expansion of Δ then the n th order approximation for the escape rate is $\gamma_n = \log(z_0^{(n)})$ where $z_0^{(n)}$ is the smallest positive real zero of $\Delta^{(n)}$. The results of numerical calculations of $\operatorname{tr} \mathcal{L}^n$, Δ_n , and the n th order approximation to escape rate γ_n for the logistic map with $r = 4.5$ are

n	trace	det	escape rate
1	0.571429	-0.571429	0.5596157879354226
2	0.484848	-0.0791589	0.3758540476477187
3	0.33894	-0.00555001	0.3616614486696799
4	0.235402	-0.0000426985	0.3615098389545287
5	0.164061	-3.33517×10^{-8}	0.3615096690460328
6	0.114285	1.08885×10^{-10}	0.3615096698423153
7	0.0796138	-2.71053×10^{-14}	0.3615096698420306

The escape rate agrees beautifully with the decay rate of a histogram of survival based on one hundred thousand orbits starting from random initial points.



The dynamical zeta function

The magic of the cycle expansion for the escape rate lies in the exquisite precision obtained using only a small number of orbits. To try to understand this phenomenon let us consider a slightly

modified cycle expansion. We start by separating the expanding eigenvalues $\Lambda_{e,k}$ and the contracting eigenvalues $\Lambda_{c,k}$ appearing in the denominator in equation (4)

trin

$$|1 - \Lambda_{p,1}| \cdots |1 - \Lambda_{p,d}| = |\Lambda_p| \prod_k \left| 1 - \frac{1}{\Lambda_{e,k}} \right| \prod_k |1 - \Lambda_{c,k}| \tag{11}$$

where $\Lambda_p = \prod_k \Lambda_{e,k}$. For uniformly hyperbolic systems $\left| 1 - \frac{1}{\Lambda_{e,k}} \right|$ and $|1 - \Lambda_{c,k}|$ are very close to 1, so it makes sense to replace $\text{tr} \mathcal{L}^n$ by

trin

$$\Gamma_n = \sum_{n\text{-cycles } p} \frac{n_p}{|\Lambda_p|} \tag{12}$$

in (6) obtaining

zetcyc

$$\begin{aligned} \frac{1}{\zeta(z)} &= \exp\left(-\sum_{n=1}^{\infty} \frac{z^n \Gamma_n}{n}\right) = \\ &= \exp\left(-\sum_{\text{prime cycles } p} \sum_r \frac{z^r n_p}{r |\Lambda_p|^r}\right) = \exp\left(-\sum_p \log\left(1 - \frac{z^{n_p}}{|\Lambda_p|}\right)\right) = \prod_p \left(1 - \frac{z^{n_p}}{|\Lambda_p|}\right) \end{aligned} \tag{13}$$

It is possible to express ζ as a ratio of two spectral determinants, showing that $\frac{1}{\zeta}$ has the same zeros as the spectral determinant for nice enough systems.

The dynamical zeta function $\zeta(z)$ is a generalisation of the Riemann zeta function defined for $\text{Re } s > 1$ by

delcyc

$$\zeta_R(s) = \sum_{n=1}^{\infty} \frac{1}{n^{-s}} = \sum_{r_1, r_2, \dots} \frac{1}{p_1^{-s r_1} p_2^{-s r_2} \dots} = \prod_{\text{primes } p} \frac{1}{1 - p^{-s}} \tag{14}$$

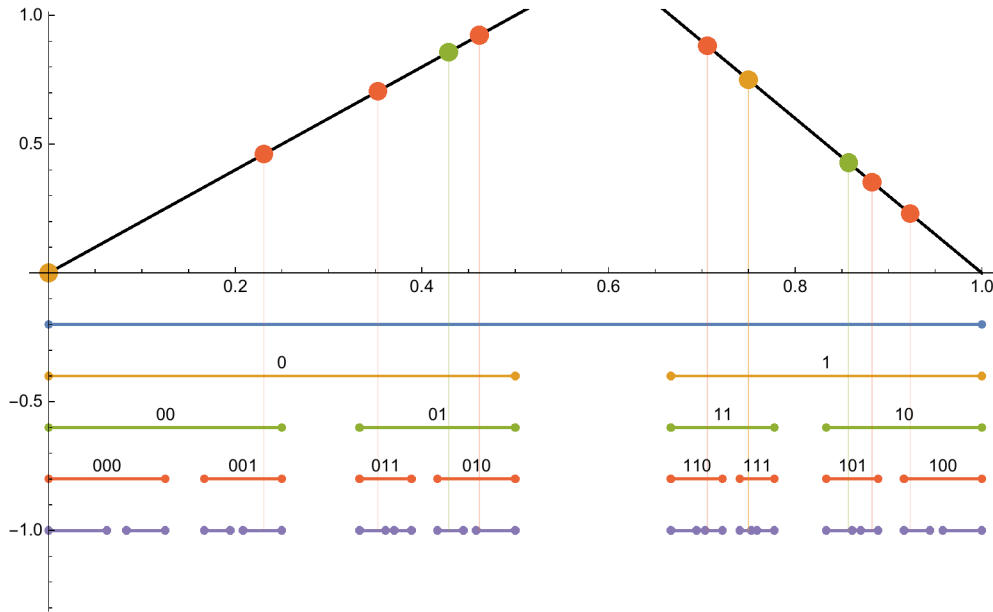
where p_1, p_2, \dots are the prime integers, and r_1, r_2, \dots are summed from zero to infinity.

We will calculate ζ explicitly for the piecewise linear repelling tent map

delcyc

$$t(x) = \begin{cases} s_0 x & x \leq \frac{s_1}{s_0 + s_1} \\ s_1(1 - x) & x \geq \frac{s_1}{s_0 + s_1} \end{cases} \tag{15}$$

with positive slopes s_0, s_1 such that $s_0 s_1 > s_0 + s_1$. To find the periodic orbits we calculate the inverse images of the unit interval: $t^{-1}([0, 1]) = \left[0, \frac{1}{s_0}\right] \cup \left[1 - \frac{1}{s_1}, 1\right]$, $t^{-2}([0, 1]) = \left[0, \frac{1}{s_0^2}\right] \cup \dots$ etc. $t^{-n}([0, 1])$ is the union of 2^n disjoint interval, each of which can be labelled by a sequence of branches $b_n b_{n-1} \cdots b_1$ of t^{-1} (read from right to left).



Each interval contains precisely one point of an n -cycle, namely $\lim_{r \rightarrow \infty} (t_{b_n}^{-1} \circ t_{b_{n-1}}^{-1} \circ \dots \circ t_{b_1}^{-1})^r [0, 1]$. The monodromy of a $b_n b_{n-1} \dots b_1$ cycle is $\pm \prod_{k=1}^n s_{b_k}$ and each cycle (prime or composite) is represented by $\frac{n}{r}$ points so that

trin

$$\Gamma_n = \sum_{b_1 b_2 \dots b_n} \frac{1}{\prod_{k=1}^n s_{b_k}} = \left(\frac{1}{s_0} + \frac{1}{s_1} \right)^n \tag{16}$$

and

trin

$$\frac{1}{\zeta} = \exp\left(-\sum_n \frac{z^n}{n} \left(\frac{1}{s_0} + \frac{1}{s_1}\right)^n\right) = e^{\log\left(1 - \frac{z}{s_0} - \frac{z}{s_1}\right)} = 1 - \left(\frac{1}{s_0} + \frac{1}{s_1}\right)z \tag{17}$$

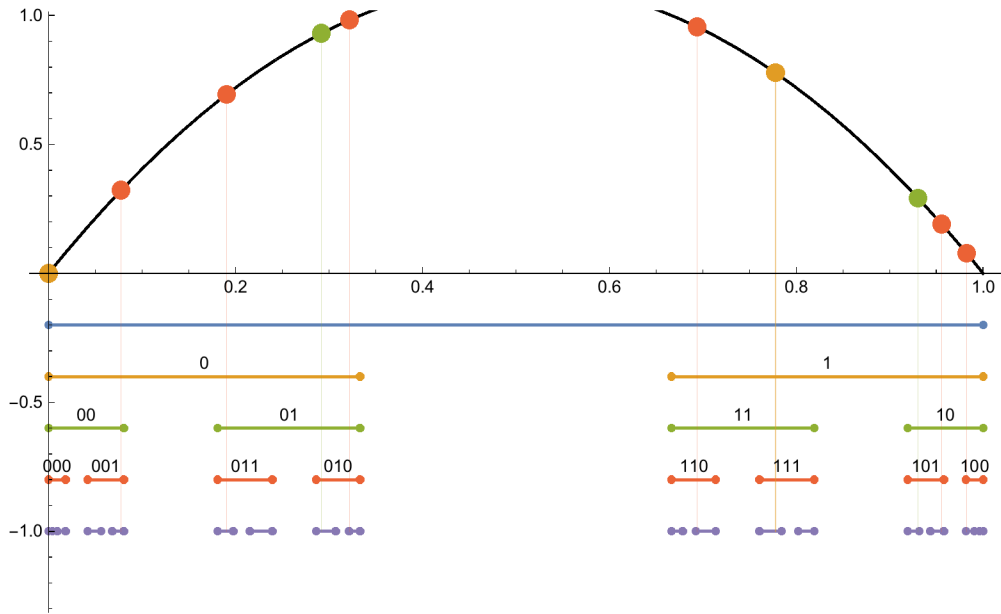
so that $\gamma = \log\left(\frac{1}{s_0} + \frac{1}{s_1}\right)$.

