

Chapter 12

Fixed points, and how to get them

HAVING SET UP the dynamical context, now we turn to the key and unavoidable piece of numerics in this subject; search for the solutions (x, T) , $x \in \mathbb{R}^d$, $T \in \mathbb{R}$ of the *periodic orbit condition*

$$f^{t+T}(x) = f^t(x), \quad T > 0 \tag{12.1}$$

for a given flow or mapping.

We know from chapter 16 that cycles are the necessary ingredient for evaluation of spectra of evolution operators. In chapter 10 we have developed a qualitative theory of how these cycles are laid out topologically.

This chapter is intended as a hands-on guide to extraction of periodic orbits, and should be skipped on first reading - you can return to it whenever the need for finding actual cycles arises. Sadly, searching for periodic orbits will never become as popular as a week on Côte d'Azur, or publishing yet another log-log plot in *Phys. Rev. Letters*. A serious cyclist will want to also learn about the variational methods to find cycles, chapter 27. They are particularly useful when little is understood about the topology of a flow, such as in high-dimensional periodic orbit searches.

[chapter 27]



fast track:
chapter 13, p. 212

A *prime cycle* p of period T_p is a single traversal of the periodic orbit, so our task will be to find a cycle point $x \in p$ and the shortest time T_p for which (12.1) has a solution. A cycle point of a flow f^t which crosses a Poincaré section n times is a fixed point of the P^n iterate of P , the return map (3.1), hence we shall refer to all cycles as “fixed points” in this chapter. By cyclic invariance, stability eigenvalues and the period of the cycle are independent of the choice of the initial point, so it will suffice to solve (12.1) at a single cycle point.

[section 5.2]

If the cycle is an attracting limit cycle with a sizable basin of attraction, it can be found by integrating the flow for sufficiently long time. If the cycle is unstable, simple integration forward in time will not reveal it, and methods to be described here need to be deployed. In essence, any method for finding a cycle is based on devising a new dynamical system which possesses the same cycle, but for which this cycle is attractive. Beyond that, there is a great freedom in constructing such systems, and many different methods are used in practice.

Due to the exponential divergence of nearby trajectories in chaotic dynamical systems, fixed point searches based on direct solution of the fixed-point condition (12.1) as an initial value problem can be numerically very unstable. Methods that start with initial guesses for a number of points along the cycle, such as the multipoint shooting method described here in sect. 12.3, and the variational methods of chapter 27, are considerably more robust and safer.

[chapter 27]

A prerequisite for any exhaustive cycle search is a good understanding of the topology of the flow: a preliminary step to any serious periodic orbit calculation is preparation of a list of all distinct admissible prime periodic symbol sequences, such as the list given in table 10.1. The relations between the temporal symbol sequences and the spatial layout of the topologically distinct regions of the state space discussed in chapters 10 and 11 should enable us to guess location of a series of periodic points along a cycle. Armed with such informed guess we proceed to improve it by methods such as the Newton-Raphson iteration; we show how this works by applying the Newton method to 1- and d -dimensional maps. But first, where are the cycles?

12.1 Where are the cycles?

Q: What if you choose a really bad initial condition and it doesn't converge? A: Well then you only have yourself to blame.

— T.D. Lee

The simplest and conceptually easiest setting for guessing where the cycles are is the case of planar billiards. The Maupertuis principle of least action here dictates that the physical trajectories extremize the length of an approximate orbit that visits a desired sequence of boundary bounces.

Example 12.1 Periodic orbits of billiards. Consider how this works for 3-disk pinball game of sect. 11.1. Label the three disks by 1, 2 and 3, and associate to every trajectory an itinerary, a sequence of labels indicating the order in which the disks are visited, as in figure 3.2. Given the itinerary, you can construct a guess trajectory by taking a point on the boundary of each disk in the sequence, and connecting them by straight lines. Imagine that this is a rubber band wrapped through 3 rings, and shake the band until it shrinks into the physical trajectory, the rubber band of shortest length.

[section 11.1]

[section 1.4]

Extremization of a cycle length requires variation of n bounce positions s_i . The computational problem is to find the extremum values of cycle length $L(s)$ where $s = (s_1, \dots, s_n)$, a task that we postpone to sect. 27.3. As an example, the short

[exercise 27.2]

[exercise 12.10]

periods and stabilities of 3-disk cycles computed this way are listed table 27.2, and some examples are plotted in figure 3.2. It's a no brainer, and millions of such cycles have been computed.

If we were only so lucky. Real life finds us staring at something like Yang-Mills or Navier-Stokes equations, utterly clueless. What to do?

One, there is always mindless computation. In practice one might be satisfied with any rampaging robot that finds “the most important” cycles. Ergodic exploration of recurrences that we turn to next sometimes perform admirably well.

12.1.1 Cycles from long time series

Two wrongs don't make a right, but three lefts do.
—Appliance guru

(L. Rondoni and P. Cvitanović)

The equilibria and periodic orbits (with the exception of sinks and stable limit cycles) are never seen in simulations and experiments because they are unstable. [remark 12.1] Nevertheless, one does observe close passes to the least unstable equilibria and periodic orbits. Ergodic exploration by long-time trajectories (or long-lived transients, in case of strange repellers) can uncover state space regions of low velocity, or finite time recurrences. In addition, such trajectories preferentially sample the natural measure of the ‘turbulent’ flow, and by initiating searches within the state space concentrations of natural measure bias the search toward the dynamically important invariant solutions. [section 14.1]

The search consists of following a long trajectory in state space, and looking for close returns of the trajectory to itself. Whenever the trajectory almost closes in a loop (within a given tolerance), another point of this near miss of a cycle can be taken as an initial condition. Supplemented by a Newton routine described below, a sequence of improved initial conditions may indeed rapidly lead to closing a cycle. The method preferentially finds the least unstable orbits, while missing the more unstable ones that contribute little to the cycle expansions.

