Ionization Times of Classical Hydrogen in Parallel Electric and Magnetic Fields:

Scaling Mechanisms in the Heteroclinic Tangle

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science with Honors in
Physics from the College of William and Mary in Virginia,

by Brian P. Tighe

Accepted for _____
(Honors, High Honors, or Highest Honors)

_____
Advisor: John Delos, Physics, W&M

_____
Gene Tracy, Physics, W&M

_____
Gina Hoatson, Physics, W&M

_____
Don Campbell, Economics, W&M

Williamsburg, Virginia
May 2000

Abstract

High energy classical orbits of the Hydrogen atom in parallel electric and magnetic fields are chaotic under certain field strengths. Ionization times of orbits originating in chaotic regions of phase space are very sensitive to initial conditions, and structure-within-structure is observed, suggesting a fractal pattern. Chaotic classical orbits are associated with the system's heteroclinic tangle, and the structure of the tangle has been used to describe the ionization process. Escape regions in the tangle have been identified and their back-iterates shown to impose an exponential grid on the surface of section in the vicinity of a stable manifold. This grid, elsewhere discussed as a fractal tiling of phase space [1], is the physical source of a generic exponential scaling found in the ionization times. Non-generic features and their source have also been identified, but may not yet be predicted. We give evidence that a graph of the ionization time versus initial condition may be a fractal curve. We call this an epistrophic fractal, characterized by multiple base structures which scale as a single parameter. This is in contrast to a multifractal, which possesses a single base structure and multiple scaling parameters.

Contents

1 Introduction

This thesis is a theoretical investigation of the classical dynamics of Hydrogen in parallel electric and magnetic fields. It is a continuation of work contained in the undergraduate honors thesis of Abigail Flower. Specifically, our work has sought to further understand the ionization times of this system. These times have sensitive dependence on initial conditions, especially near discontinuities in the vicinity of periodic orbits. Evidence of structure-within-structure indicates that the dependence is a multifractal. In this work, we identify structures which repeat themselves on all scales. Also, we identify structures which are nonrepeating, but which constitute initial steps of subsequent repeating structures. [3].

2 Background

2.1 The system

It is reasonable to describe an electron of large principal quantum number classically. [4] In such a description of the Hydrogen atom the electron moves in an orbit around the proton, just as the planets around the sun. We consider such an atom (having been excited, for example, by a laser pulse) in the presence of parallel electric and magnetic fields. These electric and magnetic forces, together with the Coulomb attraction of the nucleus, cause the equations of motion to be nonlinear. In addition, the parallel nature of the fields allows the system to be studied in two dimensions.

The Hamiltonian of the system is [5]

$$H = \frac{(\vec{p} - q\vec{A})^2}{2m_e} + qV_{total}$$

$$V_{total} = V_{Coulomb} + V_{electric}$$

$$q = -e$$

It is most convenient to study the system in rotational parabolic coordinates. With respect to polar coordinates this two-dimensional coordinate system may be written

$$u = \sqrt{r}\sqrt{1 + \cos\theta}$$
$$v = \sqrt{r}\sqrt{1 - \cos\theta}$$

We take angular momentum about the z-axis to be zero and transform to a rotating coordinate frame to eliminate the paramagnetic term [2]. Letting E represent the energy of the orbit, F the magnitude of the electric field, and B the magnitude of the magnetic field, we may scale the electric field strength to 1 in the following way (scaled variables are hatted) so as to reduce the number of parameters to two.

$$p = \hat{p}$$
$$u = \hat{u}F^{-\frac{1}{4}}$$
$$v = \hat{v}F^{-\frac{1}{4}}$$
$$E = \hat{E}F^{\frac{1}{2}}$$
$$B = \hat{B}F^{\frac{3}{4}}$$

In its final form the Hamiltonian is

$$\hat{H} = \frac{\hat{p}_u^2 + \hat{p}_v^2}{2} - \hat{E}(\hat{u}^2 - \hat{v}^2) + \hat{B}^2\frac{\hat{u}^4\hat{v}^2 + \hat{u}^2\hat{v}^4}{8} + \frac{\hat{u}^4 - \hat{v}^4}{2} - 2$$

Note that the term involving B^2 (the diamagnetic term) couples $\hat{u}$ and $\hat{v}$ and introduces nonlinearity. In its absence the system is a separable 2-dimensional anharmonic oscillator.

Coordinate space contours of the effective potential energy are shown in Figure 1. Considering the proton to be fixed at the origin, we see that if an electron is to escape, it must do so over one of the saddles on the $\hat{v}$ axis.

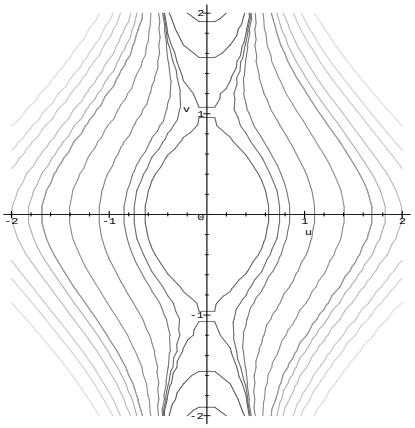

Figure 1: Coordinate space contours of the effective potential of Hydrogen in parallel electric and magnetic fields

2.2 Chaotic motion in phase space

Hamiltonian systems are deterministic, meaning that once the initial state of the system is specified, all future states are thereby determined. That is, the motion of the system may be predicted for all times to come, and there is no ambiguity in that

prediction.

Many nonlinear systems, including the one under study, display chaotic motion. To understand this, we need several definitions. A dense orbit comes arbitrarily close to any point in the domain of the mapping. A transitive system is one which, given any two points, possesses an orbit which approaches arbitrarily close to those two points. A sensitive system is any system such that, given some arbitrary distance, two orbits will eventually evolve so as to be further apart than that distance. Formally, then, Devaney defines chaotic systems as follows [10]:

1. Periodic orbits are dense
2. The dynamical system is transitive
3. The system is sensitive

To fully specify the state of a two-dimensional Hamiltonian system, one must know not only its $\hat{u}$ and $\hat{v}$ coordinates but its velocity along both of those axes. The time evolution of the system occurs in this four-dimensional space, called phase space. From its starting point (the system's initial conditions) the system traces a curve in phase space, called a trajectory. Because the system is deterministic, no two trajectories may cross. Should they cross, there would be ambiguity about the future evolution of the crossing point; there would be two possible paths away from that single point.

To simplify visualization of phase space we make use of Poincaré's surface of section, whereby the system is reduced to two dimensions by fixing the energy and the value of one coordinate. This reduction in dimension allows us to choose a plane in space defined by one coordinate and its conjugate momentum; each time a trajectory

crosses the plane, we mark the point of crossing. Thus, a set of four differential equations has been reduced to a two-dimensional difference equation or mapping function. Each choice of energy and fixed coordinate specifies a different plane and mapping function [6]. We have fixed the $\hat{p}_u$ coordinate and have chosen $\hat{E} = -1.0$ and $\hat{B} = 3.26$.