How is it that the infinite product (13) is equal to a linear polynomial? Since each prime cycle in the product is weighted by z^{n^p} we can find the low order Taylor coefficients of $\frac{1}{\zeta}$ by looking at the short prime cycles. The prime cycles of order up to 3 are 0, 1, 01, 001, 011, so that

trin

$$\begin{aligned} \frac{1}{\zeta} &= \\ & \left(1 - \frac{z}{s_0}\right) \left(1 - \frac{z}{s_1}\right) \left(1 - \frac{z^2}{s_0 s_1}\right) \left(1 - \frac{z^3}{s_0^2 s_1}\right) \left(1 - \frac{z^3}{s_0 s_1^2}\right) \dots = 1 - \left(\frac{1}{s_0} + \frac{1}{s_1}\right)z - \left(\frac{1}{s_0 s_1} - \frac{1}{s_0} \frac{1}{s_1}\right)z^2 - \\ & \left(\frac{1}{s_0^2 s_1} + \frac{1}{s_0 s_1^2} - \left(\frac{1}{s_0} + \frac{1}{s_1}\right) \frac{1}{s_0 s_1}\right)z^3 + \dots = 1 - \left(\frac{1}{s_0} + \frac{1}{s_1}\right)z \end{aligned} \tag{18}$$

The exact cancellation of high order Taylor coefficients of $\frac{1}{\zeta}$ is a special property of piecewise linear maps, but the qualitative structure we found for the tent map is valid for any unimodal map repelling map f .



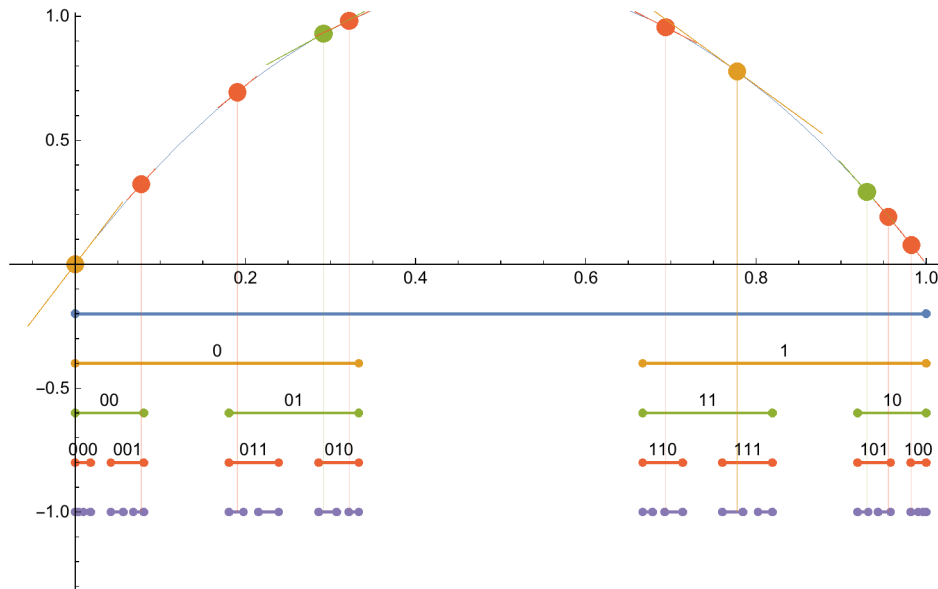
As with the tent map the n th inverse image of the interval $f^{-n}([0, 1])$ consists of 2^n disjoint intervals labelled by binary sequences of length n , each containing an n cycle, so that

trin

$$\frac{1}{\zeta} = \left(1 - \frac{z}{\Lambda_0}\right) \left(1 - \frac{z}{\Lambda_1}\right) \left(1 - \frac{z^2}{\Lambda_{01}}\right) \left(1 - \frac{z^3}{\Lambda_{001}}\right) \left(1 - \frac{z^3}{\Lambda_{011}}\right) \cdots =$$

$$1 - \left(\frac{1}{\Lambda_0} + \frac{1}{\Lambda_1}\right)z - \left(\frac{1}{\Lambda_{01}} - \frac{1}{\Lambda_0} \frac{1}{\Lambda_1}\right)z^2 - \left(\frac{1}{\Lambda_{001}} + \frac{1}{\Lambda_{011}} - \left(\frac{1}{\Lambda_0} + \frac{1}{\Lambda_1}\right) \frac{1}{\Lambda_{01}}\right)z^3 + \cdots \tag{19}$$

The coefficient of z^n no longer cancel exactly in this case, but if f is smooth $\frac{1}{\Lambda_{p_1 p_2}} - \frac{1}{\Lambda_{p_1}} \frac{1}{\Lambda_{p_2}}$, where $p_1 p_2$ is the prime cycle obtained by juxtaposing the binary representations of p_1 and p_2 , decays exponentially in the length of $p_1 p_2$, and the Taylor expansion of $\frac{1}{\zeta}$ has a radius of convergence larger than 1 (Figures and table shown for the logistic map $I_{4.5}$).



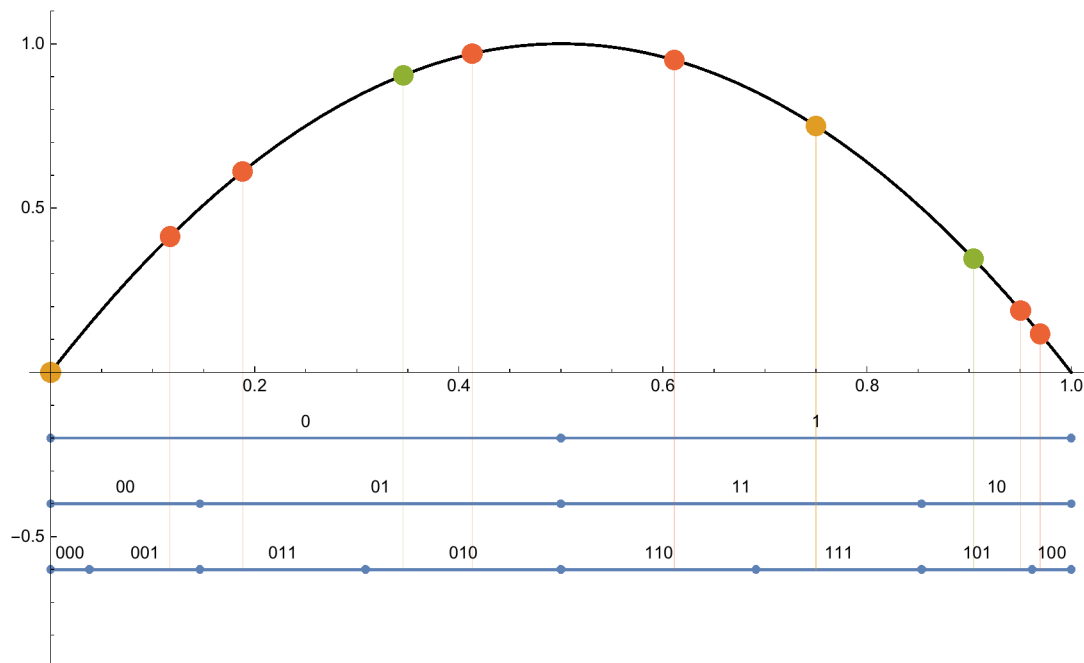
n	$\frac{1}{\zeta}$	det	escape rate (ζ)	escape rate (det)
1	-0.622222	-0.571429	0.4744579795951156	0.5596157879354220
2	-0.0490421	-0.0791589	0.3667382300226438	0.3758540476477180
3	-0.00213844	-0.00555001	0.3609881774789707	0.3616614486696790
4	0.000142274	-0.0000426985	0.3615305562095015	0.3615098389545280
5	-3.81085×10^{-6}	-3.33517×10^{-8}	0.3615096647443185	0.3615096690460320
6	-6.38317×10^{-9}	1.08885×10^{-10}	0.3615096145164176	0.3615096698423150
7	5.22714×10^{-9}	-2.71053×10^{-14}	0.3615096735602104	0.3615096698420300
8	-2.35567×10^{-10}	1.66533×10^{-16}	0.3615096697405278	0.3615096698420330
9	4.22687×10^{-12}	-2.1048×10^{-16}	0.3615096698389136	0.3615096698420280
10	1.08083×10^{-13}	2.25167×10^{-16}	0.361509669842525	0.3615096698420350