This blind search is seriously flawed: in contrast to the 3-disk example 12.1, it is not systematic, it gives no insight into organization of the ergodic sets, and can easily miss very important cycles. Foundations to a systematic exploration of ergodic state space are laid in chapters 10 and 11, but are a bit of work to implement.

12.1.2 Cycles found by thinking

Thinking is extra price.
—Argentine saying

A systematic charting out of state space starts out by a hunt for equilibrium points. If the equations of motion are a finite set of ODEs, setting the velocity field $v(x)$ in (2.6) to zero reduces search for equilibria to a search for zeros of a set of algebraic equations. We should be able, in principle, to enumerate and determine all real and complex zeros in such cases, e.g. the Lorenz example 2.2 and the Rössler example 2.3. If the equations of motion and the boundary conditions are invariant under some symmetry, some equilibria can be determined by symmetry considerations: if a function is e.g. antisymmetric, it must vanish at origin, e.g. the Lorenz $EQ_0 = (0, 0, 0)$ equilibrium.

As to other equilibria: if you have no better idea, create a state space grid, about 50 x_k across \mathcal{M} in each dimension, and compute the velocity field $v_k = v(x_k)$ at each grid point; a few million v_k values are easily stored. Plot x_k for which $|v_k|^2 < \epsilon$, $\epsilon \ll |v_{max}|^2$ but sufficiently large that a few thousand x_k are plotted. If the velocity field varies smoothly across the state space, the regions $|v_k|^2 < \epsilon$ isolate the (candidate) equilibria. Start a Newton iteration with the smallest $|v_k|^2$ point within each region. Barring exceptionally fast variations in $v(x)$ this should yield all equilibrium points.

For ODEs equilibria are fixed points of algebraic sets of equations, but steady states of PDEs such as the Navier-Stokes flow are themselves solutions of ODEs or PDEs, and much harder to determine.

Equilibria—by definition—do not move, so they cannot be “turbulent.” What makes them dynamically important are their stable/unstable manifolds. A chaotic trajectory can be thought of as a sequence of near visitations of equilibria. Typically such neighborhoods have many stable, contracting directions and a handful of unstable directions. Our strategy will be to generalize the billiard Poincaré section maps $P_{s_{n+1} \leftarrow s_n}$ of example 3.2 to maps from a section of the unstable manifold of equilibrium s_n to the section of unstable manifold of equilibrium s_{n+1} , and thus reduce the continuous time flow to a sequence of maps. These Poincaré section maps do double duty, providing us both with an exact representation of dynamics in terms of maps, and with a covering symbolic dynamics.

invariant

We showed in the Lorenz flow example 10.5 how to reduce the 3-dimensional Lorenz flow to a 1- d return map.

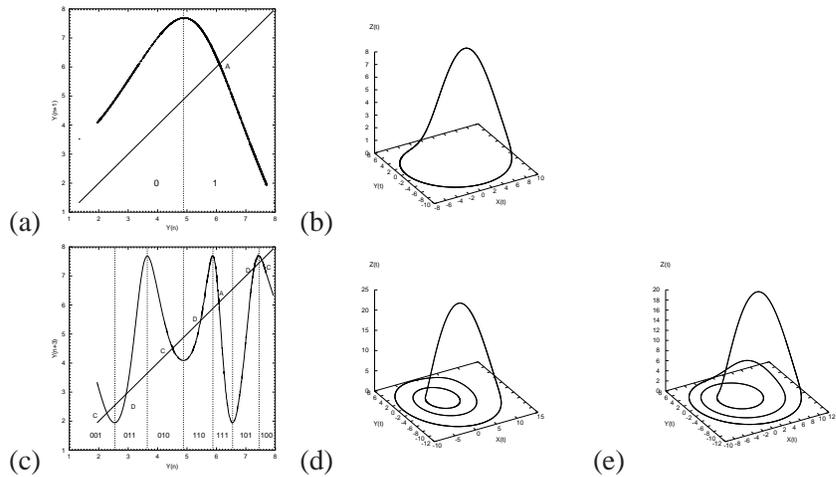
In the Rössler flow example 2.3 we sketched the attractor by running a long chaotic trajectory, and noted that the attractor is very thin, but otherwise the return maps that we plotted were disquieting – figure 3.6 did not appear to be a 1-to-1 map. In the next example we show how to use such information to approximately locate cycles. In the remainder of this chapter and in chapter 27 we shall learn how to turn such guesses into highly accurate cycles.

