Chapter 16

Fixed points, and how to get them

Cycles. Is there anything they can’t do?
— Mason Porter, channeling Homer Simpson

Having set up the dynamical context, we now turn to the key and unavoidable numerical task in this subject; we must search for the solutions \((x, T)\), \(x \in \mathbb{R}^d, T \in \mathbb{R}^+\) satisfying the periodic orbit condition

\[
\begin{align*}
f^T(x) &= x, \quad T > 0, \quad \text{(flow)} \\
f^n(x) &= x, \quad n \geq 1, \quad \text{(map)}
\end{align*}
\]

for a given flow or map.

In chapters 21 and 22 we will establish that spectra of evolution operators can be extracted from periodic orbit sums:

\[
\sum \text{(spectral eigenvalues)} = \sum \text{(periodic orbits)}.
\]

Hence, periodic orbits are the necessary ingredient for evaluation of the spectra of evolution operators. We need to know what periodic orbits can exist, and the symbolic dynamics developed so far is an invaluable tool toward this end.

This chapter, a continuation of chapter 7, is intended as a hands-on guide to extracting periodic orbits, and should be skipped on first reading - you can return to it whenever the need for finding actual cycles arises. A serious cyclist will want chapter 33 to also learn about the variational methods to find cycles, chapter 33. They are particularly useful when little is known about the topology of a flow, such as in high-dimensional periodic orbit searches.
Due to the exponential divergence of nearby trajectories in chaotic dynamical systems, fixed point searches based on direct solutions of the fixed-point condition (16.1) 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. 16.2, and the variational methods of chapter 33, are considerably more robust and safer.

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 preparing a list of all distinct admissible prime periodic symbol sequences, such as the list given in table 18.1. The relations between the temporal symbol sequences and the spatial layout of the topologically distinct regions of the state space discussed in chapters 14 and 15 should enable us to guess the location of a series of periodic points along a cycle. Armed with such an informed guess we proceed to improve it by methods such as Newton-Raphson iteration; we show how this works by applying Newton method to 1- and d-dimensional maps. But first, where are the cycles?

16.1 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 minimize the length of an approximate orbit that visits a desired sequence of boundary bounces.

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. Some might be satisfied with any rampaging robot that finds “the most important” cycles. The ergodic explorations of recurrences sometimes perform admirably well, and we discuss this next.

16.1.1 Cycles from long time series

Two wrongs don’t make a right, but three lefts do.

—Appliance guru
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. Nevertheless, one does observe close passes to the least unstable equilibria and periodic orbits, as in figure 16.1. Ergodic exploration by long-time trajectories (or long-lived transients, in case of strange repellers) can uncover state space regions with near 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.

The search consists of following a long trajectory in state space, and looking for close returns of the trajectory to itself, see figure 16.1. Whenever the trajectory almost closes in a loop (within a given tolerance), a point close to 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 16.1, 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 14 and 15, but are a bit of work to implement.

16.1.2 Cycles found by thinking

Thinking is extra price.
—Dicho Colombiano

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 (2.7) 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 grid points across $M$ in each dimension, and compute the velocity field $v_k = v(x_k)$ at each grid point $x_k$; 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 visitations to equilibrium neighborhoods. 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 15.9 to maps from a section of the unstable manifold of equilibrium $s_n$ to the section of stable 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.

We showed in the Lorenz flow example 14.5 how to reduce the 3-dimensional Lorenz flow to a 1-dimensional return map. In the Rössler flow example 2.3 we sketched the attractor by running a long chaotic trajectory, and noting that the attractor is very thin, but that otherwise the return maps that we plotted were disquieting – figure 3.3 did not appear to be a 1-to-1 map. In the next example we show how to use such information to locate cycles approximately. In the remainder of this chapter and in chapter 33 we shall learn how to turn such guesses into highly accurate cycles.

### 16.2 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. The function $f^n$ oscillates wildly, with as many as $2^n$ or more closely spaced fixed points, and finding a specific periodic point, such as 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 $n$. Furthermore, the Jacobian of $f^n$ can be ill-conditioned because its matrix elements can grow like $\Lambda^n$, where $\Lambda$ is the leading multiplier of a single discrete time step Jacobian. A better alternative is the multipoint or multiple shooting method, with the Jacobian matrix broken up into a product of single-step Jacobian matrices, each with eigenvalues $\approx \Lambda$.