Trajectories which remain in some neighborhood of phase space cross the Poincaré plane repeatedly in that region. Periodic trajectories in phase space intersect the surface of section at fixed points; these fixed points play an important role in organizing the motion of the system. Trajectories having equal energy but different initial conditions describe one-dimensional manifolds on the surface of section; one manifold is a smooth curve composed of the discrete crossings of many trajectories. Stable fixed points (O points) are enclosed by elliptical manifolds; unstable fixed points are at the intersection of two manifolds (called separatrices): the inset and the outset. Trajectories move towards and away from an unstable fixed point (X point) along the inset and outset, respectively. The inset and outset of an X point are also called the stable and unstable manifolds of that point. On each new iterate, a trajectory on an inset crosses the surface of section closer to the fixed point; it only reaches the fixed point, however, in the limit as time goes to infinity[6].

The physical pendulum provides a simple example of a dynamical system in phase space. The system possesses a stable equilibrium (hanging straight down) and an unstable equilibrium (balancing upright). The ellipses about the stable fixed point represent oscillations about the equilibrium position, as in the ticking of a grandfather clock or the swinging of a child on a swing-set. The separatrices which approach the

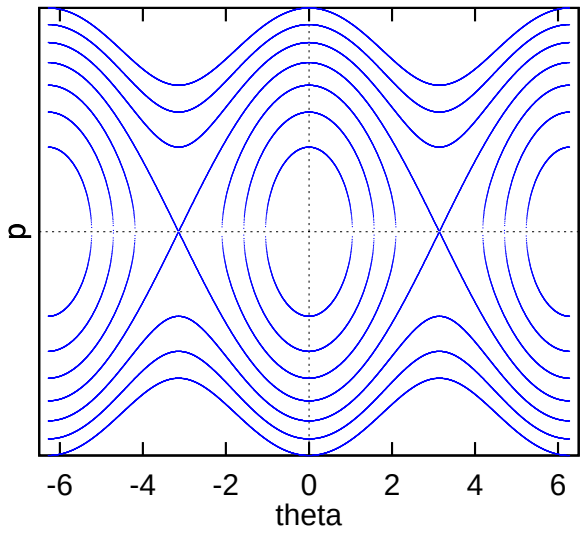

Figure 2: The phase space representation of the simple physical pendulum. There is a stable fixed point, or O point, at $\theta = 0$ and an unstable fixed point, or X point, at $\theta = \pi$. These points are repeated periodically every $\pm n\pi$, $n = 1, 2, 3 \ldots$.

unstable fixed point represent the admittedly unlikely scenario in which the swinging child reaches the upright position and balances there (or rather, reaches the balancing point in the limit of infinite time); conversely, it is also possible to fall away from the upright position. Which scenario is actually the case depends whether the separatrix is an inset or an outset. There is one other class of motion, in which the oscillations of the swinging child are so energetic that they repeatedly make a full circle about the bar from which the swing hangs. Figure 2 shows these motions in phase space. The separatrices are so named because they separate the regions of qualitatively different motion. An unusual feature of the pendulum is that insets and outsets join smoothly.

In contrast to Figure 2, which depicts purely regular motion, stands Figure 3, a surface of section for the parallel fields system generated by Flower. [2] Regular motion about a single X-point and two O-points is apparent, but exterior to this

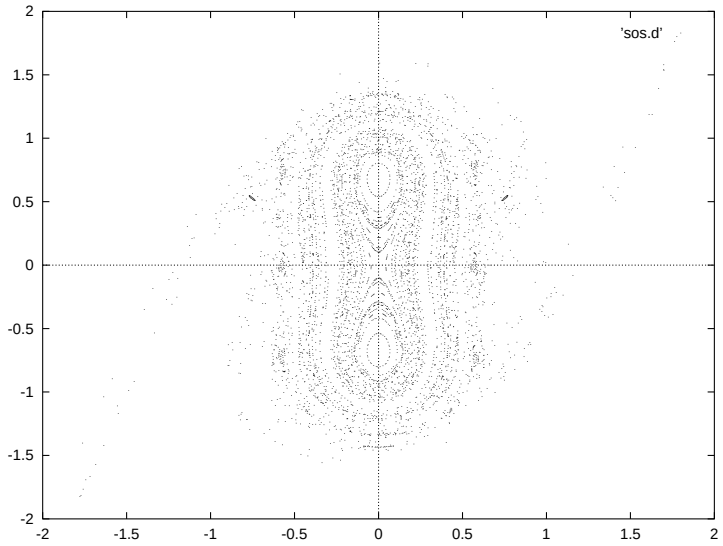

Figure 3: The Hydrogen atom in parallel electric fields system with scaled parameters $\hat{E} = -1$ and $\hat{B} = 3.26$. We observe regular motion (two 0 points and an X point) in a sea of chaos.

region we see what looks like random noise. This is chaotic motion in phase space, as viewed in the Poincaré plane.

A manifestation of chaos in Hamiltonian systems is that an inset and outset do not join smoothly but cross each other instead. Such a crossing is a point shared by both the inset and the outset; because motion in the Poincaré plane is described by a single difference equation, all future iterates of this point will also be shared. As a consequence of this, once the manifolds cross once, they must cross an infinite number of times, each crossing moving successively closer to the inset's fixed point [6].

An interesting feature of Hamiltonian orbits or maps is that they preserve oriented areas in the surface of section[7]. Given an area in the plane, all future iterates of

that area will be of equal area. This is of interest as manifolds cross because between the first and second crossings the inset and outset enclose an area. All future pairs of crossings will involve the enclosure of an equal area. These enclosed areas take the form of "lobes" having one separatrix as their base. If we consider lobes having the inset as their base, it is apparent that with each iterate forward in time, the base of the lobe becomes shorter, forcing the lobe to become narrower. In order to preserve area, then, the lobe must become longer than it was on the previous iterate. Moreover, an inset or outset can never cross itself. These elongated lobes wrap around each other in the Poincaré plane forming incredibly complicated structures. If the intersecting inset and outset belong to the same fixed point, the structure is called a homoclinic tangle; if they belong to different fixed points the structure is a heteroclinic tangle. The motion of trajectories which are on or near tangles is chaotic[6].

2.3 Ionization and the tent map

Fractals are geometric objects which, at all scales below some maximum scale display self-similarity. Just as a square may be decomposed into smaller squares which may in turn be similarly decomposed, a fractal at one size scale may be decomposed into smaller units of its "large-scale" structure. Multifractals, more complex than a simple fractal structure, possess multiple scaling parameters.