4 Dynamical averages

Symbolic dynamics and pruning

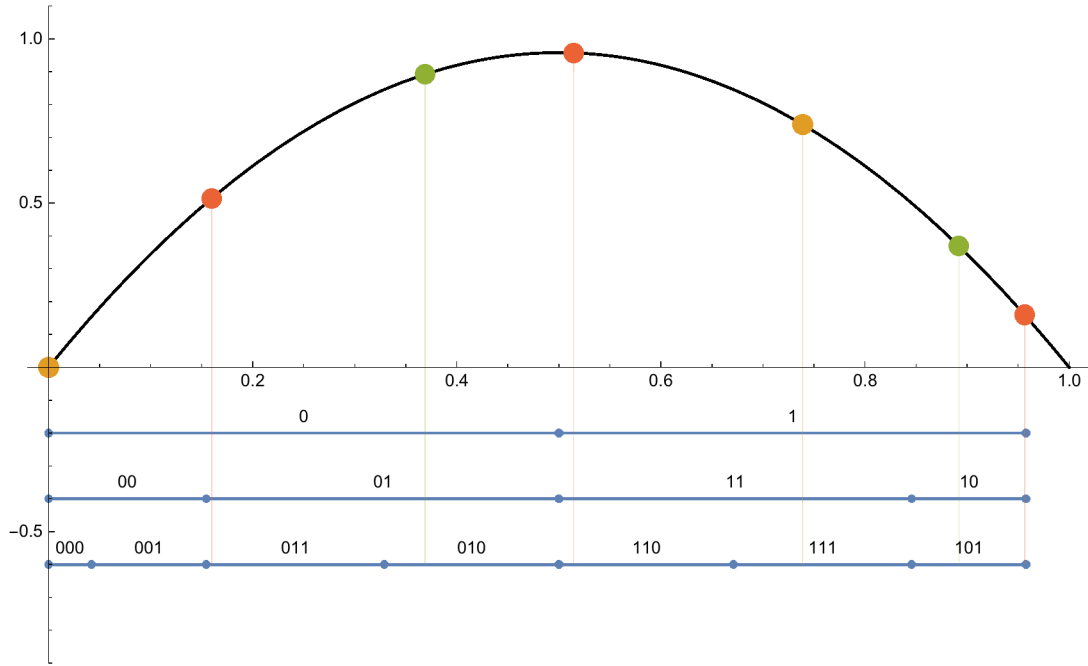
In this section our purpose is to calculate expectation values on the natural invariant measure of chaotic unimodal maps with no escaping orbits. The spectral determinant and $1/\zeta$ necessarily have a zero at $z = 1$ for such systems. However, before proceeding to study more interesting quantities we have to face a new issue.

For this purpose note that each infinite orbit of the repelling maps studied above was uniquely determined by its binary symbol sequence, and that each binary sequence corresponds to an admissible orbit. This is also true for the Ulam map I_4 ,



except that here the inverse images cover the entire interval, since the invariant set, an attractor, covers the entire interval. This is the reason that the dynamics is conjugate to the Bernoulli shift. In particular, every repeating binary sequence corresponds to an actual periodic orbit, and the cancellation of terms in the expansion of the dynamical ζ function is good.

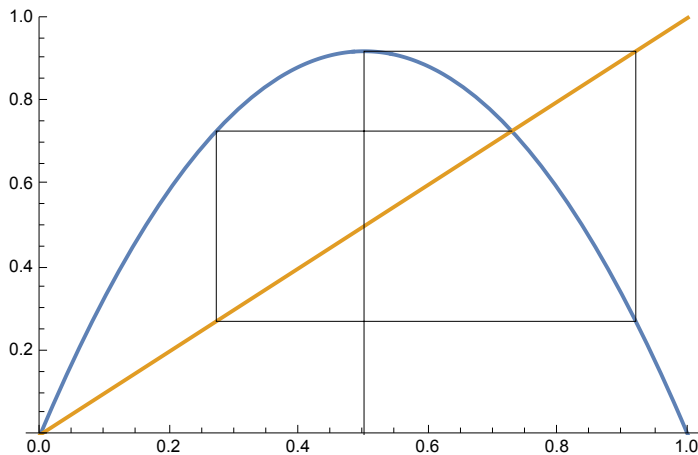
For $r < 4$ the situation is no longer as simple. Since $\max I_r = \frac{r}{4} < 1$ the points in the subinterval $]\frac{r}{4}, 1]$ are transient, and so are all their inverse images. Consequently, not all symbolic itineraries are allowed



Here, for example, $r = 3.831$, and the interval labelled 100 is *pruned*. The 100 periodic orbit is therefore also pruned, and there is only one prime 3-cycle. The pruning spoils the cancellation of the cycle expansions, which converge much more slowly, and nonuniformly:

n	$\frac{1}{\xi_{r=4}(1)}$	$\frac{1}{\xi_{r=3.9}(1)}$
1	0.25	0.217274
2	0.125	0.0589716
3	0.0625	-0.139632
4	0.03125	0.0711449
5	0.015625	0.108127
6	0.0078125	0.0449979
7	0.00390625	0.0142551
8	0.00195312	-0.176742
9	0.000976562	-0.0221847
10	0.000488281	0.00506693

The convergence improves significantly when the pruning is described by a finite set of pruning rules. Then the dynamics is conjugate to a finite Bernoulli subshift, and cancellations becomes effective once periods longer than the longest pruning rule are taken into account. An example of this type is the logistic map with $r = r_{gm} \approx 3.68$, where $\frac{1}{2}$ is mapped in four step to the unstable fixed point x_1



The symbolic dynamics is such that every itinerary without two consecutive 0 symbols is allowed.

To get good convergence the $\bar{0}$ cycle that is not in the attractor has to be pruned as well.

n	$\frac{1}{\zeta_{x=4}(1)}$	$\frac{1}{\zeta_{x=3.9}(1)}$ (with $\bar{0}$)	$\frac{1}{\zeta_{x_{\text{pm}}}(1)}$ (w/out $\bar{0}$)
1	0.25	0.217274	0.404256
2	0.125	0.0589716	-0.0555656
3	0.0625	-0.139632	0.21837
4	0.03125	0.0711449	-0.00843835
5	0.015625	0.108127	0.126682
6	0.0078125	0.0449979	-0.00128575
7	0.00390625	0.0142551	0.07495
8	0.00195312	-0.176742	-0.000196076
9	0.000976562	-0.0221847	0.0445717
10	0.000488281	0.00506693	-0.0000299061
11	0.000244141	0.0347074	0.0265413
12	0.000122071	0.000988961	-4.56151×10^{-6}
13	0.0000610348	-0.005118	0.0158099
14	0.00003052	0.00488018	-6.95757×10^{-7}
15	0.0000152584	0.00919761	0.0094184
16	7.62168×10^{-6}	0.00394753	-1.06122×10^{-7}

Operator eigenvalues as a generating function

Our next goal is to use cycle expansions to calculate expectation values of the form $\langle a \rangle = \int a d\mu$, where a is a function of the dynamical variable x and μ is the natural invariant measure of a closed discrete dynamical system. For a chaotic system, this expectation value is equal to the long-time limit of a dynamical average taken on a typical orbit $\langle a \rangle = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n a_n$, with $a_n = a(f^n(x_0))$.

The idea of cycle expansions for dynamical averages is to construct a weighted evolution operator \mathcal{L}_β whose leading eigenvalue is a moment generating function. Namely, let

$\mathcal{L}_\beta \rho(x) = \int e^{\beta a(f(\xi))} \rho(\xi) \delta(x - f(\xi)) d\xi$, so that

$$\mathcal{L}_\beta^2 \rho(x) = \int e^{\beta(a(f(\xi_1)) + a(f(\xi_2)))} \delta(x - f(\xi_1)) \rho(\xi_2) \delta(\xi_1 - f(\xi_2)) d\xi_1 d\xi_2 = \int e^{\beta(a(f(\xi)) + a(f^2(\xi)))} \delta(x - f^2(\xi)) \rho(\xi) d\xi \quad (20)$$

and

$$\int dx \mathcal{L}_\beta^n \rho(x) = \int e^{\beta(a(f(\xi)) + a(f^2(\xi)) + \dots + a(f^n(\xi)))} \rho(\xi) d\xi = \langle e^{\beta A_n} \rangle \quad (21)$$

defining $A_n(x) = \sum_{k=1}^n a_n$ so that $\langle a \rangle = \lim_{n \rightarrow \infty} \frac{\langle A_n \rangle}{n} = \lim_{n \rightarrow \infty} \frac{1}{n} \frac{d}{d\beta} \log \langle e^{\beta A_n} \rangle_{\beta=0}$.