Example 12.2 Rössler attractor

(G. Simon and P. Cvitanović)

Run a long simulation of the Rössler flow f^t , plot a Poincaré section, as in figure 3.5, and extract the corresponding Poincaré return map P , as in figure 3.6. Luck is with us; the figure 12.1 (a) return map $y \rightarrow P_1(y, z)$ looks much like a parabola, so we

Figure 12.1: (a) $y \rightarrow P_1(y, z)$ return map for $x = 0, y > 0$ Poincaré section of the Rössler flow figure 2.5. (b) The $\bar{1}$ -cycle found by taking the fixed point $y_{k+n} = y_k$ together with the fixed point of the $z \rightarrow z$ return map (not shown) an initial guess $(0, y^{(0)}, z^{(0)})$ for the Newton-Raphson search. (c) $y_{k+3} = P_1^3(y_k, z_k)$, the third iterate of Poincaré return map (3.1) together with the corresponding plot for $z_{k+3} = P_2^3(y_k, z_k)$, is used to pick starting guesses for the Newton-Raphson searches for the two 3-cycles: (d) the 001 cycle, and (e) the 011 cycle. (G. Simon)



take the unimodal map symbolic dynamics, sect. 10.2.1, as our guess for the covering dynamics. Strictly speaking, the attractor is “fractal,” but for all practical purposes the return map is 1-dimensional; your printer will need a resolution better than 10^{14} dots per inch to start resolving its structure.

Periodic points of a prime cycle p of cycle length n_p for the $x = 0, y > 0$ Poincaré section of the Rössler flow figure 2.5 are fixed points $(y, z) = P^{n_p}(y, z)$ of the n th Poincaré return map.

Using the fixed point $y_{k+1} = y_k$ in figure 12.1 (a) together with the simultaneous fixed point of the $z \rightarrow P_1(y, z)$ return map (not shown) as a starting guess $(0, y^{(0)}, z^{(0)})$ for the Newton-Raphson search for the cycle p with symbolic dynamics label $\bar{1}$, we find the cycle figure 12.1 (b) with the Poincaré section point $(0, y_p, z_p)$, period T_p , expanding, marginal, contracting stability eigenvalues $(\Lambda_{p,e}, \Lambda_{p,m}, \Lambda_{p,c})$, and Lyapunov exponents $(\lambda_{p,e}, \lambda_{p,m}, \lambda_{p,c})$:

[exercise 12.7]

$$\begin{aligned}
 \bar{1}\text{-cycle:} \quad (x, y, z) &= (0, 6.09176832, 1.2997319) \\
 T_1 &= 5.88108845586 \\
 (\Lambda_{1,e}, \Lambda_{1,m}, \Lambda_{1,c}) &= (-2.40395353, 1 + 10^{-14}, -1.29 \times 10^{-14}) \\
 (\lambda_{1,e}, \lambda_{1,m}, \lambda_{1,c}) &= (0.149141556, 10^{-14}, -5.44). \tag{12.2}
 \end{aligned}$$

The Newton-Raphson method that we used is described in sect. 12.4.

As an example of a search for longer cycles, we use $y_{k+3} = P_1^3(y_k, z_k)$, the third iterate of Poincaré return map (3.1) plotted in figure 12.1 (c), together with a corresponding plot for $z_{k+3} = f^3(y_k, z_k)$, to pick starting guesses for the Newton-Raphson searches for the two 3-cycles plotted in figure 12.1 (d), (e). For a listing of the short cycles of the Rössler flow, consult exercise 12.7.

The numerical evidence suggests (but a proof is lacking) that all cycles that comprise the strange attractor of the Rössler flow are hyperbolic, each with an expanding eigenvalue $|\Lambda_e| > 1$, a contracting eigenvalue $|\Lambda_c| < 1$, and a marginal eigenvalue $|\Lambda_m| = 1$ corresponding to displacements along the direction of the flow.

For the Rössler flow the contracting eigenvalues turn out to be insanely contracting, a factor of e^{-32} per one par-course of the attractor, so their numerical determination is quite difficult. Fortunately, they are irrelevant; for all practical purposes the strange attractor of the Rössler flow is 1-dimensional, a very good realization of a horseshoe template.

Figure 12.2: The inverse time path to the $\overline{01}$ -cycle of the logistic map $f(x) = 4x(1-x)$ from an initial guess of $x = 0.2$. At each inverse iteration we chose the 0, respectively 1 branch.

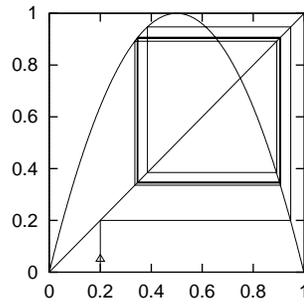
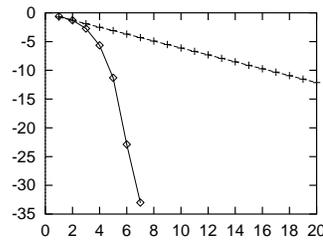


Figure 12.3: Convergence of Newton method (\diamond) vs. inverse iteration ($+$). The error after n iterations searching for the $\overline{01}$ -cycle of the logistic map $f(x) = 4x(1-x)$ with an initial starting guess of $x_1 = 0.2, x_2 = 0.8$. y-axis is \log_{10} of the error. The difference between the exponential convergence of the inverse iteration method and the super-exponential convergence of Newton method is dramatic.



12.2 One-dimensional mappings

(F. Christiansen)

12.2.1 Inverse iteration

Let us first consider a very simple method to find unstable cycles of a 1-dimensional map such as the logistic map. Unstable cycles of 1- d maps are attracting cycles of the inverse map. The inverse map is not single valued, so at each backward iteration we have a choice of branch to make. By choosing branch according to the symbolic dynamics of the cycle we are trying to find, we will automatically converge to the desired cycle. The rate of convergence is given by the stability of the cycle, i.e., the convergence is exponentially fast. Figure 12.2 shows such path to the $\overline{01}$ -cycle of the logistic map.