While it might very hard to give a precise initial 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. That is why in chapter 14 we have developed a qualitative theory of how these cycle points are laid out topologically.

For a 1-dimensional map a cycle of length $n$ is a zero of the $n$-dimensional vector function $F$:

$$F(x) = F\left(\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array}\right) = \left(\begin{array}{c} x_1 - f(x_n) \\ x_2 - f(x_1) \\ \vdots \\ x_n - f(x_{n-1}) \end{array}\right).$$

The relationship between the temporal symbol sequences and the spatial layout of the topologically distinct regions of state space discussed in chapter 14 enable us to guess the 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 Newton’s method now takes the form

$$\frac{d}{dx}F(x)(x' - x) = -F(x), \quad (16.2)$$

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

$$\frac{d}{dx}F(x) = \left(\begin{array}{cccc} 1 & -f'(x_1) & \cdots & \cdots & -f'(x_n) \\ -f'(x_1) & 1 & \cdots & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1 & -f'(x_{n-1}) \\ -f'(x_{n-1}) & \cdots & \cdots & 1 & 1 \end{array}\right). \quad (16.3)$$

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.

When one sets up Newton iteration on a 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 and 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 iteration the vector containing the right hand side is the correction to the initial guess.
16.2.1  \(d\)-dimensional maps

Armed with clever initial guesses from a system’s symbolic dynamics, we can easily extend the Newton-Raphson iteration method to \(d\)-dimensional maps. In this case \(f'(x_i)\) is a \([d \times d]\) matrix, and \(\frac{\partial F(x)}{\partial x}\) is an \([nd \times nd]\) matrix. In each of the steps 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 inverting the \(n\)th element of the diagonal we are inverting a \([d \times d]\) matrix \((1 - \prod f'(x_i))\) which can be done as long as none of the eigenvalues of \(\prod f'(x_i)\) equals 1, i.e., if the cycle has no marginally stable eigen-directions.

16.3 Cost function

(R. Paškauskas and P. Cvitanović)

It pays to think in terms of a cost (or error) function \(I(\Delta x) = (x + \Delta x - f(x + \Delta x))^2/2\). Periodic orbit condition (16.1) corresponds both to a zero of \(I(\Delta x)\), and of its first \(\Delta x\) variation. Expand \(I(\Delta x)\) to the second order in \(\Delta x\), \(\tilde{I} \approx \frac{1}{2} \Delta x \cdot \tilde{\Delta} x + (x - f(x)) \cdot \tilde{\Delta} x + \frac{1}{2} (x - f(x))^2/2\), where \(\tilde{\Delta} x = (1 - J(x))\Delta x\). To find an extremum, we set the derivative with respect to \(\tilde{\Delta} x\) to zero. As the term \((x - f(x))^2/2\) is a constant under \(\Delta x\) variation, let us define an unconstrained cost function

\[
I_0(\tilde{\Delta} x) = \frac{1}{2} \tilde{\Delta} x \cdot \tilde{\Delta} x + (x - f(x)) \cdot \tilde{\Delta} x ,
\]

(16.4)

Setting the derivative of this function

\[
\frac{\partial I_0(\tilde{\Delta} x)}{\partial \Delta x} = \tilde{\Delta} x + x - f(x) = (1 - J(x)) \cdot \Delta x + x - f(x)
\]

(16.5)

to zero recovers the Newton setup (7.3)

Next, we need to enforce the constraint that curbs the directions in which \(\Delta x\) can point. Lagrange multipliers come to help.

A local surface of section can be constructed when \(f(x)\) is “near” the initial point \(x\). A natural choice is a hyperplane perpendicular to the velocity vector \(v(x)\). The reference point \(x_0\) in (7.10) is \(x\) itself, and the surface of section condition is \(U(x + \Delta x) = v(x) \cdot \Delta x = 0\). Introduce a Lagrange multiplier \(\lambda\), and assemble a cost function with the constraint:

\[
I_1(\tilde{\Delta} x, \lambda) = \frac{1}{2} \tilde{\Delta} x \cdot \tilde{\Delta} x + [x - f(x)] \cdot \tilde{\Delta} x + \lambda v(x) \cdot \tilde{\Delta} x .
\]

(16.6)

Now we differentiate \(I_1(\Delta x, \lambda)\) with respect to each argument and set the derivatives to zero. We recover the Newton setup (7.7), with the Lagrange multiplier
\[ \lambda = \Delta t \] interpreted as the time increment needed to place \( f(x) \) onto the section, \( f(x) \to f(x) + v(f(x))\Delta t \).