Now, if $e^{s(\beta)}$ is the largest eigenvalue of \mathcal{L}_β then $\langle e^{\beta A_n} \rangle \sim e^{n s(\beta)}$ and $\langle a \rangle = \left. \frac{ds}{d\beta} \right|_{\beta=0}$. We already know

how to use cycle expansion to calculate e^s : It is the inverse of smallest positive zero of $\frac{1}{\zeta_\beta}$, the inverse zeta function corresponding to \mathcal{L}_β . Retracing the steps leading to (13) we obtain that

$\Gamma_{\beta,n} = \sum_{n\text{-periods } p} \frac{n_p e^{\beta A_p}}{|\Lambda_p|}$, where $A_p = \sum_k a(f(x_p^{(k)}))$ and therefore

zetcyc

$$\frac{1}{\zeta_\beta(z)} = \exp\left(-\sum_{n=1}^{\infty} \frac{z^n \Gamma_{\beta,n}}{n}\right) = \prod_{\text{prime cycles } p} \left(1 - \frac{z^{n_p} e^{\beta A_p}}{|\Lambda_p|}\right) \quad (22)$$

To get $\langle a \rangle$ we can differentiate the implicit relation $\frac{1}{\zeta_\beta(e^{-s(\beta)})} = 0$ obtaining

$$\left(\frac{\partial}{\partial \beta} - s'(\beta) z \frac{\partial}{\partial z}\right) \frac{1}{\zeta_\beta} \Big|_{\beta=0, z=1} = 0 \implies \langle a \rangle = s'(0) = \frac{\sum_p \prod_{q \neq p} \left(1 - \frac{1}{|\Lambda_q|}\right) \frac{A_p}{|\Lambda_p|}}{\sum_p \prod_{q \neq p} \left(1 - \frac{1}{|\Lambda_q|}\right) \frac{n_p}{|\Lambda_p|}} = \frac{[A]}{[n]} \quad (23)$$

defining $\langle \cdot \rangle$ for additive functions $B_p = \sum_n b(x_p^{(n)})$, n being additive for the constant function 1.

Examples: Right dwell fraction and the Lyapunov exponent

Two example dynamical averages will be shown for the logistic map. The right dwell fraction is the probability to visit the decreasing part of the map, $p(x > \frac{r}{8}) = \langle \Theta(x - \frac{r}{8}) \rangle$, where Θ is the unit step function

$\mathbb{P}(x > \frac{r}{8})$	$r=4$	$r=3.9$	r_{gm} (w/out $\bar{0}$)
1	0.666667	0.672414	1.
2	0.625	0.622759	0.696565
3	0.578947	0.540562	0.731991
4	0.547619	0.636826	0.742197
5	0.52809	0.707581	0.699989
6	0.516304	0.615374	0.756479
7	0.509333	0.58925	0.713609
8	0.505277	0.542435	0.802301
9	0.502951	0.565944	0.75066
10	0.501634	0.585134	0.805908
11	0.500897	0.648843	0.77293
12	0.500489	0.583056	0.800847
13	0.500265	0.575411	0.783433
14	0.500142	0.588143	0.802898
15	0.500076	0.59642	0.790757
16	0.500041	0.587392	0.803952
17	0.500021	0.579733	0.795714
18	0.500013	0.585421	0.805507
19	0.500001	0.582077	0.799432
20	0.500006	0.579021	0.805819
10^6 orbit	0.499317	0.580483	0.806497
2×10^6 orbit	0.499615	0.580549	0.806099
Exact	0.5		

The second example is the Lyapunov exponent $\lambda = \langle \log |f'(x)| \rangle$

lyapunov	$r=4$	$r=3.9$	r_{gm} (w/out $\bar{0}$)
1	0.924196	0.877429	0.517944
2	0.779791	0.706101	0.439363
3	0.729629	0.560364	0.448537
4	0.709651	0.562051	0.404545
5	0.700935	0.612267	0.40763
6	0.696914	0.593696	0.378749
7	0.694996	0.560029	0.383327
8	0.694062	0.381056	0.363434
9	0.693602	0.441828	0.36846
10	0.693374	0.49078	0.354598
11	0.69326	0.59209	0.359144
12	0.693204	0.509531	0.34948
13	0.693175	0.496493	0.353197
14	0.693161	0.510633	0.346491
15	0.693154	0.52249	0.349351
16	0.693151	0.510767	0.344733
17	0.693149	0.499492	0.346845
18	0.693148	0.506918	0.343693
10^6 orbit	0.693138	0.495094	0.342219
2×10^6 orbit	0.693147	0.496758	0.342325
Exact	$\text{Log}[2] = 0.693147$		

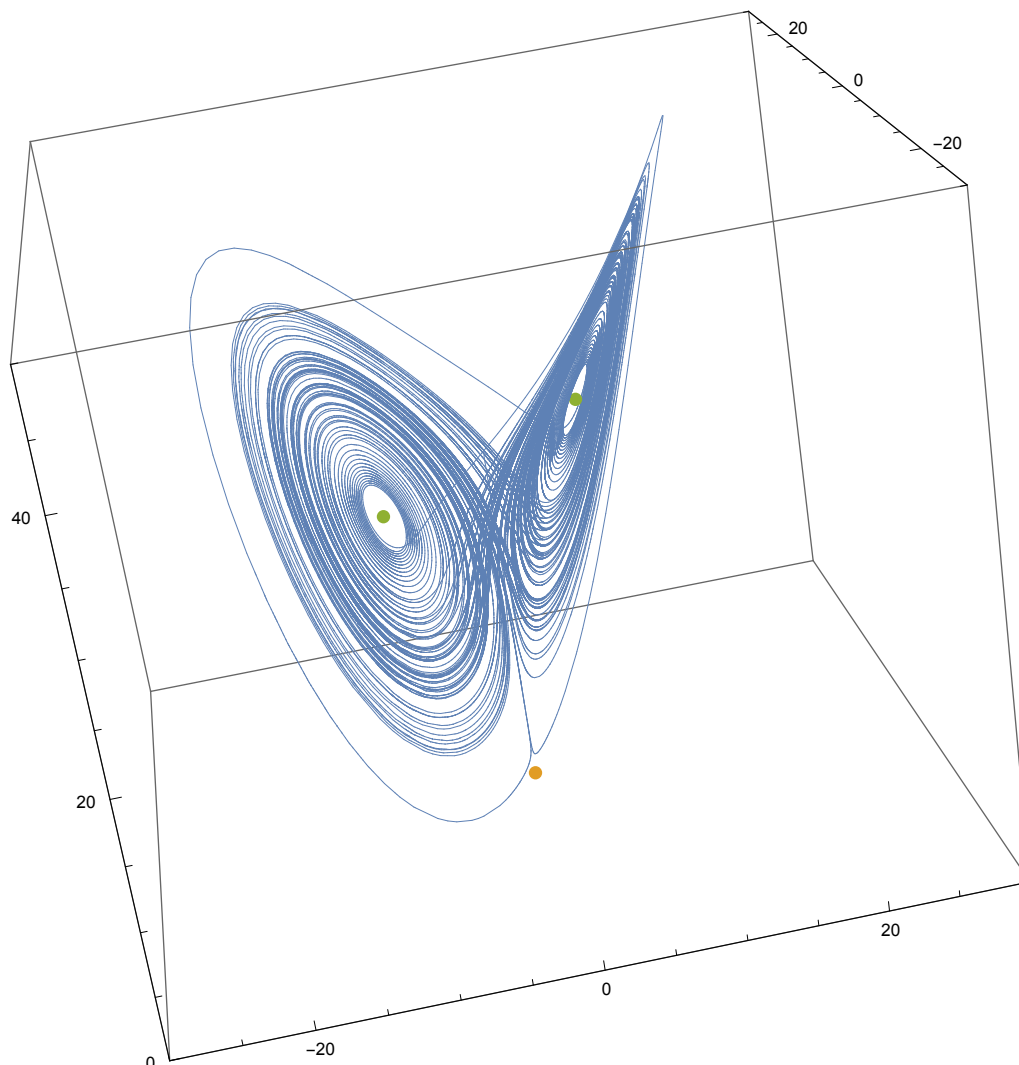
The cycle expansions seem to work, but the performance is mediocre. Although the convergence is exponential - the number of significant digits grows linearly with the maximum cycle length, the performance is comparable to that of simple orbit averaging, because the number of periodic orbits grows exponentially with their length. In practice, it is therefore advantageous to accelerate the convergence by replacing the ζ function with a spectral determinant, control of the pruning process to allow for better cancellations, tail resummation, or other methods. These are beyond the present scope.

5 The Lorenz attractor

Qualitative dynamics

The Lorenz dynamics has three fixed points, one at the origin, and two, related by symmetry, at the centres of the wings. For the standard parameter values all fixed points are unstable, that is, at least one of the eigenvalues of the Jacobian matrix $\partial f/\partial x$ has positive real part. The Jacobian eigenvectors associated with the unstable eigenvalues span unstable directions. Because the Lorenz system is strongly dissipative there are also stable directions at each fixed point, so they are all hyperbolic.