[exercise 12.10]

The method of inverse iteration is fine for finding cycles for 1-d maps and some 2- d systems such as the repeller of exercise 12.10. It is not particularly fast, especially if the inverse map is not known analytically. However, it completely fails for higher dimensional systems where we have both stable and unstable directions. Inverse iteration will exchange these, but we will still be left with both stable and unstable directions. The best strategy is to directly attack the problem of finding solutions of $f^T(x) = x$.

12.2.2 Newton method

Newton method for determining a zero x^* of a function $F(x)$ of one variable is based on a linearization around a starting guess x_0 :

$$F(x) \approx F(x_0) + F'(x_0)(x - x_0). \quad (12.3)$$

An approximate solution x_1 of $F(x) = 0$ is

$$x_1 = x_0 - F(x_0)/F'(x_0). \quad (12.4)$$

The approximate solution can then be used as a new starting guess in an iterative process. A fixed point of a map f is a solution to $F(x) = x - f(x) = 0$. We determine x by iterating

$$\begin{aligned} x_m &= g(x_{m-1}) = x_{m-1} - F(x_{m-1})/F'(x_{m-1}) \\ &= x_{m-1} - \frac{1}{1 - f'(x_{m-1})}(x_{m-1} - f(x_{m-1})). \end{aligned} \quad (12.5)$$

Provided that the fixed point is not marginally stable, $f'(x) \neq 1$ at the fixed point x , a fixed point of f is a super-stable fixed point of the Newton-Raphson map g , $g'(x) = 0$, and with a sufficiently good initial guess, the Newton-Raphson iteration will converge super-exponentially fast.

To illustrate the efficiency of the Newton method we compare it to the inverse iteration method in figure 12.3. Newton method wins hands down: the number of significant digits of the accuracy of x estimate doubles with each iteration.

In order to avoid jumping too far from the desired x^* (see figure 12.4), one often initiates the search by the *damped Newton method*,

$$\Delta x_m = x_{m+1} - x_m = -\frac{F(x_m)}{F'(x_m)} \Delta\tau, \quad 0 < \Delta\tau \leq 1,$$

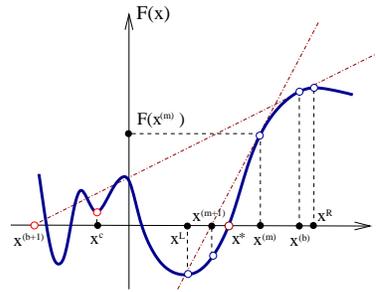
takes small $\Delta\tau$ steps at the beginning, reinstating to the full $\Delta\tau = 1$ jumps only when sufficiently close to the desired x^* .

12.3 Multipoint shooting method

(F. Christiansen)

Periodic orbits of length n are fixed points of f^n so in principle we could use the simple Newton method described above to find them. However, this is not an optimal strategy. f^n will be a highly oscillating function with perhaps as many as 2^n or more closely spaced fixed points, and finding a specific periodic point, for example one with a given symbolic sequence, requires a *very* good starting guess. For binary symbolic dynamics we must expect to improve the accuracy of our initial guesses by at least a factor of 2^n to find orbits of length n . A better alternative is the *multipoint shooting method*. While it might very hard to give a precise initial point guess for a long periodic orbit, if our guesses are informed by a good state space partition, a rough guess for each point along the desired trajectory might suffice, as for the individual short trajectory segments the errors have no time to explode exponentially.

Figure 12.4: Newton method: bad initial guess $x^{(b)}$ leads to the Newton estimate $x^{(b+1)}$ far away from the desired zero of $F(x)$. Sequence $\dots, x^{(m)}, x^{(m+1)}, \dots$, starting with a good guess converges super-exponentially to x^* . The method diverges if it iterates into the basin of attraction of a local minimum x^c .



A cycle of length n is a zero of the n -dimensional vector function F :

$$F(x) = F \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 - f(x_n) \\ x_2 - f(x_1) \\ \dots \\ x_n - f(x_{n-1}) \end{pmatrix}.$$

The relations between the temporal symbol sequences and the spatial layout of the topologically distinct regions of the state space discussed in chapter 10 enable us to guess location of a series of periodic points along a cycle. Armed with such informed initial guesses we can initiate a Newton-Raphson iteration. The iteration in the Newton method now takes the form of

$$\frac{d}{dx}F(x)(x' - x) = -F(x), \tag{12.6}$$

where $\frac{d}{dx}F(x)$ is an $[n \times n]$ matrix:

$$\frac{d}{dx}F(x) = \begin{pmatrix} 1 & & & & -f'(x_n) \\ -f'(x_1) & 1 & & & \\ & \dots & & & \\ & & 1 & & \\ & & & \dots & \\ & & & & 1 \\ -f'(x_{n-1}) & & & & 1 \end{pmatrix}. \tag{12.7}$$

This matrix can easily be inverted numerically by first eliminating the elements below the diagonal. This creates non-zero elements in the n th column. We eliminate these and are done.