A global surface of section is a fixed surface \( U(x + \Delta x) - U(x_0) \approx \partial U(x)\Delta x + U(x) - U(x_0) \) that hopefully transects all or a significant portion of recurrent parts of the flow. It is not as 'natural' as the local section (7.6), but hard to avoid in practice, and one is interested not only in the fixed point itself, but in the global reach of its unstable manifold as well. The simplest choice is a hyperplane (7.10).

The cost function and the variational equations are then

\[
I_2(\Delta x, \Delta t) = \frac{1}{2} \Delta x[1 - J(x)]\Delta x + (x - f(x))\Delta x + \Delta t (\partial U(x)\Delta x + U(x) - U(x_0)), \tag{16.7}
\]

\[
\begin{bmatrix}
1 - J(x) \\
\partial U(x)
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta t
\end{bmatrix} = -
\begin{bmatrix}
x - f(x) \\
U(x) - U(x_0)
\end{bmatrix}. \tag{16.8}
\]

Further continuous symmetries can be handled in the same fashion. Suppose, for example, that we are searching for periodic orbits of a Hamiltonian flow. There, periodic orbits not only have the time-translation symmetry, but energy-translation symmetry as well. What is energy-translation symmetry? If there exists a periodic orbit at \( x \) with energy \( H(x) = E \), and period \( T \), it is very likely that it belongs to a family of orbits \((x + \epsilon \Delta x(E), T + \epsilon \Delta t(E))\) continuous under variation of \( E \). As with the time-translation symmetry, this implies a unit Floquet multiplier: indeed, we know from sect. 8.4 that symplectic eigenvalues come in pairs, so unit multiplier in the time direction implies a unit multiplier in its dual, the energy direction, \((\Lambda_t, \Lambda_E, \cdots) = (1, 1, \cdots)\). But extending the number of constraints is no longer a problem: add more Lagrange multipliers. Consider the following system

\[
I_3(\Delta x, \lambda_1, \lambda_2) = \Delta x[1 - J(x)]\Delta x/2 + (x - f(x))\Delta x + \lambda_1 (U(x + \Delta x) - U(x_0)) + \lambda_2 (H(x + \Delta x) - E_0) \tag{16.9}
\]

\[
\begin{bmatrix}
1 - J(x) \\
\partial U(x) \\
\partial H(x)
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\lambda_1 \\
\lambda_2
\end{bmatrix} = -
\begin{bmatrix}
x - f(x) \\
U(x) - U(x_0) \\
H(x) - E_0
\end{bmatrix}. \tag{16.10}
\]

This is the Newton iteration setup for how to search for a periodic orbit of a Hamiltonian flow with a global surface of section \( U(x) = U(x_0) \) and fixed energy \( E_0 \). Note that these instructions do not put every iteration on a surface \( U(x) = U(x_0) \) and energy \( H(x) = E_0 \), unless the surface is a plane \( U(x) = a \cdot (x - x_0) \), but instead assure that the iterations (provided they converge) will approach the surface super-exponentially.

For periodic orbits multi-point shooting generalizes in the same way as (16.3), but with \( n \) additional equations – one for each point on a Poincaré section. The
Newton setup looks like this:

\[
\begin{bmatrix}
  1 & -J_n \\
  -J_1 & 1 & & \cdots \\
  & \cdots & \cdots \\
  a & & & & -J_{n-1} & 1 \\
  \vdots & \ddots & \ddots & \ddots \\
  a & & & & & 1
\end{bmatrix}
\begin{bmatrix}
  v_1 \\
  \vdots \\
  v_n
\end{bmatrix}
= \begin{bmatrix}
  \Delta x_1 \\
  \Delta x_2 \\
  \vdots \\
  \Delta x_n \\
  \Delta t_1 \\
  \vdots \\
  \Delta t_n
\end{bmatrix}
= \begin{bmatrix}
  -F_1 \\
  -F_2 \\
  \vdots \\
  -F_n \\
  0 \\
  \vdots \\
  0
\end{bmatrix}
\]

(16.11)

Solving this equation resembles the corresponding task for maps. However, we will need to invert a \([(d + 1)n \times (d + 1)n]\) matrix rather than a \([d \times d]\) matrix.