The wing fixed points have a complex conjugate pair of unstable eigenvalues which generate the outward spiralling, and a strongly stable eigenvalue which compresses the dynamics to the almost two-dimensional wings. The fixed point at the origin has one unstable direction, tangent to the intersection of the attractor with the xy plane, a stable direction along the z axis, and a strongly stable direction transverse to the attractor

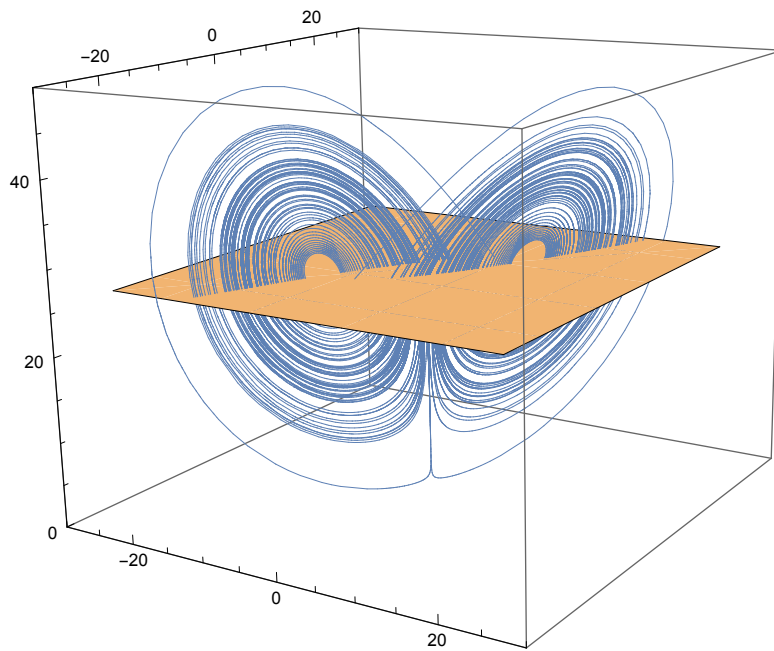


This information is sufficient to understand the qualitative dynamics. Since orbits leave the origin in the unstable direction, the unstable manifold of the origin fixed point delimits the outer boundaries of the wings. These orbits are outside the wing switching radius, so they continue to the other wing, where they reach as close as possible to the wing fixed points, and start to spiral out close to the unstable manifold of the wing fixed points. The intersection of the wing with the two dimensional

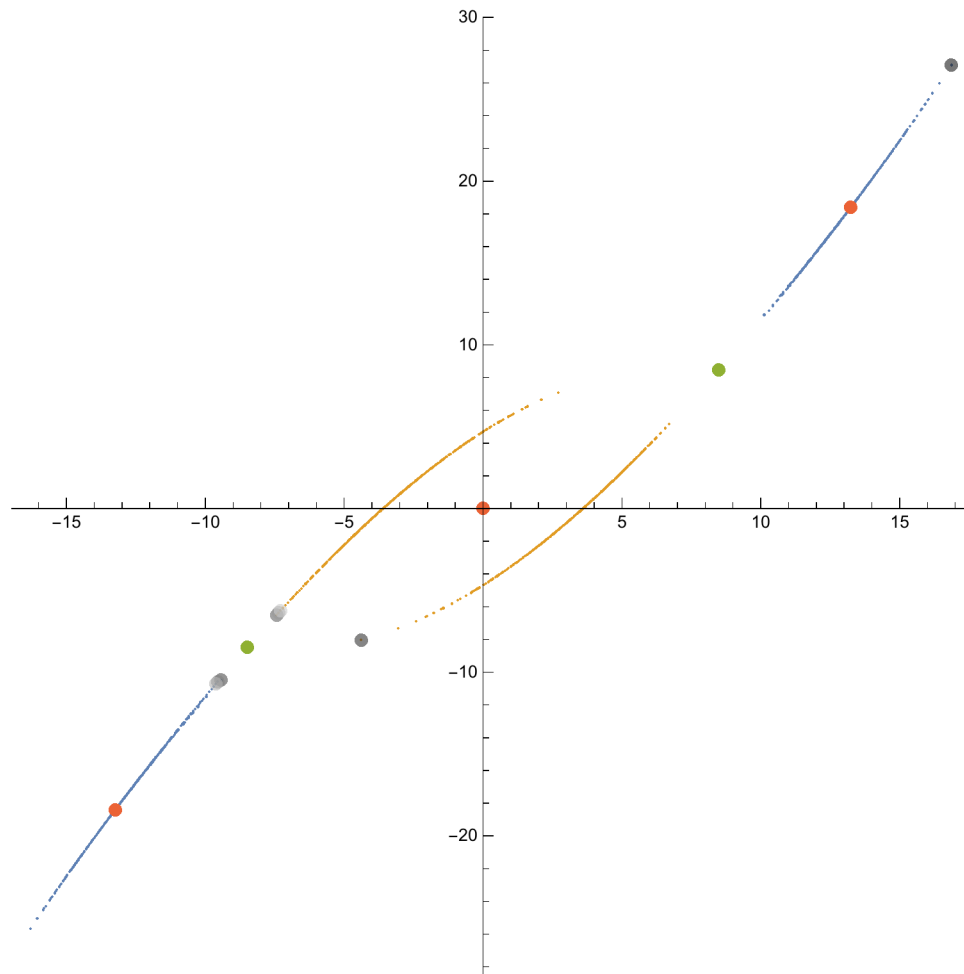
stable manifold of the origin fixed point is the separatrix that separates between staying and switching orbits

The Poincaré map

A Poincaré section of a d -dimensional continuous dynamical system is simply a set of intersection points of an orbit or orbits with a $d - 1$ -dimensional submanifold. The key observation is that the flow induces a map on the section, the Poincaré or return map, which maps each intersection point to the next one. A good section for the Lorenz system is the plane $z = r - 1$ which passes through the wing fixed points.

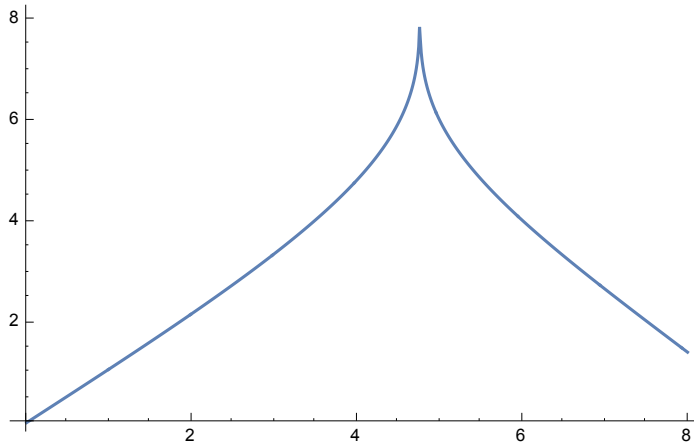


The section of the orbit shown above, which starts on the section plane close to the z axis is



Points that intersect the plane going up are marked in blue and going down in orange. The two green points are the wing fixed points, and the red points are on the stable manifold of the fixed point at the origin. The point in the center is the initial point, which is not on the attractor. The other starting points of the orbit are marked in grey: the orbit first visits the right wing, going up and then down at its outer boundary, and then hits the inner part of the left wing, where it starts spiralling outward.

It is convenient to define the return map between upward going points. Then a mapped point remains on the same wing if it is inside the origin stable manifold intersection (outer red points on the graph, with approximate displacement 4.76, 9.92 from the wing fixed point). In this inner region the distance of the mapped point from the fixed point is monotonically increasing as a function of the distance of the original point. Beyond the separatrix points are mapped to the other wing, and the distance of the mapped point from the fixed point (of the new wing) is monotonically decreasing as a function of the distance of the original point. This explains the structure of the return map, here parameterized by the x distance from the fixed point



To be sure, the Poincare map arises from an invertible flow, so it is also invertible, and the two inverse images of 4, say, are not mapped to precisely the same point; rather their images are very slightly spaced in the transverse, fractal, direction of the wing. This distinction however hardly matters for the statistical properties we're studying here.

Switching rate & the Lyapunov exponent

The Lorenz system dynamics having been effectively reduced to a unimodal map, can now be analyzed using the ideas and tools developed above. I will present two examples, the first of which is the switching rate γ_d between the two wings. Given that points on the far side of the attractor, with $x > x_m \approx 4.76$ switch and points with $x < x_m$ stay, $\gamma_d = p(x > x_m)$. This gives the discrete version of the switching rate, where 'time' is measured with attractor cycles. The continuous switching rate γ_c can be calculated by the flow-version of the dynamical ζ function, which was not presented here, is equal to $\frac{[\Theta(x-x_m)]}{[T]}$ instead of $\frac{[\Theta(x-x_m)]}{[n]}$ for γ_d where T_p is the flow time of cycle p .

n	γ_d	γ_c
1	1.	1.28309
2	0.753485	0.971894
3	0.636427	0.826553
4	0.568264	0.742421
5	0.524732	0.688835
6	0.495566	0.652998
7	0.475547	0.628446
8	0.457728	0.606528
9	0.450241	0.597445
10	0.445255	0.591357
11	0.439818	0.584699
12	0.430837	0.573687
13	0.43255	0.575727
14	0.430812	0.573571
15	0.426658	0.568471
16	0.421008	0.561554
$10^5/t=10^4$ orbit	0.42553	0.5799
$2 \times 10^5/t=2 \times 10^4$ orbit	0.426085	0.56975

The second example is the Lyapunov exponent, again with discrete version λ_d and continuous version λ_c .