Fractal structure may be illustrated by a simplified one-dimensional model of atomic ionization, the tent map[3] [7]. Let the position of the electron be described by the variable $0 \leq x < \infty$, and let the proton be at $x = 0$. Let us say that for

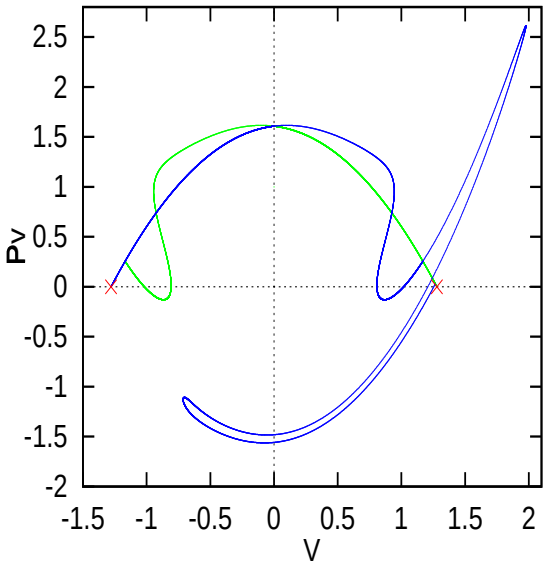

Figure 4: Early development of a heteroclinic tangle after the first crossing of inset and outset. The red X points are unstable periodic orbits. The blue curve is part of the outset from the left-hand periodic orbit. The green curve is part of the inset of the right-hand periodic orbit. Following the evolution of the blue outset further in time reveals increasingly narrow, elongated, and complex lobe structures.

$x \leq 1$, the system is bound; for $x > 1$, the electron has escaped. We will describe the evolution of the electron's position by a discrete mapping, just as the surface of section represents a discrete mapping of continuously varying motion in phase space. Moreover, we will take the dynamics of this mapping to be given by the tent map

$$\Delta = \begin{cases} 3\text{x} & \text{for } 0 \leq x \leq \frac{1}{2} \\ \text{-3(x-1)} & \text{for } \frac{1}{2} < x \leq 1 \end{cases}$$

We see that, for each iteration, the first third and final third of the entire unit interval $0 \leq x \leq 1$ is mapped back onto the interval. The middle third is mapped to values $x > 1$. Once mapped beyond the unit interval, the tent map provides no means for return to that interval; escape is permanent. On each iteration of the tent map, the middle third of each remaining interval is mapped to $x > 1$, and is eliminated.

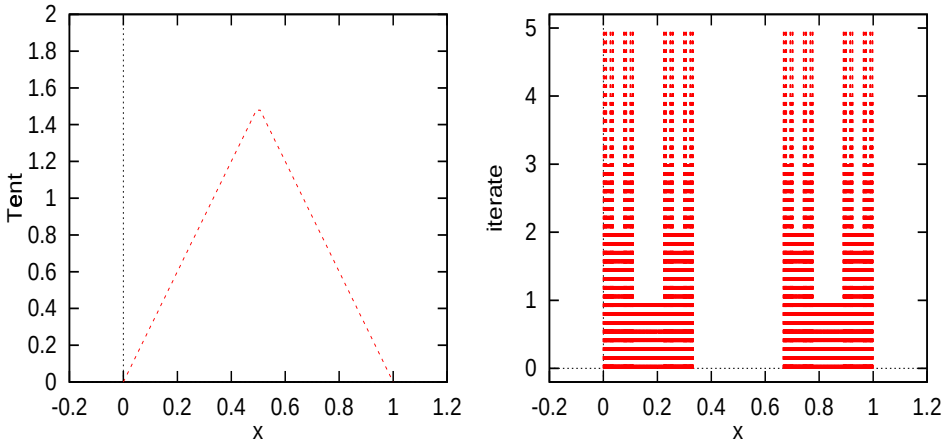

Figure 5: The tent map and the time evolution of the Cantor set, which serves as a simplistic approximation to the ionization process.

After an infinite number of iterations, we are left with a set of points known as the Cantor set. These points are bound orbits of the system. Figure 5 shows how the progressive escape of segments on the initial unit interval (the set of initial conditions) eventually leaves only these bound orbits behind. The structure-within-structure here is evident in the time evolution of the system, a repeated stripping away of the middle third of a line segment.

Maps having less symmetry than the tent map have comparable properties; any map having a single maximum of height greater than 1 on the unit interval will produce a pattern similar to the middle thirds pattern, two maxima generate an even fifths pattern, and so on [9].

3 Motivations

3.1 Escape of high energy orbits

"Escape" means that the electron has left the Coulomb field of the proton, or equivalently, ionization of the Hydrogen atom has occurred. Orbits of the system are calculated by numerical integration of the equations of motion (see Appendix A); this integration places practical constraints on the study. It is possible for the electron to spend a very long time in the vicinity of the proton before escaping, and a bound orbit will spend an infinite amount of time near the proton. Not knowing beforehand whether a particular orbit will escape, we must constrain ourselves to a certain window of time or risk integrating a bound orbit forever. If, within the allotted time, the electron does not escape, we declare the orbit to be bound; if it reaches some escape condition (of our choosing), we declare the orbit to have escaped. In this sense we are studying escape orbits having short ionization times. The chosen condition for escape is the hyperbola

$$\hat{v}^2 - \hat{u}^2 = 2$$

It can be shown that particles reaching this curve with velocity away from the proton continue their motion away from the proton for all time.

For Figure 6, each orbit was started from the origin moving toward the upper-right quadrant of $\hat{u} - \hat{v}$ space; initial momentum in the $\hat{v}$ direction was varied. Such orbits of the system possess qualitative similarities: each makes some number of "loops" around the proton before escaping, unless it is a bound orbit, which loops for all

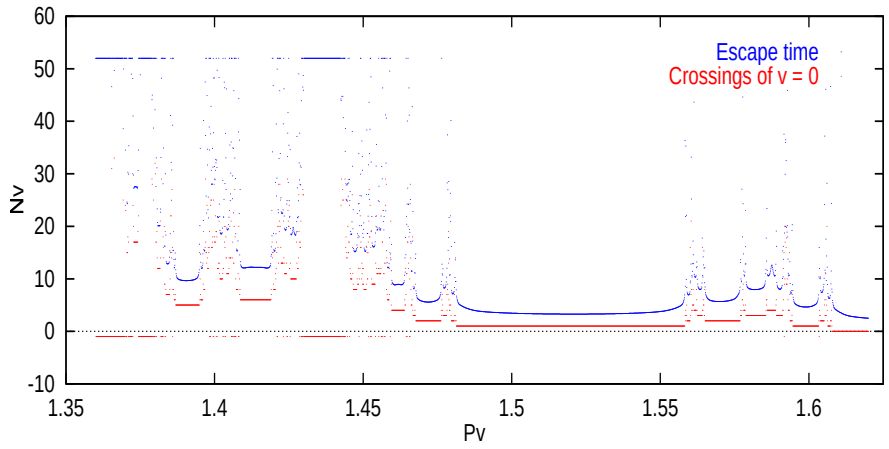

Figure 6: Escape times and crossings of the line $v = 0$ in the region of the heteroclinic tangle

time. Figure 7 shows two such orbits.

For initial $\hat{p}_v < 1.36$, all orbits appear to be bound; for $\hat{p}_v > 1.62$, it appears that all orbits escape immediately. This region in $\hat{p}_v$ corresponds to the heteroclinic tangle identified by Flower. [2] Between these two values, escape time depends sensitively on initial momentum; indeed, the functional relationship possesses multiple discontinuities, as shown in Figure 6.