Résumé

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. The 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 which regions are most frequently visited. Periodic orbit searches can then be initialized by taking nearly recurring orbit segments and deforming them into closed orbits. With a sufficiently good initial guess, Newton-Raphson iteration then yields the period \(T\) and the location of a periodic point \(x_p\).

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. 16.2, are more robust. 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. Just as these methods are intimately related to variational principles and path integrals, we postpone their introduction until chapter 33.

Commentary

Remark 16.1 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. [A1.31] is frequently cited. Such methods have been deployed by many, among them...

Remark 16.2 Cycles, searches, and symmetries. A few comments about the role of symmetries in actual extraction of cycles. In the $N$-disk billiard example, a fundamental domain is a sliver of the $N$-disk configuration space delineated by a pair of adjoining symmetry axes. The flow may further be reduced to a return map on a Poincaré surface of section. While in principle any Poincaré surface of section will do, a natural choice in the present context are crossings of symmetry axes, see example 8.7. In actual numerical integrations only the last crossing of a symmetry line needs to be determined. The cycle is run in global coordinates and the group elements associated with the crossings of symmetry lines are recorded; integration is terminated when the orbit closes in the fundamental domain. Periodic orbits with non-trivial symmetry subgroups are particularly easy to find since their points lie on crossings of symmetry lines, see example 8.7.

Remark 16.3 Symmetries of the symbol square. For a discussion of symmetry lines see refs. [16.5, 14.10, 16.6, 8.7, 8.8]. It is an open question (see remark 25.2) as to how time reversal symmetry can be exploited for reduction of cycle expansions of chapter 23. For example, the fundamental domain symbolic dynamics for reflection symmetric systems is discussed in some detail in sect. 25.5, but how does one recode from time-reversal symmetric symbol sequences to desymmetrized 1/2 state space symbols?

16.4 Examples

Example 16.1 Periodic orbits of billiards. Consider how this works for 3-disk pinball game of sect. 15.5. Label the three disks by 1, 2 and 3, and associate to an itinerary an itinerary, a sequence of labels indicating the order in which the disks are visited, as in figure 15.14. 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. 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 $s = (s_1, \ldots, s_n)$, a task that we postpone to sect. 33.3. As an example, the short periods and stabilities of 3-disk cycles computed this way are listed table 33.3, and some examples are plotted in figure 15.14. It’s a no brainer, and millions of such cycles have been computed.

Example 16.2 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 periodic 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 \( f'(x_2)f'(x_1)f'(x_3) \) represents 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

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

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, and the right hand side is an explicit formula for the corrections to our initial guess. With this, we have gone through one Newton iteration.

Example 16.3 Newton method for time delay maps. Some \( d \)-dimensional maps (such as the Hénon map (3.17)) can be written as 1-dimensional time delay maps of the form

\[
f(x_i) = f(x_{i-1}, x_{i-2}, \ldots, x_{i-d}).
\]

In this case, \( \frac{\partial}{\partial x} F(x) \) 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.
Exercises

16.1. Ulam map periodic points. (continued from exercise 14.8)

(a) compute the five periodic points of cycle 10011 for the Ulam map (14.21) \( f(x) = 4x(1-x) \) using your Newton or other routine.

(b) compute the five periodic points of cycle 10000

(c) plot the above two cycles on the graph of the Ulam map, verify that their topological ordering is as in the 'canonical' full tent map exercise 14.8.

(d) (optional) This works only for the Ulam map: compute periodic points by conjugating the full tent map periodic points of exercise 14.8 using exercise A2.4.