n	λ_d	λ_c
1	0.776805	0.99671
2	0.763056	0.984239
3	0.744968	0.967519
4	0.729353	0.952879
5	0.716855	0.941041
6	0.707161	0.931811
7	0.699774	0.924768
8	0.691422	0.916192
9	0.688996	0.91426
10	0.687138	0.912609
11	0.684527	0.910018
12	0.678415	0.903354
13	0.680096	0.905212
14	0.679235	0.904315
15	0.676184	0.900934
16	0.671277	0.89537
$10^5/t=10^3$ orbit	0.674379	0.907762
$2 \times 10^5/t=2 \times 10^3$ orbit	0.674712	0.901022

Auxiliary definitions

```
In[1]:= lm[r_, x_] := r x (1 - x)
```

```
In[2]:= ilm[r_, s_, y_] :=  $\frac{1}{2} \left( 1 + (2s - 1) \sqrt{1 - 4 \frac{y}{r}} \right)$ 
```

```
In[3]:= lmx[r_] :=  $\frac{r}{4}$ 
```

```
In[4]:= rgm =  
r /. FindRoot[Last@(Differences@NestList[lm[r, #] &, 0.5, 4]), {r, 3.68, 3.67}]
```

```
Out[4]= 3.67857
```

```
In[5]:= tent[{ls_, rs_}, x_] := Piecewise[{{ls x, x ≤  $\frac{-rs}{ls - rs}$ }, {rs (x - 1), x >  $\frac{-rs}{ls - rs}$ }}
```

```
In[6]:= itent[{ls_, rs_}, 0, y_] :=  $\frac{y}{ls}$ 
```

```
In[7]:= itent[{ls_, rs_}, 1, y_] :=  $1 + \frac{y}{rs}$ 
```

```
In[8]:= plotOrbit[f_, s_, n_] :=  
Module[{orbit}, orbit = NestWhileList[f[#] &, s, # < 1 &, 1, n];  
Show[Plot[{f[x], x}, {x, 0, 1}, PlotRange → {0, 1}, PlotStyle → Thick], Graphics[  
Line[Join[{{orbit[[1]], 0}}, Flatten[Table[{{orbit[[k - 1]], orbit[[k]]},  
{orbit[[k]], orbit[[k]]}}, {k, 2, Length[orbit]}, 1]]]]]
```

```
In[9]:= plotOrbits[f_, s_List, n_, st_List] :=  
Module[{orbits}, orbits = NestWhileList[f[#] &, #, # < 1 &, 1, n] & /@ s;  
Show[Plot[{f[x], x}, {x, 0, 1}, PlotRange → {0, 1}, PlotStyle → Thick], Graphics[  
Transpose[{st, Line[Join[{{#[[1]], 0}}, Flatten[Table[{{#[[k - 1]], #[[k]]},  
{#[[k]], #[[k]]}}, {k, 2, Length[#]}, 1]]] & /@ orbits}}]]]
```

```
In[10]:= lifeSpan[f_, s_, cutoff_] := Module[{n = 0}, NestWhileList[(n++;  
f[#]) &, s, # < 1 &, 1, cutoff];  
n]
```

```
In[11]:= imapInterval[if_, 0, int_List, fmx_] := if[0, #] & /@ int
```

```

In[12]:= imapInterval[if_, 1, int_List, fmx_] := Module[{ii}, ii = if[1, #] & /@ int;
  If[ii[[2]] > fmx, Throw["pruned"]];
  Reverse[MapAt[Min[#, fmx] &, ii, 1]]]

In[13]:= imapInterval[if_, 0, int_List, fmn_, fmx_] := Module[{ii}, ii = if[0, #] & /@ int;
  If[ii[[2]] < fmn, Throw["pruned"]];
  MapAt[Max[#, fmn] &, ii, 1]]

In[14]:= imapInterval[if_, 1, int_List, fmn_, fmx_] := Module[{ii}, ii = if[1, #] & /@ int;
  If[ii[[2]] > fmx, Throw["pruned"]];
  Reverse[MapAt[Min[#, fmx] &, ii, 1]]]

In[15]:= unimodalPO[if_, bits_, iter_] :=
  Fold[if[#2, #1] &, 0.5, Flatten@ConstantArray[bits, Ceiling@ $\frac{\text{iter}}{\text{Length}[\text{bits}]}$ ]]]

In[16]:= unimodalPO[if_, bits_, fmx_, iter_] :=
  Catch[Mean@Fold[imapInterval[if, #2, #1, fmx] &, {0, fmx},
    Flatten[ConstantArray[bits, Ceiling[ $\frac{\text{iter}}{\text{Length}[\text{bits}]}$ ]]]]]]]

In[17]:= unimodalPO[if_, bits_, fmn_, fmx_, iter_] :=
  Catch[Mean@Fold[imapInterval[if, #2, #1, fmn, fmx] &, {fmn, fmx},
    Flatten[ConstantArray[bits, Ceiling[ $\frac{\text{iter}}{\text{Length}[\text{bits}]}$ ]]]]]]]

In[18]:= monodromy[f_, x_, n_] := Module[{λ = 1}, NestWhile[(λ *= f'[#];
  f[#]) &, x, # < 1 &, 1, n];
  λ]

In[19]:= orbitAv[b_, f_, x_, n_] := Module[{av = 0}, NestList[(av += b[#];
  f[#]) &, x, n];
   $\frac{\text{av}}{n}$ ]

In[20]:= lyapunov[f_, x_, n_] := orbitAv[Log[Abs[f'[#]]] &, f, x, n]

In[21]:= lyapunov[f_, x_, n_] := Module[{λ = 0}, NestList[(λ += Log[Abs[f'[#]]];
  f[#]) &, x, n];
   $\frac{\lambda}{n}$ ]

In[22]:= trace[f_, if_, n_] := With[{iter = 100},
  Total@(1/Abs[monodromy[f, #, n] & /@ unimodalPO[if, n, iter] - 1])]

In[23]:= spectralDet[f_, if_, n_] := Module[{tr, det = {}}, tr = trace[f, if, #] & /@ Range[n];
  Do[AppendTo[det, - $\frac{1}{k}$ (tr[[k]] + Reverse[det].Take[tr, k - 1])], {k, n}];
  det]

In[24]:= escapeRate[f_, if_, n_] :=
  Log[x /. FindRoot[1 + spectralDet[f, if, n].xRange[n], {x, 0}]]]

In[25]:= escapeRateTable[f_, if_, n_] := Module[{sd}, sd = 1 + spectralDet[f, if, n].xRange[n];
  Log[x /. FindRoot[Take[sd, # + 1], {x, 0}]] & /@ Range[n]]

In[26]:=

```



```

In[27]:= primeCycles[n_] := (* With cyclic permutations *)
  primeCycles[n] = Delete[Range[2n-1, 2n - 1],
    List /@ Flatten[ ( (2n - 1) / (2# - 1) ) primeCycles[#] & /@ Drop[Divisors[n], -1] ] - 2n-1 + 1]

In[28]:= primeCycles[1] = {1}
Out[28]= {1}

In[29]:= removeCyclic[1] = {0, 1}
Out[29]= {0, 1}

In[30]:= removeCyclic[n_] := Cases[DeleteDuplicatesBy[Flatten[
  Transpose@Table[{k, RotateRight[#, k]} & /@ IntegerDigits[primeCycles[n], 2],
    {k, 0, n - 1}], 1], Last], {0, 1_List} :> FromDigits[1, 2]]

In[31]:= discardZero[k_] := removeCyclic[k]

In[32]:= discardZero[1] = {1}
Out[32]= {1}

In[33]:= pcinvζ[f_, if_, n_] :=
  Module[{iter = 100, k, ζ, tfactors}, ζ = Prepend[ConstantArray[0, n], 1];
  Do[For[tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@
    (Fold[if[#2, #1] &, 0.5, Flatten@ConstantArray[#, Ceiling@  $\frac{\text{iter}}{k}$ ]] & /@
      IntegerDigits[removeCyclic[k], 2]), tfactors ≠ {}, tfactors = Rest[
    tfactors], ζ += Take[PadLeft[ζ First[tfactors], n + 1 + k], n + 1], {k, n}];
  Rest[ζ]]

In[34]:= pcinvζ[f_, if_, fmx_, n_] :=
  Module[{iter = 100, k, ζ, tfactors}, ζ = Prepend[ConstantArray[0, n], 1];
  Do[For[tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@ Select[
    (unimodalPO[if, #, fmx, iter] & /@ IntegerDigits[removeCyclic[k], 2]),
    NumberQ], tfactors ≠ {}, tfactors = Rest[tfactors],
    ζ += Take[PadLeft[ζ First[tfactors], n + 1 + k], n + 1], {k, n}];
  Rest[ζ]]

In[35]:= pcinvζ[f_, if_, fmn_, fmx_, n_] :=
  Module[{iter = 100, k, ζ, tfactors}, ζ = Prepend[ConstantArray[0, n], 1];
  Do[For[tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@ Select[
    (unimodalPO[if, #, fmn, fmx, iter] & /@ IntegerDigits[removeCyclic[k], 2]),
    NumberQ], tfactors ≠ {}, tfactors = Rest[tfactors],
    ζ += Take[PadLeft[ζ First[tfactors], n + 1 + k], n + 1], {k, n}];
  Rest[ζ]]