This relationship may be expressed in a different manner. As stated above, each escape orbit oscillates about the proton some finite number of times before escaping; as it does so, it repeatedly crosses the line $\hat{v} = 0$. We have chosen to examine the number n_v of crossings of the $\hat{v}$-axis before escape, and we record this n_v as a function of initial direction of motion, represented by $\hat{p}_v$. Figure 6 demonstrates the effect of such a change in focus. Clearly this transformation does little violence to the qualitative features of the escape time dependence; moreover the discreteness of the n_v variable allows us to easily identify regions of common qualitative characteristics

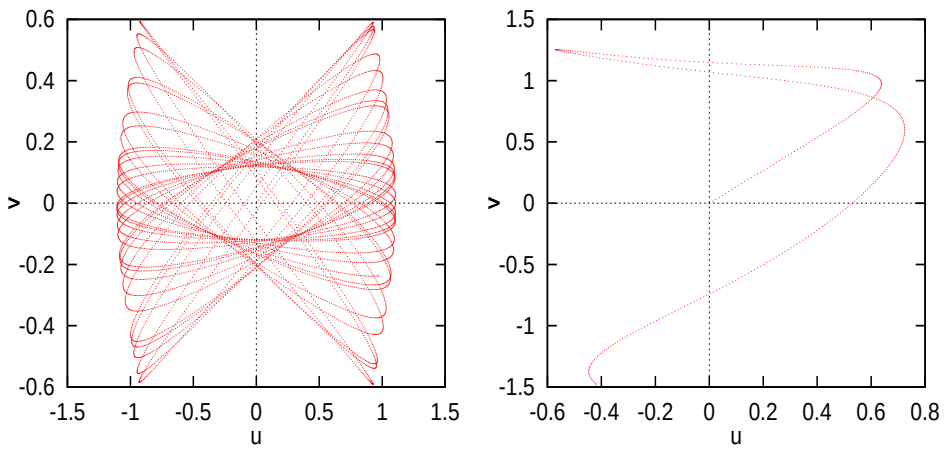

Figure 7: Bounded and unbounded orbits of Hydrogen in parallel electric and magnetic fields. Both orbits were started from the origin, and differ only in their initial momentum along the $\hat{v}$ coordinate, $\hat{p}_v$.

on the line of initial conditions.

3.2 Structure within structure

Our graphs of n_v vs. $\hat{p}_v$ are the proxy for escape time dependence on initial conditions. In these graphs there are several wide swaths of constant n_v separated by cascades of varying n_v in the vicinity of the discontinuities of the ionization time function. We anticipate similarity between both the shape of these cascades and the spacing between them [3][8]. We may look for intra- or cross-generational similarities, a generation referring to some scale in $\hat{p}_v$. These similarities may be most readily seen for cascades of low n_v, corresponding to short escape times. Cascades described by longer orbits have more complex structure, and it is unclear whether that structure is repeated within or between generations. Observation of the two cascades at the border

of the long orbit region ($\hat{p}_v \approx 1.47$) reveals structure similar to that of short orbit cascades. Figure 8 shows similarity between three generations, each a progressively smaller window in $\hat{p}_v$. Note that the third and fourth generations are roughly the flipped image of the second.

Whether there is repeated cascade structure of high n_v value, particularly on the left of the first generation, is not clear. Moreover, spacing between cascades seems to be mapped nonlinearly between generations.

4 The Heteroclinic Tangle

The heteroclinic tangle of the parallel fields system, identified by Flower, provides a framework in which to describe qualitatively the structure of the escape-time function and to quantify some features. Figure 9 shows the heteroclinic tangle in its entirety; Figures 10 - 17 display its early iterates singly.

4.1 Landmarks in the Heteroclinic Tangle

The heteroclinic tangle in its entirety is too complex to make much sense of. It is more helpful to view it iterate by iterate, watching as it unfolds. The first iterates map out several simple structures which are useful reference points; these are elaborated here.

There are two insets and two outsets. From the fixed points to the first intersection of each pair of separatrices is enclosed a region in phase space which shall be called the "complex". Calculations indicate that if a point maps from inside the complex to outside the complex, then it will within a few more iterations reach the escape

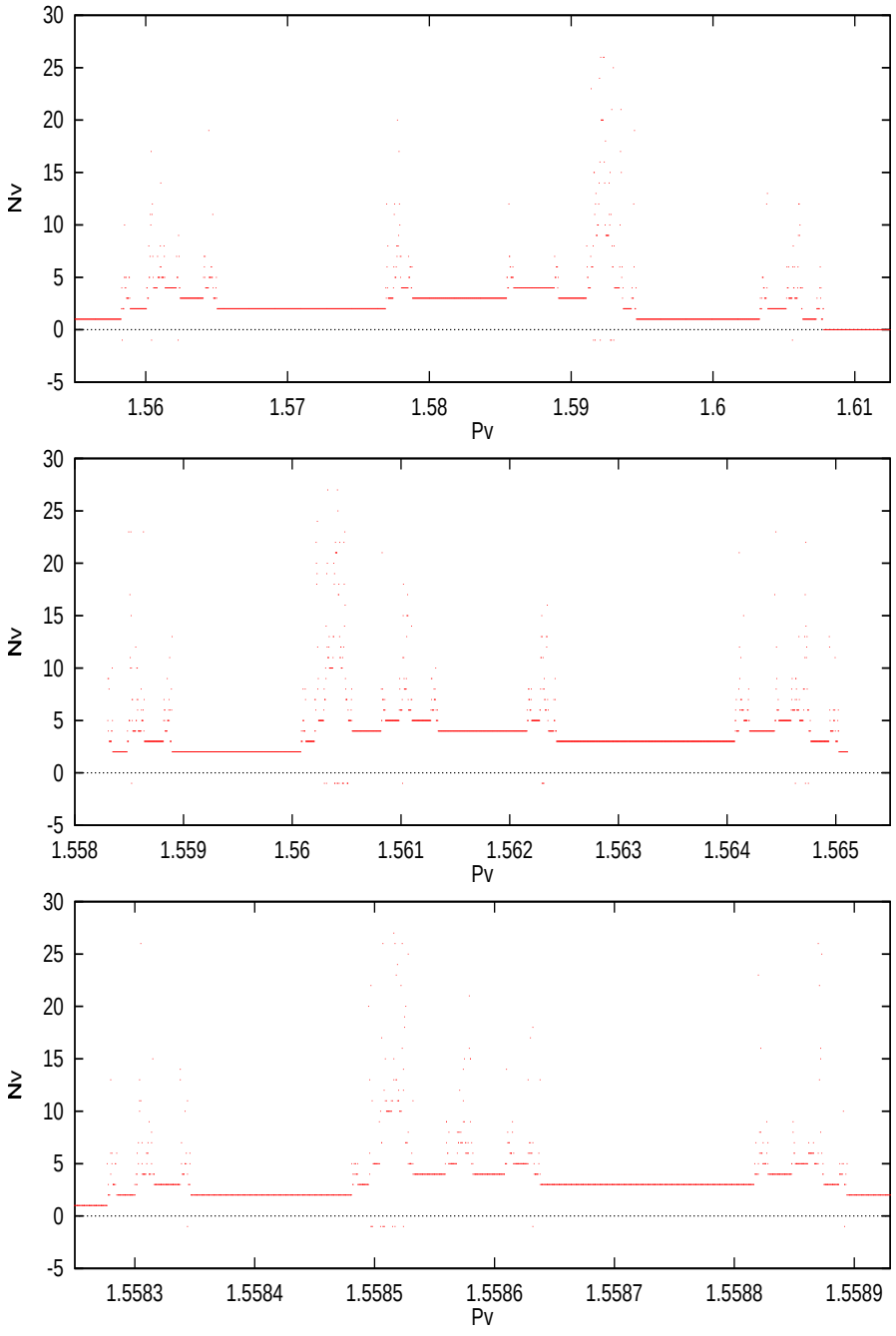

Figure 8: Three generations of nearly self-similar structure. The first generation pictured is approximately the flipped image of the latter two.