16.2. Cycles stability for the Ulam map (exact). In exercise 16.1 you should have observed that the numerical results for the cycle Floquet multipliers (4.43) are exceptionally simple: the Floquet multiplier 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 (14.20). This problem is perhaps too hard, but give it a try - the answer is in many introductory books on nonlinear dynamics.)

16.3. Newton-Raphson method. Implement the Newton-Raphson method in 2-dimensional and apply it to the determination of pinball cycles.

16.4. Cycle stability. Add to the pinball simulator of exercise 9.1 a routine that evaluates the expanding eigenvalue for a given cycle.

16.5. 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.

16.6. Fundamental domain fixed points. Use the formula (9.11) for billiard Jacobian matrix to compute the periods \( T_p \) and the expanding eigenvalues \( \Lambda_p \) of the fundamental domain \( \bar{0} \) (the 2-cycle of the complete 3-disk space) and \( \bar{T} \) (the 3-cycle of the complete 3-disk space) fixed points:

\[

\begin{array}{l|cc}
   & T_p & \Lambda_p \\
\hline
   \bar{0}: & R - 2 & R - 1 + R \sqrt{1 - 2/R} \quad (16.13) \\
   \bar{T}: & R - \sqrt{3} & -2R/\sqrt{3} + 1 - 2R/\sqrt{3}/R \\
\end{array}
\]

We have set the disk radius to \( a = 1 \).

16.7. Fundamental domain 2-cycle. Verify that for the \( 10 \)-cycle the cycle length and the trace of the Jacobian matrix are given by

\[

\begin{align*}
   L_{10} &= 2\sqrt{3} - \sqrt{3}/R + 1 - \sqrt{3}/R \\
   \text{tr} J_{10} &= \Lambda_{10} + 1/\Lambda_{10} \\
   &= 2L_{10} + 2 + \frac{1}{2} L_{10}(L_{10} + 2)^2
\end{align*} \quad (16.14)
\]

The \( 10 \)-cycle is drawn in figure 15.12. The unstable eigenvalue \( \Lambda_{10} \) follows from (8.30).


16.9. Rössler flow cycles. (continuation of exercise 7.1) Determine all cycles for the Rössler flow (2.27), as well as their stabilities, up to 5 Poincaré section returns (Hint: implement (16.3), the multipoint shooting methods for flows; you can cross-check your shortest cycles against the ones listed in the table.)

Table: The Rössler flow (2.27): The itinerary \( p \), a periodic point \( x_p = (0, y_p, z_p) \) and the expanding eigenvalue \( \Lambda_p \) for all cycles up to topological length 7.

\[
\begin{array}{cccccc}
   n_p & p & y_p & z_p & \Lambda_p \\
\hline
   1 & 01 & 6.091768 & 1.299732 & -2.403953 \\
   2 & 01 & 3.158504 & 3.692833 & -3.512007 \\
   3 & 001 & 2.278281 & 7.416481 & -2.341923 \\
   4 & 10 & 2.938277 & 5.670836 & -5.344908 \\
   5 & 0111 & 3.466759 & 4.506218 & -16.69746 \\
   6 & 00110 & 1.622904 & 7.866173 & -6.857665 \\
   7 & 001011 & 4.059211 & 3.46266 & -61.64909 \\
\end{array}
\]

16.10. Colinear 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 8.2.

16.11. 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 maps; can you show that there is ex-
ponential contraction to a unique periodic point with a
given itinerary? Exercise 33.1 might be helpful in this
effort.


(a) We have formulated three Newton setups for
flows: the ‘local’ setup (7.7), the ‘hyperplane’
set up (7.11), and the ‘global’ setup (16.8). Derive
(16.8) and verify that if the surface of section is
a hyperplane, it reduces to (7.11). (Hint: it is not
inconceivable that (7.11) is wrong as it stands.)

(b) (optional) Derive (16.10), the Newton setup for
Hamiltonian flows.

References

[16.1] D. Auerbach, P. Cvitanović, J.-P. Eckmann, G.H. Gunaratne and I. Procaci-


[16.4] find Helleman et all Fourier series methods


1, 45 (1990).


70, 568 (1993).


mations for detecting unstable periodic orbits in high-dimensional flows,”

Vrahatis, “Locating periodic orbits by Topological Degree theory;”

013108 (2005).