Example 12.3 Newton inversion for a 3-cycle. Let us illustrate how this works step by step for a 3-cycle. The initial setup for a Newton step is:

$$\begin{pmatrix} 1 & 0 & -f'(x_3) \\ -f'(x_1) & 1 & 0 \\ 0 & -f'(x_2) & 1 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{pmatrix} = - \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix},$$

where $\Delta x_i = x'_i - x_i$ is the correction to our initial guess x_i , and $F_i = x_i - f(x_{i-1})$ is the error at i th cycle point. Eliminate the sub-diagonal elements by adding $f'(x_1)$ times the first row to the second row, then adding $f'(x_2)$ times the second row to the third row:

$$\begin{pmatrix} 1 & 0 & -f'(x_3) \\ 0 & 1 & -f'(x_1)f'(x_3) \\ 0 & 0 & 1 - f'(x_2)f'(x_1)f'(x_3) \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{pmatrix} = - \begin{pmatrix} F_1 \\ F_2 + f'(x_1)F_1 \\ F_3 + f'(x_2)F_2 + f'(x_2)f'(x_1)F_1 \end{pmatrix}.$$

The next step is to invert the last element in the diagonal, i.e., divide the third row by $1 - f'(x_2)f'(x_1)f'(x_3)$. If this element is zero at the periodic orbit this step cannot work. As $f'(x_2)f'(x_1)f'(x_3)$ is the stability of the cycle (when the Newton iteration has converged), this is not a good method to find marginally stable cycles. We now have

$$\begin{pmatrix} 1 & 0 & -f'(x_3) \\ 0 & 1 & -f'(x_1)f'(x_3) \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{pmatrix} = - \begin{pmatrix} F_1 \\ F_2 + f'(x_1)F_1 \\ \frac{F_3 + f'(x_2)F_2 + f'(x_2)f'(x_1)F_1}{1 - f'(x_2)f'(x_1)f'(x_3)} \end{pmatrix}.$$

Finally we add $f'(x_3)$ times the third row to the first row and $f'(x_1)f'(x_3)$ times the third row to the second row. The left hand side matrix is now the unit matrix, the right hand side is an explicit formula for the corrections to our initial guess. We have gone through one Newton iteration.

When one sets up the Newton iteration on the computer it is not necessary to write the left hand side as a matrix. All one needs is a vector containing the $f'(x_i)$'s, a vector containing the n 'th column, i.e., the cumulative product of the $f'(x_i)$'s, and a vector containing the right hand side. After the iteration the vector containing the right hand side should be the correction to the initial guess.

[exercise 12.1]

12.3.1 d -dimensional mappings



Armed with clever, symbolic dynamics informed initial guesses we can easily extend the Newton-Raphson iteration method to d -dimensional mappings. In this case $f'(x_i)$ is a $[d \times d]$ matrix, and $\frac{d}{dx}F(x)$ is an $[nd \times nd]$ matrix. In each of the steps that we went through above we are then manipulating d rows of the left hand side matrix. (Remember that matrices do not commute - always multiply from the left.) In the inversion of the n th element of the diagonal we are inverting a $[d \times d]$ matrix $(1 - \prod f'(x_i))$ which can be done if none of the eigenvalues of $\prod f'(x_i)$ equals 1, i.e., if the cycle has no marginally stable eigen-directions.

Example 12.4 Newton method for time delay maps. Some d -dimensional mappings (such as the Hénon map (3.18)) can be written as 1-dimensional time delay mappings of the form

$$f(x_i) = f(x_{i-1}, x_{i-2}, \dots, x_{i-d}). \quad (12.8)$$

In this case $\frac{d}{dx}F(x)$ is an $[n \times n]$ matrix as in the case of usual 1-dimensional maps but with non-zero matrix elements on d off-diagonals. In the elimination of these off-diagonal elements the last d columns of the matrix will become non-zero and in the final cleaning of the diagonal we will need to invert a $[d \times d]$ matrix. In this respect, nothing is gained numerically by looking at such maps as 1-dimensional time delay maps.

12.4 Flows

(F. Christiansen)

Further complications arise for flows due to the fact that for a periodic orbit the stability eigenvalue corresponding to the flow direction of necessity equals unity; the separation of any two points along a cycle remains unchanged after a completion of the cycle. More unit eigenvalues can arise if the flow satisfies conservation laws, such as the energy invariance for Hamiltonian systems. We now show how such problems are solved by increasing the number of fixed point conditions.

[section 5.2.1]

12.4.1 Newton method for flows

A flow is equivalent to a mapping in the sense that one can reduce the flow to a mapping on the Poincaré surface of section. An autonomous flow (2.6) is given as

$$\dot{x} = v(x), \quad (12.9)$$

The corresponding fundamental matrix M (4.43) is obtained by integrating the linearized equation (4.9)

$$\dot{M} = \mathbf{A}M, \quad A_{ij}(x) = \frac{\partial v_i(x)}{\partial x_j}$$

along the trajectory. The flow and the corresponding fundamental matrix are integrated simultaneously, by the same numerical routine. Integrating an initial condition on the Poincaré surface until a later crossing of the same and linearizing around the flow we can write

$$f(x') \approx f(x) + M(x' - x). \quad (12.10)$$

Notice here, that, even though all of x' , x and $f(x)$ are on the Poincaré surface, $f(x')$ is usually not. The reason for this is that M corresponds to a specific integration time and has no explicit relation to the arbitrary choice of Poincaré section. This will become important in the extended Newton method described below.