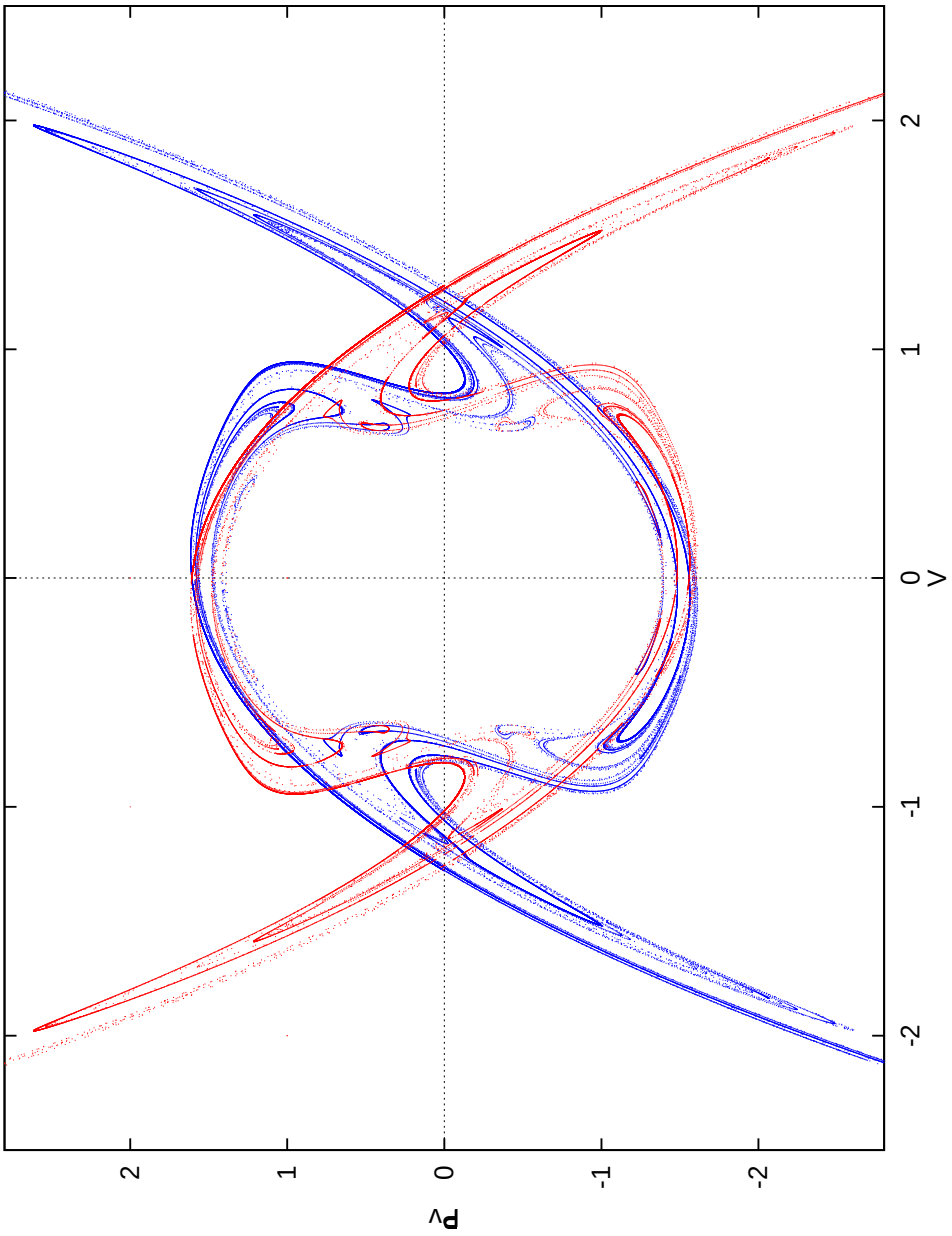

Figure 9: The heteroclinic tangle of the parallel fields system owing to the separatrices in the upper-half plane. In blue is the upper-half plane outset of the left X point; in red is the upper-half plane inset of the right X point. There is a conjugate tangle created by the lower-half plane separatrices; by symmetry its structure is the same.