```

```

In[36]:= pcinvζdz[f_, if_, fmx_, n_] :=
Module[{iter = 100, k, ζ, tfactors}, ζ = Prepend[ConstantArray[0, n], 1];
Do[For[tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@
Select[(unimodalPO[if, #, fmx, iter] & /@ IntegerDigits[discardZero[k], 2]),
NumberQ], tfactors ≠ {}, tfactors = Rest[tfactors],
ζ += Take[PadLeft[ζ First[tfactors], n + 1 + k], n + 1], {k, n}];
Rest[ζ]]

In[37]:= escapeRateζ[f_, if_, n_] := Log[x /. FindRoot[1 + pcinvζ[f, if, n].xRange[n], {x, 0}]]

In[38]:= escapeRateζTable[f_, if_, n_] := Module[{sd}, sd = 1 + pcinvζ[f, if, n].xRange[n];
Log[x /. FindRoot[Take[sd, # + 1], {x, 0}]] & /@ Range[n]]

In[39]:=

In[40]:= dynAvg[b_, f_, if_, fmx_, n_] :=
Module[{iter = 100, k, ζ, da, na, xp, tfactors, bterms},
ζ = Prepend[ConstantArray[0, n - 1], 1];
na = da = ConstantArray[0, n];
Do[For[xp = Select[(unimodalPO[if, #, fmx, iter] & /@
IntegerDigits[removeCyclic[k], 2]), NumberQ];
tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@ xp;
bterms = Total[b /@ NestList[f, #, k - 1]] & /@ xp;
tfactors ≠ {}, tfactors = Rest[tfactors];
bterms = Rest[bterms], da += Take[PadLeft[da First[tfactors], n + k], n] -
Take[PadLeft[First[bterms] First[tfactors] ζ, n + k - 1], n];
na += Take[PadLeft[na First[tfactors], n + k], n] -
Take[PadLeft[k First[tfactors] ζ, n + k - 1], n];
ζ += Take[PadLeft[ζ First[tfactors], n + k], n], {k, n}];
Accumulate[da]
Accumulate[na]]

In[41]:= dynAvg[b_, f_, if_, fmn_, fmx_, n_] :=
Module[{iter = 100, k, ζ, da, na, xp, tfactors, bterms},
ζ = Prepend[ConstantArray[0, n - 1], 1];
na = da = ConstantArray[0, n];
Do[For[xp = Select[(unimodalPO[if, #, fmn, fmx, iter] & /@
IntegerDigits[removeCyclic[k], 2]), NumberQ];
tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@ xp;
bterms = Total[b /@ NestList[f, #, k - 1]] & /@ xp;
tfactors ≠ {}, tfactors = Rest[tfactors];
bterms = Rest[bterms], da += Take[PadLeft[da First[tfactors], n + k], n] -
Take[PadLeft[First[bterms] First[tfactors] ζ, n + k - 1], n];
na += Take[PadLeft[na First[tfactors], n + k], n] -
Take[PadLeft[k First[tfactors] ζ, n + k - 1], n];
ζ += Take[PadLeft[ζ First[tfactors], n + k], n], {k, n}];
Accumulate[da]
Accumulate[na]]

```

```

In[42]:= dynAvgDZ[b_, f_, if_, fmx_, n_] :=
Module[{iter = 100, k, ζ, da, na, xp, tfactors, bterms},
  ζ = Prepend[ConstantArray[0, n - 1], 1];
  na = da = ConstantArray[0, n];
  Do[For[xp = Select[(unimodalPO[if, #, fmx, iter] & /@
    IntegerDigits[discardZero[k], 2]), NumberQ];
    tfactors = -  $\frac{1}{\text{Abs}[\text{monodromy}[f, \#, k]]}$  & /@ xp;
    bterms = Total[b /@ NestList[f, #, k - 1]] & /@ xp;
    tfactors ≠ {}, tfactors = Rest[tfactors];
    bterms = Rest[bterms], da += Take[PadLeft[da First[tfactors], n + k], n] -
      Take[PadLeft[First[bterms] First[tfactors] ζ, n + k - 1], n];
    na += Take[PadLeft[na First[tfactors], n + k], n] -
      Take[PadLeft[k First[tfactors] ζ, n + k - 1], n];
    ζ += Take[PadLeft[ζ First[tfactors], n + k], n], {k, n}];
  Accumulate[da]
  Accumulate[na]

```

```

In[43]:=

```

```

In[44]:= survIntervals[if_, n_] :=
Partition[#, 2] &@ (Sort@ (Fold[if[#2, #1] &, 0., #] & /@ Tuples[{0, 1}, n]))

```

```

In[45]:= itinerary[bits_List] := Module[{w = bits},
Do[If[w[[i]] == 1, w[[i + 1 ;;]] = 1 - w[[i + 1 ;;]], {i, Length[w]}];
w]

```

```

In[46]:= survPlot[f_, if_, ng_, ns_] :=
Show[Plot[f[x], {x, 0, 1}, PlotStyle → Black], NumberLinePlot[
  Table[Interval@@survIntervals[if, d], {d, ng}], Spacings → {-0.2}],
  Table[ListPlot[Partition[NestList[f, unimodalPO[if, #, 100], k], 2, 1] & /@
    IntegerDigits[removeCyclic[k], 2], Filling → -1,
  PlotStyle → ColorData[97][k + 1]], {k, 1, ns}], Graphics[Table[MapThread[
  Text, {StringJoin /@ Map[ToString, itinerary /@ Tuples[{0, 1}, k], {2}],
  Outer[Sequence, Mean /@ survIntervals[if, k + 1], {-.2 k - .15}]}],
  {k, ns}], PlotRange → {-0.2 (ng + 1), 0.9}]

```

```

In[47]:= rectPlot[f_, if_, ng_, ns_] :=
Show[Plot[f[x], {x, 0, 1}, PlotStyle → Directive[Thin]], NumberLinePlot[
  Table[Interval@@survIntervals[if, d], {d, ng}], Spacings → {-0.2}],
  Table[ListPlot[Partition[NestList[f, unimodalPO[if, #, 100], k], 2, 1] & /@
    IntegerDigits[removeCyclic[k], 2],
  Filling → -1, PlotStyle → ColorData[97][k + 1]], {k, 1, ns}],
  Graphics[Table[{ColorData[97][k + 1], Line /@
    (tangent[f, #,  $\frac{0.5}{2^k}$ ] & /@ Partition[NestList[f, unimodalPO[if, #, 100], k],
      2, 1] & /@ IntegerDigits[removeCyclic[k], 2])}], {k, 1, ns}],
  Graphics[Table[MapThread[Text, {StringJoin /@ Map[ToString, itinerary /@
    Tuples[{0, 1}, k], {2}], Outer[Sequence, Mean /@ survIntervals[if, k + 1],
    {-.2 k - .15}]}], {k, ns}], PlotRange → {-0.2 (ng + 1), 0.9}]

```

```

In[48]:= pruneIntervals[if_, fmx_, n_] :=
Catch[#, Fold[imapInterval[if, #2, #1, fmx] &, {0, fmx}, #]] & /@
  Tuples[{0, 1}, n - 1]