To find a fixed point of the flow near a starting guess x we must solve the linearized equation

$$(1 - M)(x' - x) = -(x - f(x)) = -F(x) \quad (12.11)$$

where $f(x)$ corresponds to integrating from one intersection of the Poincaré surface to another and M is integrated accordingly. Here we run into problems with the direction along the flow, since - as shown in sect. 5.2.1 - this corresponds to a unit eigenvector of M . The matrix $(1 - M)$ does therefore not have full rank. A related problem is that the solution x' of (12.11) is not guaranteed to be in the Poincaré surface of section. The two problems are solved simultaneously by adding a small vector along the flow plus an extra equation demanding that x be in the Poincaré surface. Let us for the sake of simplicity assume that the Poincaré surface is a (hyper)-plane, i.e., it is given by the linear equation

$$(x - x_0) \cdot a = 0, \quad (12.12)$$

where a is a vector normal to the Poincaré section and x_0 is any point in the Poincaré section. (12.11) then becomes

$$\begin{pmatrix} 1 - M & v(x) \\ a & 0 \end{pmatrix} \begin{pmatrix} x' - x \\ \delta T \end{pmatrix} = \begin{pmatrix} -F(x) \\ 0 \end{pmatrix}. \quad (12.13)$$

The last row in this equation ensures that x will be in the surface of section, and the addition of $v(x)\delta T$, a small vector along the direction of the flow, ensures that such an x can be found at least if x is sufficiently close to a solution, i.e., to a fixed point of f .

To illustrate this little trick let us take a particularly simple example; consider a 3-d flow with the $(x, y, 0)$ -plane as Poincaré section. Let all trajectories cross the Poincaré section perpendicularly, i.e., with $v = (0, 0, v_z)$, which means that the marginally stable direction is also perpendicular to the Poincaré section. Furthermore, let the unstable direction be parallel to the x -axis and the stable direction be parallel to the y -axis. In this case the Newton setup looks as follows

$$\begin{pmatrix} 1 - \Lambda & 0 & 0 & 0 \\ 0 & 1 - \Lambda_s & 0 & 0 \\ 0 & 0 & 0 & v_z \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta\tau \end{pmatrix} = \begin{pmatrix} -F_x \\ -F_y \\ -F_z \\ 0 \end{pmatrix}. \quad (12.14)$$

If you consider only the upper-left $[3 \times 3]$ matrix (which is what we would have without the extra constraints that we have introduced) then this matrix is clearly not invertible and the equation does not have a unique solution. However, the full $[4 \times 4]$ matrix is invertible, as $\det(\cdot) = v_z \det(1 - M_\perp)$, where M_\perp is the monodromy matrix for a surface of section transverse to the orbit, see sect. 5.3.

For periodic orbits (12.13) generalizes in the same way as (12.7), but with n additional equations – one for each point on the Poincaré surface. The Newton

setup looks like this

$$\begin{pmatrix} 1 & & & & -J_n \\ -J_1 & 1 & & & \\ & \dots & 1 & & \\ & & \dots & 1 & \\ a & & & -J_{n-1} & 1 \\ & & & & a \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \\ \delta t_1 \\ \vdots \\ \delta t_n \end{pmatrix} = \begin{pmatrix} -F_1 \\ -F_2 \\ \vdots \\ -F_n \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Solving this equation resembles the corresponding task for maps. However, in the process we will need to invert an $[(d+1)n \times (d+1)n]$ matrix rather than a $[d \times d]$ matrix. The task changes with the length of the cycle.

This method can be extended to take care of the same kind of problems if other eigenvalues of the fundamental matrix equal 1. This happens if the flow has an invariant of motion, the most obvious example being energy conservation in Hamiltonian systems. In this case we add an extra equation for x to be on the energy shell plus an extra variable corresponding to adding a small vector along the gradient of the Hamiltonian. We then have to solve

$$\begin{pmatrix} 1 - M & v(x) & \nabla H(x) \\ a & 0 & 0 \end{pmatrix} \begin{pmatrix} x' - x \\ \delta\tau \\ \delta E \end{pmatrix} = \begin{pmatrix} -(x - f(x)) \\ 0 \end{pmatrix} \quad (12.15)$$

simultaneously with

$$H(x') - H(x) = 0. \quad (12.16)$$

The last equation is nonlinear. It is often best to treat this equation separately and solve it in each Newton step. This might mean putting in an additional Newton routine to solve the single step of (12.15) and (12.16) together. One might be tempted to linearize (12.16) and put it into (12.15) to do the two different Newton routines simultaneously, but this will not guarantee a solution on the energy shell. In fact, it may not even be possible to find any solution of the combined linearized equations, if the initial guess is not very good.

12.4.2 How good is my orbit?

Provided we understand the topology of the flow, multi-shooting methods and their variational cousins of chapter 27 enable us to compute periodic orbits of arbitrary length. A notion that errors somehow grow exponentially with the cycle length at Lyapunov exponent rate cannot be right. So how do we characterize the accuracy of an orbit of arbitrary length?

The numerical round-off errors along a trajectory are uncorrelated and act as noise, so the errors $(x(t + \Delta t) - f^{\Delta t}(x(t)))^2$ are expected to accumulate as the

sum of squares of uncorrelated steps, linearly with time. Hence the accumulated numerical noise along an orbit sliced by N intermediate sections separated by $\Delta t_k = t_{k+1} - t_k \sim T_p/N$ can be characterized by an effective diffusion constant

$$D_p = \frac{1}{2(d_e + 1)} \sum_{k=1}^N \frac{1}{\Delta t_k} (x_{k+1} - f^{\Delta t_k}(x_k))^2. \quad (12.17)$$

For hyperbolic flows errors are exponentially amplified along unstable and contracted along stable eigen-directions, so $d_e + 1$ stands for the number of unstable directions of the flow together with the single marginal direction along the flow. An honest calculation requires an honest error estimate. If you are computing a large set of periodic orbits p , list D_p along with T_p and other properties of cycles.