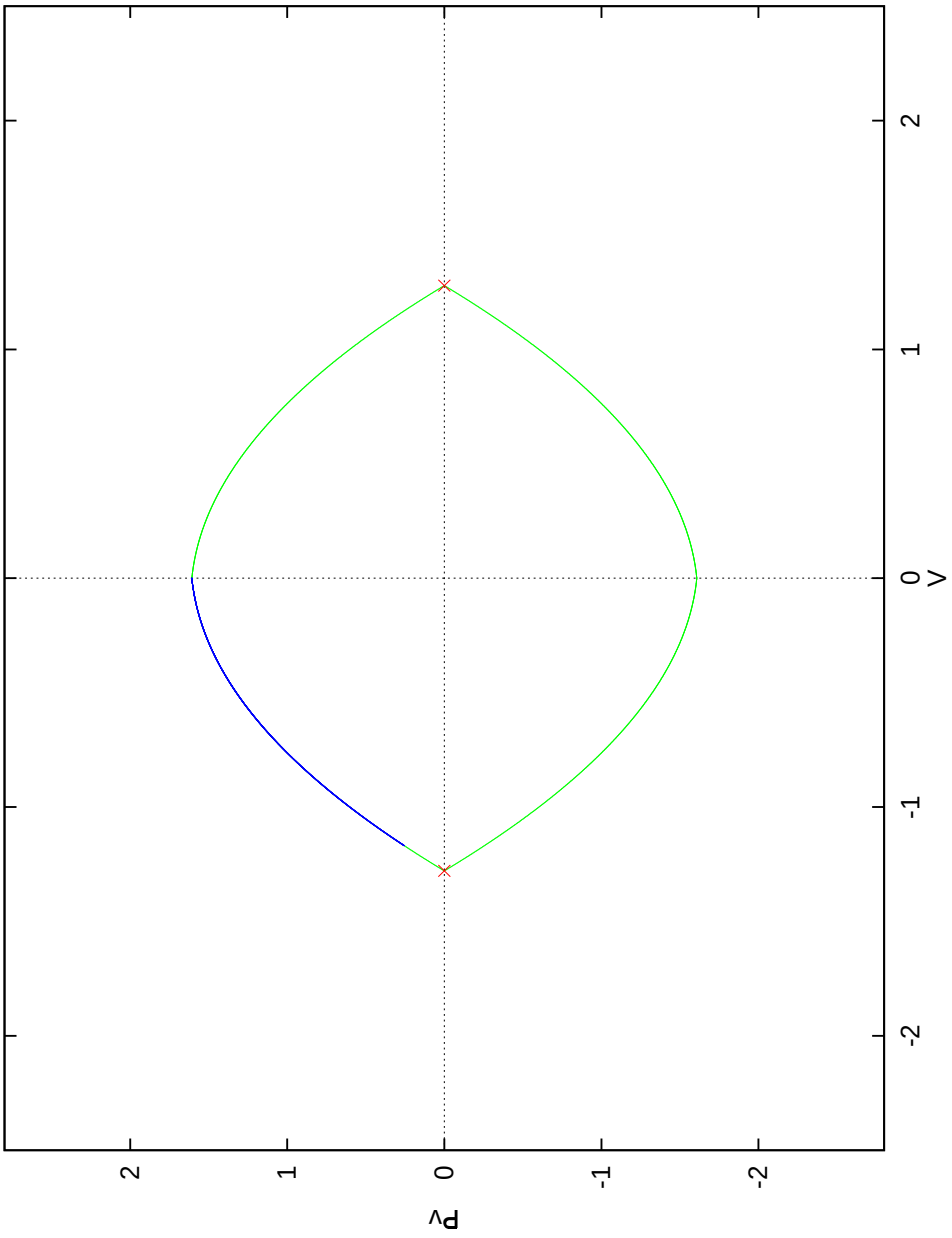

Figure 10: The 0^{th} iterate of ℓ, a curve which is on the outset of the left fixed point of a $\hat{v} - \hat{p}_v$ surface of section of the Hydrogen atom in parallel electric and magnetic fields.

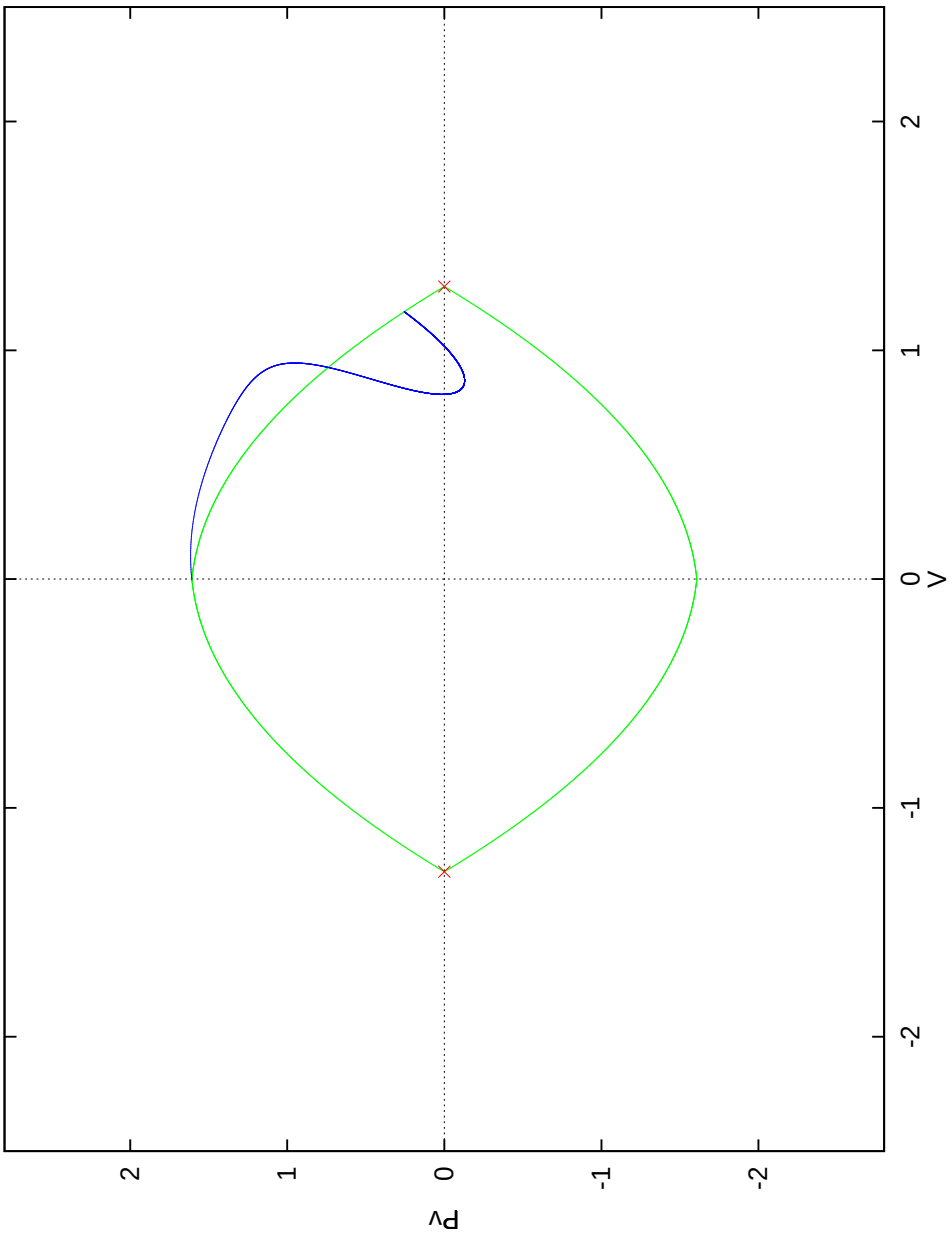

Figure 11: The first iterate of ℓ, which, together with the inset of the right fixed point, encloses the turnstile.