```

```
In[49]:= prunePlot[f_, if_, fmx_, ng_, ns_] :=
Module[{symbols, intervals, po}, Do[{symbols[k], intervals[k]} =
Transpose[SortBy[#[[2, 1]] &]@Cases[pruneIntervals[if, fmx, k], _List]];
po[k - 1] = Select[unimodalPO[if, #, fmx, 100] & /@
IntegerDigits[removeCyclic[k - 1], 2], NumberQ], {k, 2, ng}];
Show[Plot[f[x], {x, 0, 1}, PlotStyle -> Black], NumberLinePlot[
Table[Interval /@ intervals[k], {k, 2, ng}], Spacings -> {-0.2}], Table[
ListPlot[Partition[NestList[f, #, k], 2, 1] & /@ po[k], Filling -> -0.2 (ng - 1),
PlotStyle -> ColorData[97][k + 1]], {k, 1, ns}], Graphics[Table[
MapThread[Text, {StringJoin /@ Map[ToString, Reverse /@ symbols[k + 1], {2}],
Outer[Sequence, Mean /@ intervals[k + 1], {- .2 k + .05}]}],
{k, ns}]], PlotRange -> {-0.2 (ng), 1.0}]]
```

```
In[50]:=
```

```
In[51]:= lo[x_] := {σ (-x[1] + x[2]), r x[1] - x[2] - x[1] x[3], x[1] x[2] - b x[3]}
```

```
In[52]:= lofp = Solve[MapThread[#1 == #2 &, {lo[x], {0, 0, 0}}], Array[x, 3]]
```

```
Out[52]= {{x[1] -> 0, x[2] -> 0, x[3] -> 0},
{x[1] -> -√b √(-1+r), x[2] -> -√b √(-1+r), x[3] -> -1+r},
{x[1] -> √b √(-1+r), x[2] -> √b √(-1+r), x[3] -> -1+r}}
```

```
In[53]:= lojac = Grad[#, Array[x, 3]] & /@ lo[x]
```

```
Out[53]= {{-σ, σ, 0}, {r - x[3], -1, -x[1]}, {x[2], x[1], -b}}
```

```
In[54]:= lopar = {σ -> 10, b ->  $\frac{8}{3}$ , r -> 28};
```

```
In[55]:= unstable = Drop[Eigenvectors[lojac /. lopar] /. N[lopar], 1]
```

```
Out[55]= {{-0.370014 + 0.410184 i, 0.0446719 + 0.79125 i, 1},
{-0.370014 - 0.410184 i, 0.0446719 - 0.79125 i, 1}}
```

```
In[56]:= us1 = Chop[-i (Subtract @@ unstable)]
```

```
Out[56]= {0.820369, 1.5825, 0}
```

```

In[57]:= usmPlus = Module[{σ = 10, b =  $\frac{8}{3}$ , r = 28, fpPos, nudge = 0.01, us1, td = 65., usm,
  refine, usm1, usm2, nr = 0}, fpPos = { $\sqrt{b(r-1)}$ ,  $\sqrt{b(r-1)}$ , r - 1};
  us1 = Chop[-i (Subtract @@ unstable)];
  usm = Select[Reap[NDSolve[{x'[t] == σ(-x[t] + y[t]), y'[t] == r x[t] - y[t] - x[t]
    z[t], z'[t] == x[t] y[t] - b z[t], x[0] == fpPos[[1]] + nudge us1[[1]],
    y[0] == fpPos[[2]] + nudge us1[[2]], z[0] == fpPos[[3]], WhenEvent[
    z[t] == r - 1, Sow[{x[t], y[t]} - Take[fpPos, 2]]], WhenEvent[x[t] < 0,
    "StopIntegration"]}], {}, {t, 0, td}][[-1, 1]], Positive[Total[#]] &];
  While[nr++ < 450 && (refine = First@FirstPosition[Total[#[[2]] - #[[1]]] & /@
    Partition[usm, 2, 1], _? (# > 0.01 &), {"None"}]) ≠ "None", usm1 =
    Select[Reap[NDSolve[{x'[t] == σ(-x[t] + y[t]), y'[t] == r x[t] - y[t] - x[t]
      z[t], z'[t] == x[t] y[t] - b z[t], x[0] == fpPos[[1]] +  $\frac{1}{2}$ (usm[[refine -
        nr + 1, 1]] + usm[[refine - nr, 1]]), y[0] == fpPos[[2]] +  $\frac{1}{2}$ (usm[[refine -
          nr + 1, 2]] + usm[[refine - nr, 2]]), z[0] == fpPos[[3]], WhenEvent[
          z[t] == r - 1, Sow[{x[t], y[t]} - Take[fpPos, 2]]], WhenEvent[x[t] < 0,
          "StopIntegration"]}], {}, {t, 0, td}][[-1, 1]], Positive[Total[#]] &];
  usm = Sort@Join[usm, usm1];
  usm];

```

```

In[58]:= iUsmPlus = Interpolation[usmPlus]

```

```

Out[58]= InterpolatingFunction[ Domain{{0.008698, 0.1}}
  Output: scalar

```

```

In[59]:= lomapt[xi_?NumericQ] := Module[{σ = 10, b =  $\frac{8}{3}$ , r = 28, fpPos, td = 65.,
  lx, ly, lz, gr, usm, refine, usm1, usm2, nr = 0, yi}, yi = iUsmPlus[xi];
  fpPos = { $\sqrt{b(r-1)}$ ,  $\sqrt{b(r-1)}$ , r - 1};
  Reap[NDSolve[{x'[t] == σ(-x[t] + y[t]), y'[t] == r x[t] - y[t] - x[t] z[t],
    z'[t] == x[t] y[t] - b z[t], x[0] == fpPos[[1]] + xi, y[0] == fpPos[[2]] + yi,
    z[0] == fpPos[[3]], WhenEvent[z[t] == r - 1 && (x[t] > fpPos[[1]] || x[t] <
    -fpPos[[1]]), Sow[If[x[t] > 0, x[t] - fpPos[[1]], -x[t] - fpPos[[1]]]]],
    "StopIntegration"]], {}, {t, 0, td}][[-1, 1, 1]]

```

```

In[60]:= lomapt[xi_?NumericQ] :=
  Module[{σ = 10, b =  $\frac{8}{3}$ , r = 28, fpPos, td = 65., lx, ly, lz, gr,
    usm, refine, usm1, usm2, nr = 0, yi}, yi = iUsmPlus[xi];
  fpPos = { $\sqrt{b(r-1)}$ ,  $\sqrt{b(r-1)}$ , r - 1};
  Reap[NDSolve[{x'[t] == σ(-x[t] + y[t]),
    y'[t] == r x[t] - y[t] - x[t] z[t], z'[t] == x[t] y[t] - b z[t],
    x[0] == fpPos[[1]] + xi, y[0] == fpPos[[2]] + yi, z[0] == fpPos[[3]],
    WhenEvent[z[t] == r - 1 && (x[t] > fpPos[[1]] || x[t] < -fpPos[[1]]), Sow[t];
    "StopIntegration"]}], {}, {t, 0, td}][[-1, 1, 1]]

```

```
In[61]:= switchRate[td_, xi_] :=
Module[{σ = 10, b =  $\frac{8}{3}$ , r = 28, fpPos, lx, ly, lz, gr, nr = 0, yi}, yi = iUsmPlus[xi];
fpPos = { $\sqrt{b(r-1)}$ ,  $\sqrt{b(r-1)}$ , r-1};
NDSolve[{x'[t] == σ(-x[t] + y[t]), y'[t] == r x[t] - y[t] - x[t] z[t],
z'[t] == x[t] y[t] - b z[t], x[0] == fpPos[[1]] + xi, y[0] == fpPos[[2]] + yi,
z[0] == fpPos[[3]], WhenEvent[x[t] == -y[t], nr++]}, {}, {t, 0, td}];
 $\frac{nr}{td}$ ]
```

```
In[62]:= lolyapunov[td_, xi_] := Module[
{σ = 10, b =  $\frac{8}{3}$ , r = 28, fpPos, lx, ly, lz, lv, lλ, gr, yi, lor}, yi = iUsmPlus[xi];
fpPos = { $\sqrt{b(r-1)}$ ,  $\sqrt{b(r-1)}$ , r-1};
{lx, ly, lz, lv, lλ} = NDSolveValue[{x'[t] == σ(-x[t] + y[t]),
y'[t] == r x[t] - y[t] - x[t] z[t], z'[t] == x[t] y[t] - b z[t], λ'[t] ==
Log[Abs[v[t].{{-σ, σ, 0}, {r-z[t], -1, -x[t]}, {y[t], x[t], -b}}.v[t]]],
v'[t] == ({{-σ, σ, 0}, {r-z[t], -1, -x[t]}, {y[t], x[t], -b}} -
v[t].{{-σ, σ, 0}, {r-z[t], -1, -x[t]}, {y[t], x[t], -b}}).v[t],
IdentityMatrix[3]).v[t], x[0] == fpPos[[1]] + xi, y[0] == fpPos[[2]] + yi,
z[0] == fpPos[[3]], λ[0] == 0, v[0] == {1, 0, 0}}, {x, y, z, v, λ}, {t, 0, td}];
 $\frac{lλ[td]}{td}$ ]
```

```
In[63]:= ilomap = ListInterpolation[Array[lomap, 801, {0.0, 8}],
{0.0, 8}, InterpolationOrder → 3, Method → "Spline"]
```

InterpolatingFunction::mval:

Input value (0.) lies outside the range of data in the interpolating function. Extrapolation will be used >>

```
Out[63]:= InterpolatingFunction[ Domain {{0., 8.}}
Output scalar]
```

```
In[64]:= ilomapt = ListInterpolation[Array[lomapt, 801, {0.0, 8}],
{0.0, 8}, InterpolationOrder → 3, Method → "Spline"]
```

InterpolatingFunction::mval:

Input value (0.) lies outside the range of data in the interpolating function. Extrapolation will be used >>

```
Out[64]:= InterpolatingFunction[ Domain {{0., 8.}}
Output scalar]
```

```
In[65]:= iilomap[0] = Interpolation@Array[{lomap[#], #} &, 477, {0.0, 4.76}]
```

InterpolatingFunction::mval:

Input value (0.) lies outside the range of data in the interpolating function. Extrapolation will be used >>

```
Out[65]:= InterpolatingFunction[ Domain {{3.56×10-7, 7.85}}
Output scalar]
```

```
In[66]:= iilomap[1] = Interpolation@Array[{lomap[#], #} &, 800 - 475, {4.76, 8.}]
```

```
Out[66]:= InterpolatingFunction[ Domain {{1.41, 7.89}}
Output scalar]
```

```
In[67]:= lomin = 1.41; lomax = 7.84;
```