Résumé

There is no general computational algorithm that is guaranteed to find all solutions (up to a given period T_{\max}) to the periodic orbit condition

$$f^{t+T}(x) = f^t(x), \quad T > 0$$

for a general flow or mapping. Due to the exponential divergence of nearby trajectories in chaotic dynamical systems, direct solution of the periodic orbit condition can be numerically very unstable.

A prerequisite for a systematic and complete cycle search is a good (but hard to come by) understanding of the topology of the flow. Usually one starts by - possibly analytic - determination of the equilibria of the flow. Their locations, stabilities, stability eigenvectors and invariant manifolds offer skeletal information about the topology of the flow. Next step is numerical long-time evolution of “typical” trajectories of the dynamical system under investigation. Such numerical experiments build up the “natural measure,” and reveal regions most frequently visited. The periodic orbit searches can then be initialized by taking nearly recurring orbit segments and deforming them into a closed orbits. With a sufficiently good initial guess the Newton-Raphson formula [section 14.4.1]

$$\begin{pmatrix} 1 - M & v(x) \\ a & 0 \end{pmatrix} \begin{pmatrix} \delta x \\ \delta T \end{pmatrix} = \begin{pmatrix} f(x) - x \\ 0 \end{pmatrix}$$

yields improved estimate $x' = x + \delta x$, $T' = T + \delta T$. Iteration then yields the period T and the location of a periodic point x_p in the Poincaré surface $(x_p - x_0) \cdot a = 0$, where a is a vector normal to the Poincaré section at x_0 .

The problem one faces with high-dimensional flows is that their topology is hard to visualize, and that even with a decent starting guess for a point on a periodic orbit, methods like the Newton-Raphson method are likely to fail.

Methods that start with initial guesses for a number of points along the cycle, such as the multipoint shooting method of sect. 12.3, are more robust. The relaxation (or variational) methods take this strategy to its logical extreme, and start by a guess of not a few points along a periodic orbit, but a guess of the entire orbit. As these methods are intimately related to variational principles and path integrals, we postpone their introduction to chapter 27. [chapter 27]

Commentary

Remark 12.1 Close recurrence searches. For low-dimensional maps of flows (for high-dimensional flows, forget about it) picking initial guesses for periodic orbits from close recurrences of a long ergodic trajectory seems like an obvious idea. Nevertheless, ref. [1] is frequently cited. Such methods have been deployed by many, among them G. Tanner, L. Rondoni, G. Morris, C.P. Dettmann, and R.L. Davidchack [2, 13, 14, 10] (see also sect. 18.5). Sometimes one can determine most of the admissible itineraries and their weights without working too hard, but method comes with no guarantee.

Remark 12.2 Piecewise linear maps. The Lozi map (3.20) is linear, and 100,000's of cycles can be easily computed by $[2 \times 2]$ matrix multiplication and inversion.

Remark 12.3 Newton gone wild. Skowronek and Gora [21] offer an interesting discussion of Newton iterations gone wild while searching for roots of polynomials as simple as $x^2 + 1 = 0$.

Exercises

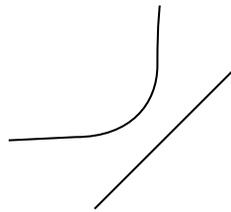
- 12.1. **Cycles of the Ulam map.** Test your cycle-searching routines by computing a bunch of short cycles and their stabilities for the Ulam map

$$f(x) = 4x(1 - x). \quad (12.18)$$

- 12.2. **Cycles stabilities for the Ulam map, exact.** In exercise 12.1 you should have observed that the numerical results for the cycle stability eigenvalues (4.50) are exceptionally simple: the stability eigenvalue of the $x_0 = 0$ fixed point is 4, while the eigenvalue of any other n -cycle is $\pm 2^n$. Prove this. (Hint: the Ulam map can be conjugated to the tent map (10.6). This problem is perhaps too hard, but give it a try - the answer is in many introductory books on nonlinear dynamics.)

- 12.3. **Stability of billiard cycles.** Compute stabilities of few simple cycles.

- (a) A simple scattering billiard is the two-disk billiard. It consists of a disk of radius one centered at the origin and another disk of unit radius located at $L+2$. Find all periodic orbits for this system and compute their stabilities. (You might have done this already in exercise 1.2; at least now you will be able to see where you went wrong when you knew nothing about cycles and their extraction.)
- (b) Find all periodic orbits and stabilities for a billiard ball bouncing between the diagonal $y = x$ and one of the hyperbola branches $y = -1/x$.



- 12.4. **Cycle stability.** Add to the pinball simulator of exercise 8.1 a routine that evaluates the expanding eigenvalue for a given cycle.
- 12.5. **Pinball cycles.** Determine the stability and length of all fundamental domain prime cycles of the binary symbol string lengths up to 5 (or longer) for $R : a = 6$ 3-disk pinball.
- 12.6. **Newton-Raphson method.** Implement the Newton-Raphson method in 2- d and apply it to determination of pinball cycles.

- 12.7. **Rössler flow cycles.** (continuation of exercise 4.4) Determine all cycles up to 5 Poincaré sections returns for the Rössler flow (2.17), as well as their stabilities.

(Hint: implement (12.13), the multipoint shooting methods for flows; you can cross-check your shortest cycles against the ones listed in the table.)