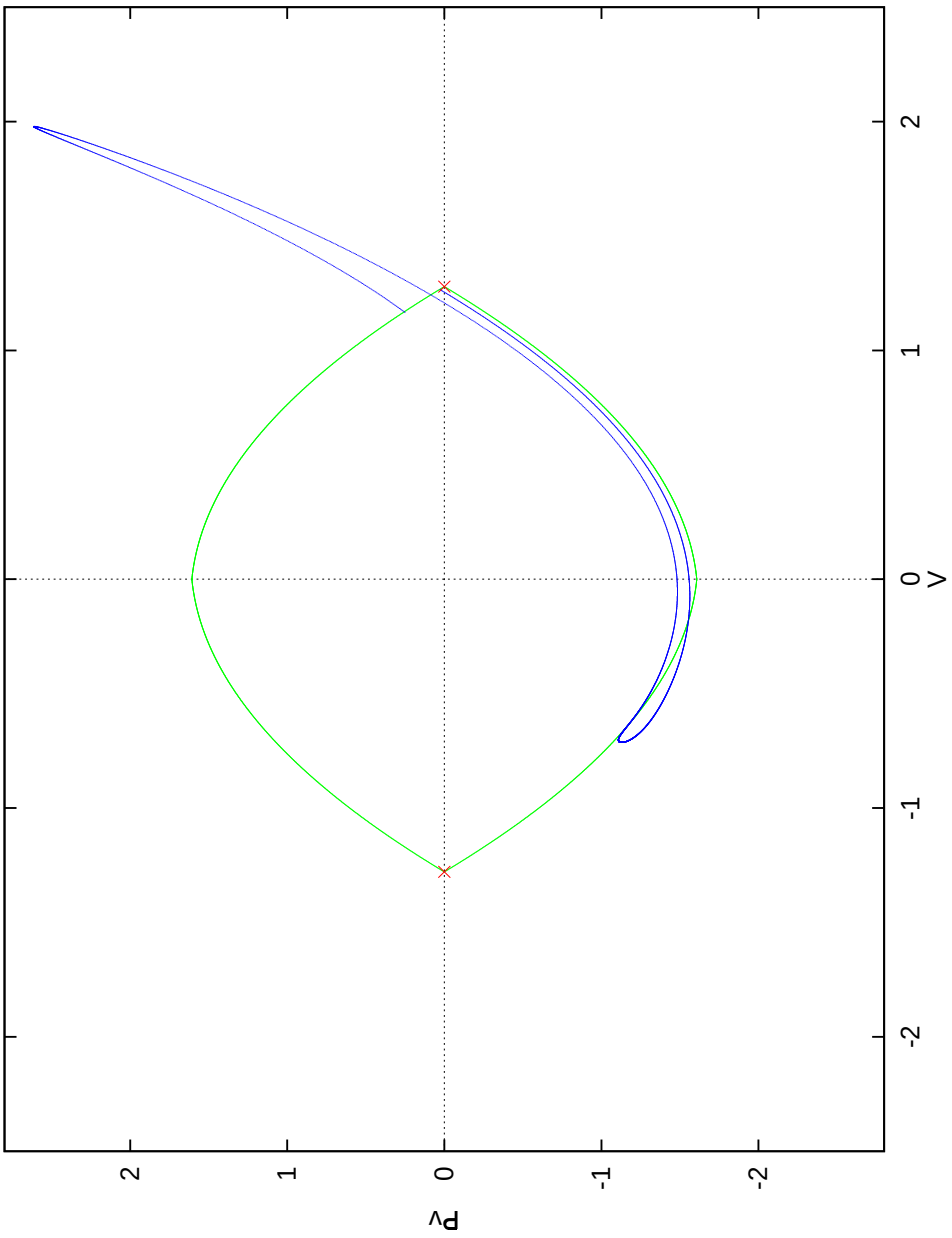

Figure 12: The second iterate of ℓ. First escape through $E(even)$.

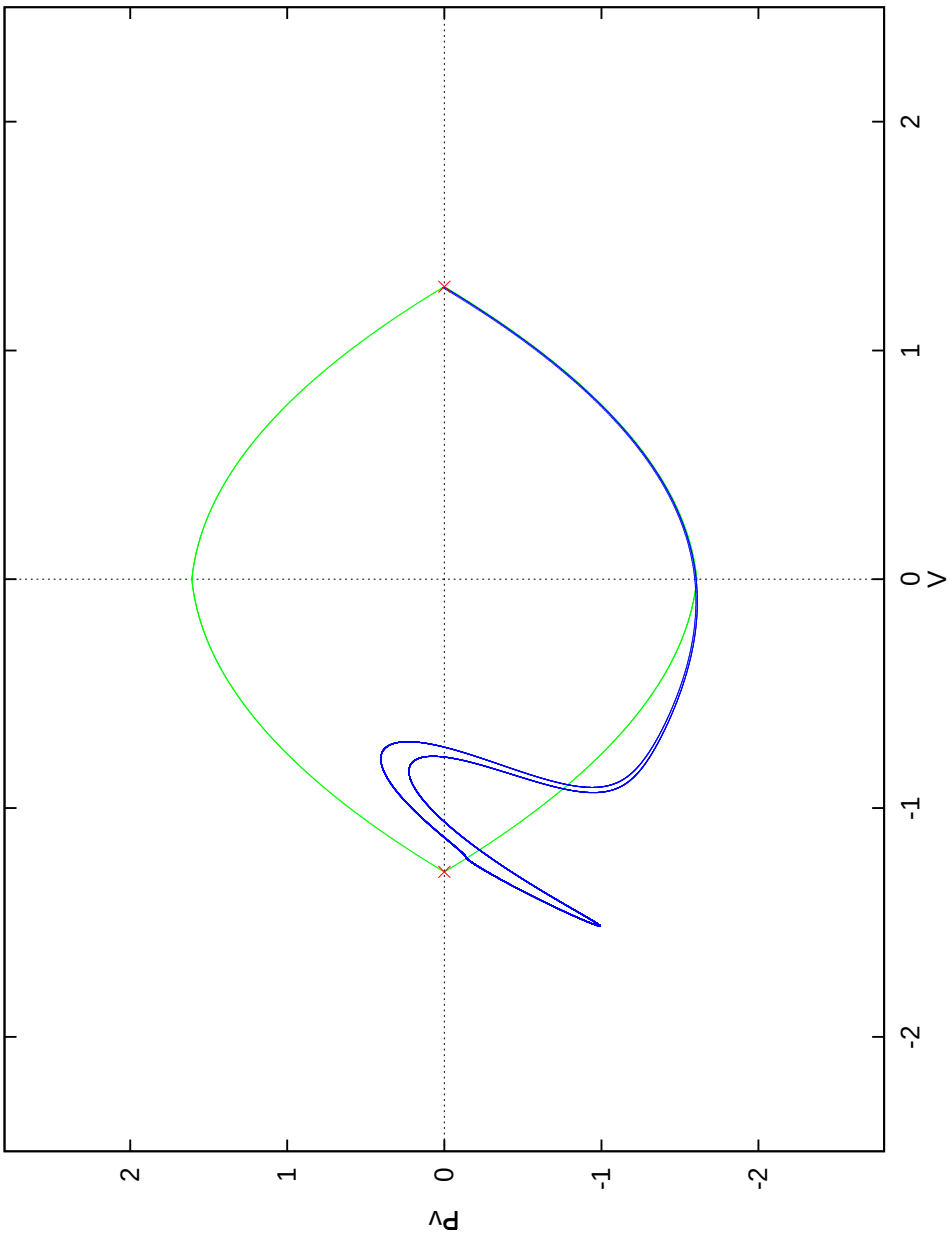

Figure 13: The third iterate of ℓ. The lobe of trajectories which escaped through $E(odd; i = 1)$ is omitted for this and all future iterates.