Table: The Rössler flow (2.17): The itinerary p , a periodic point $x_p = (0, y_p, z_p)$ and the expanding eigenvalue Λ_p for all cycles up to the topological length 7. (J. Mathiesen, G. Simon, A. Basu)

n_p	p	y_p	z_p	Λ_e
1	1	6.091768	1.299732	-2.403953
2	01	3.915804	3.692833	-3.512007
3	001	2.278281	7.416481	-2.341923
	011	2.932877	5.670806	5.344908
4	0111	3.466759	4.506218	-16.69674
5	01011	4.162799	3.303903	-23.19958
	01111	3.278914	4.890452	36.88633
6	001011	2.122094	7.886173	-6.857665
	010111	4.059211	3.462266	61.64909
	011111	3.361494	4.718206	-92.08255
7	0101011	3.842769	3.815494	77.76110
	0110111	3.025957	5.451444	-95.18388
	0101111	4.102256	3.395644	-142.2380
	0111111	3.327986	4.787463	218.0284

- 12.8. **Cycle stability, helium.** Add to the helium integrator of exercise 2.10 a routine that evaluates the expanding eigenvalue for a given cycle.

- 12.9. **Collinear helium cycles.** Determine the stability and length of all fundamental domain prime cycles up to symbol sequence length 5 or longer for collinear helium of figure 7.2.

- 12.10. **Uniqueness of unstable cycles***.** Prove that there exists only one 3-disk prime cycle for a given finite admissible prime cycle symbol string. Hints: look at the Poincaré section mappings; can you show that there is exponential contraction to a unique periodic point with a given itinerary? Exercise 27.1 might be helpful in this effort.

- 12.11. **Inverse iteration method for a Hamiltonian repeller.**

Table: All periodic orbits up to 6 bounces for the Hamiltonian Hénon mapping (12.19) with $a = 6$. Listed are the cycle itinerary, its expanding eigenvalue Λ_p , and its “center of mass.” The “center of mass” is listed because it turns out the “center of mass” is often a simple rational or a quadratic irrational.

p	Λ_p	$\sum x_{p,i}$
0	0.715168×10^1	-0.607625
1	-0.295285×10^1	0.274292
10	-0.989898×10^1	0.333333
100	-0.131907×10^3	-0.206011
110	0.558970×10^2	0.539345
1000	-0.104430×10^4	-0.816497
1100	0.577998×10^4	0.000000
1110	-0.103688×10^3	0.816497
10000	-0.760653×10^4	-1.426032
11000	0.444552×10^4	-0.606654
10100	0.770202×10^3	0.151375
11100	-0.710688×10^3	0.248463
11010	-0.589499×10^3	0.870695
11110	0.390994×10^3	1.095485
100000	-0.545745×10^5	-2.034134
110000	0.322221×10^5	-1.215250
101000	0.513762×10^4	-0.450662
111000	-0.478461×10^4	-0.366025
110100	-0.639400×10^4	0.333333
101100	-0.639400×10^4	0.333333
111100	0.390194×10^4	0.548583
111010	0.109491×10^4	1.151463
111110	-0.104338×10^4	1.366025

Consider the Hénon map (3.18) for area-preserving

(“Hamiltonian”) parameter value $b = -1$. The coordinates of a periodic orbit of length n_p satisfy the equation

$$x_{p,i+1} + x_{p,i-1} = 1 - ax_{p,i}^2, \quad i = 1, \dots, n_p, \quad (12.19)$$

with the periodic boundary condition $x_{p,0} = x_{p,n_p}$. Verify that the itineraries and the stabilities of the short periodic orbits for the Hénon repeller (12.19) at $a = 6$ are as listed above.

Hint: you can use any cycle-searching routine you wish, but for the complete repeller case (all binary sequences are realized), the cycles can be evaluated simply by inverse iteration, using the inverse of (12.19)

$$x''_{p,i} = S_{p,i} \sqrt{\frac{1 - x'_{p,i+1} - x'_{p,i-1}}{a}}, \quad i = 1, \dots, n_p.$$

Here $S_{p,i}$ are the signs of the corresponding cycle point coordinates, $S_{p,i} = x_{p,i}/|x_{p,i}|$. (G. Vattay)

12.12. **“Center of mass” puzzle**.** Why is the “center of mass,” tabulated in exercise 12.11,

References

- [12.1] D. Auerbach, P. Cvitanović, J.-P. Eckmann, G.H. Gunaratne and I. Procaccia, *Phys. Rev. Lett.* **58**, 2387 (1987).
- [12.2] M. Baranger and K.T.R. Davies *Ann. Physics* **177**, 330 (1987).
- [12.3] B.D. Mestel and I. Percival, *Physica D* **24**, 172 (1987); Q. Chen, J.D. Meiss and I. Percival, *Physica D* **29**, 143 (1987).
- [12.4] find Helleman et all Fourier series methods
- [12.5] J.M. Greene, *J. Math. Phys.* **20**, 1183 (1979)
- [12.6] H.E. Nusse and J. Yorke, “A procedure for finding numerical trajectories on chaotic saddles” *Physica D* **36**, 137 (1989).
- [12.7] D.P. Lathrop and E.J. Kostelich, “Characterization of an experimental strange attractor by periodic orbits”
- [12.8] T. E. Huston, K.T.R. Davies and M. Baranger *Chaos* **2**, 215 (1991).
- [12.9] M. Brack, R. K. Bhaduri, J. Law and M. V. N. Murthy, *Phys. Rev. Lett.* **70**, 568 (1993).
- [12.10] J.J. Crofts and R.L. Davidchack, “Efficient detection of periodic orbits in chaotic systems by stabilising transformations,” [arXiv:nlin.CD/0502013](https://arxiv.org/abs/nlin.CD/0502013).