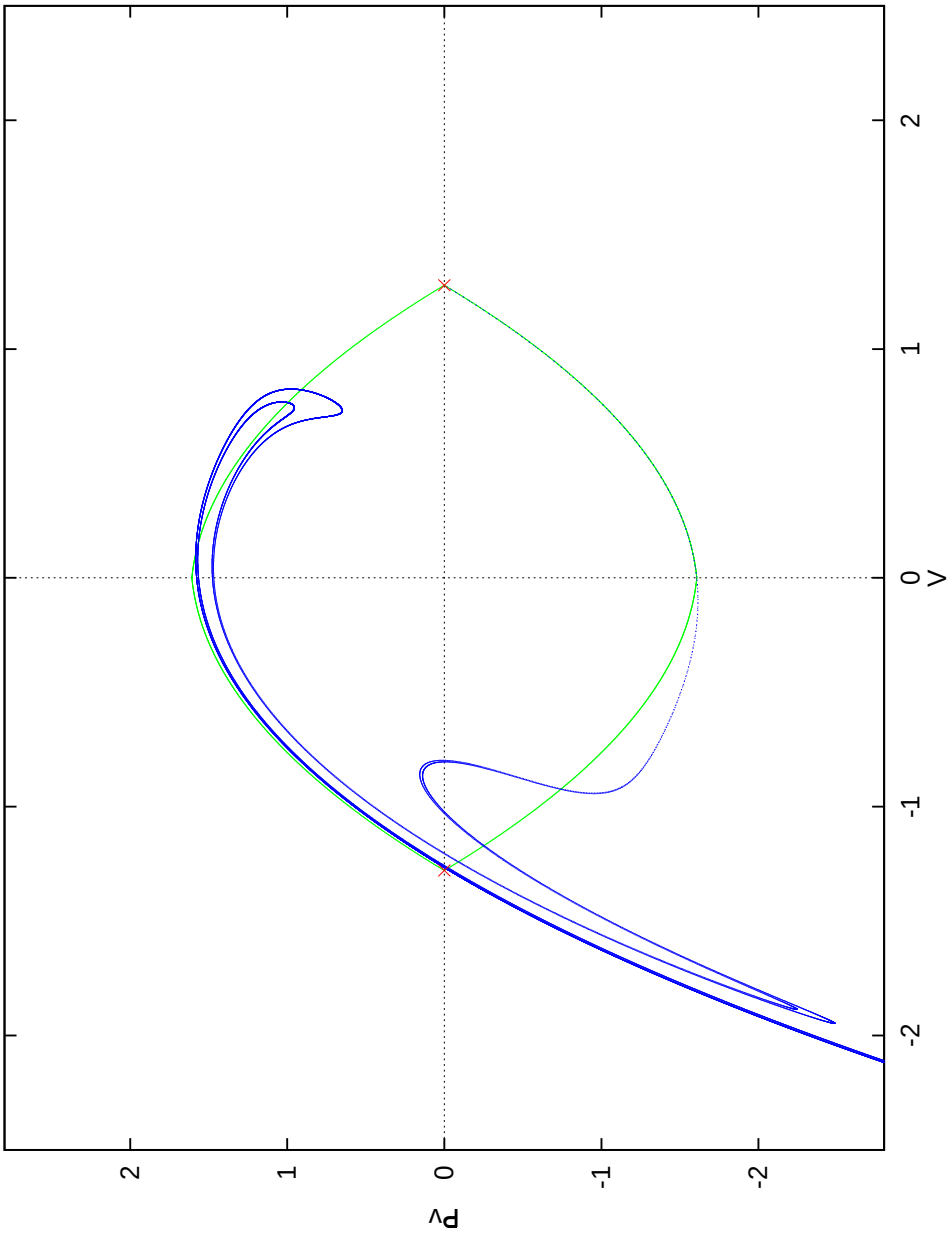

Figure 14: The fourth iterate of ℓ. First escape with $n_v = 3$.

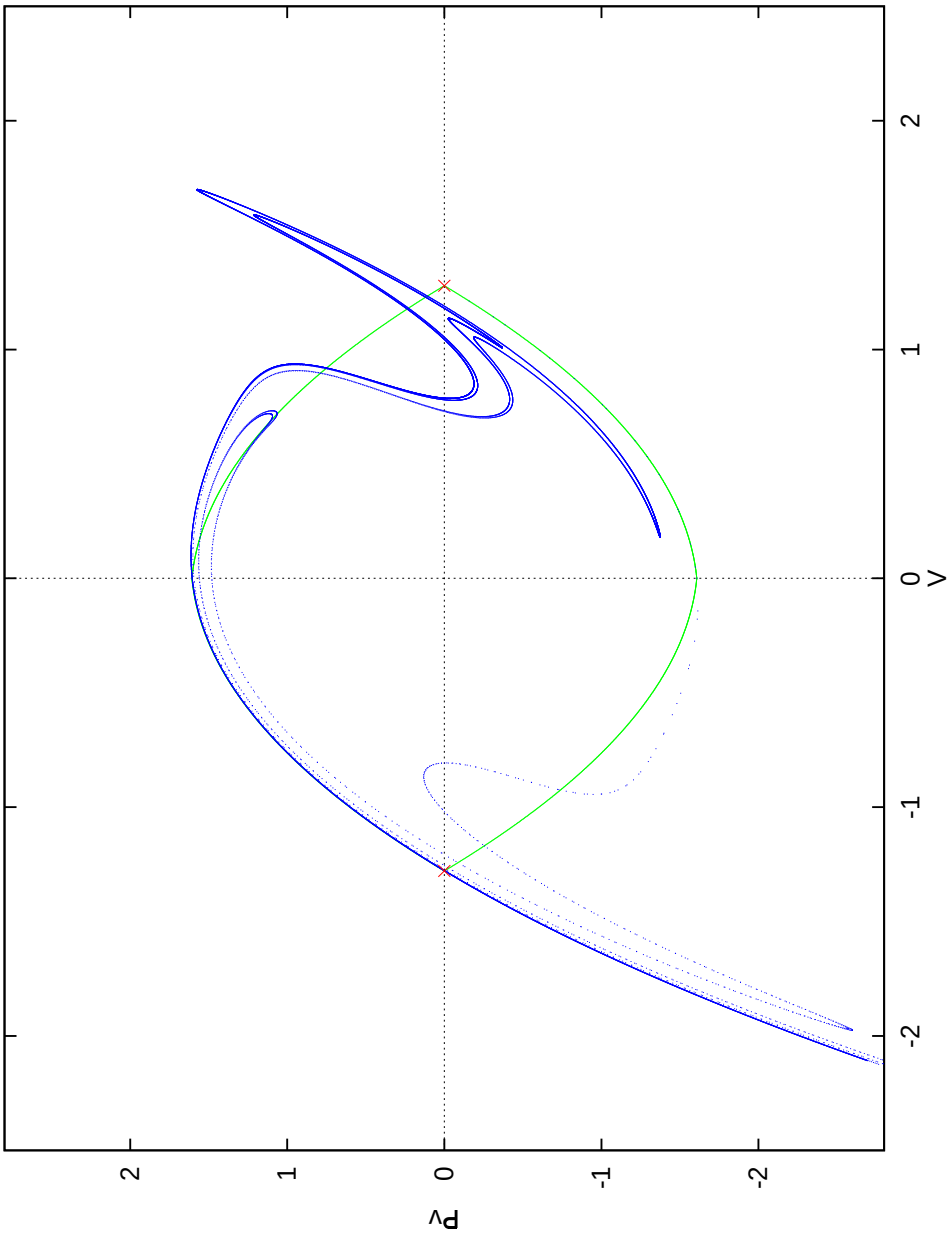

Figure 15: The fifth iterate of ℓ.

iterate05

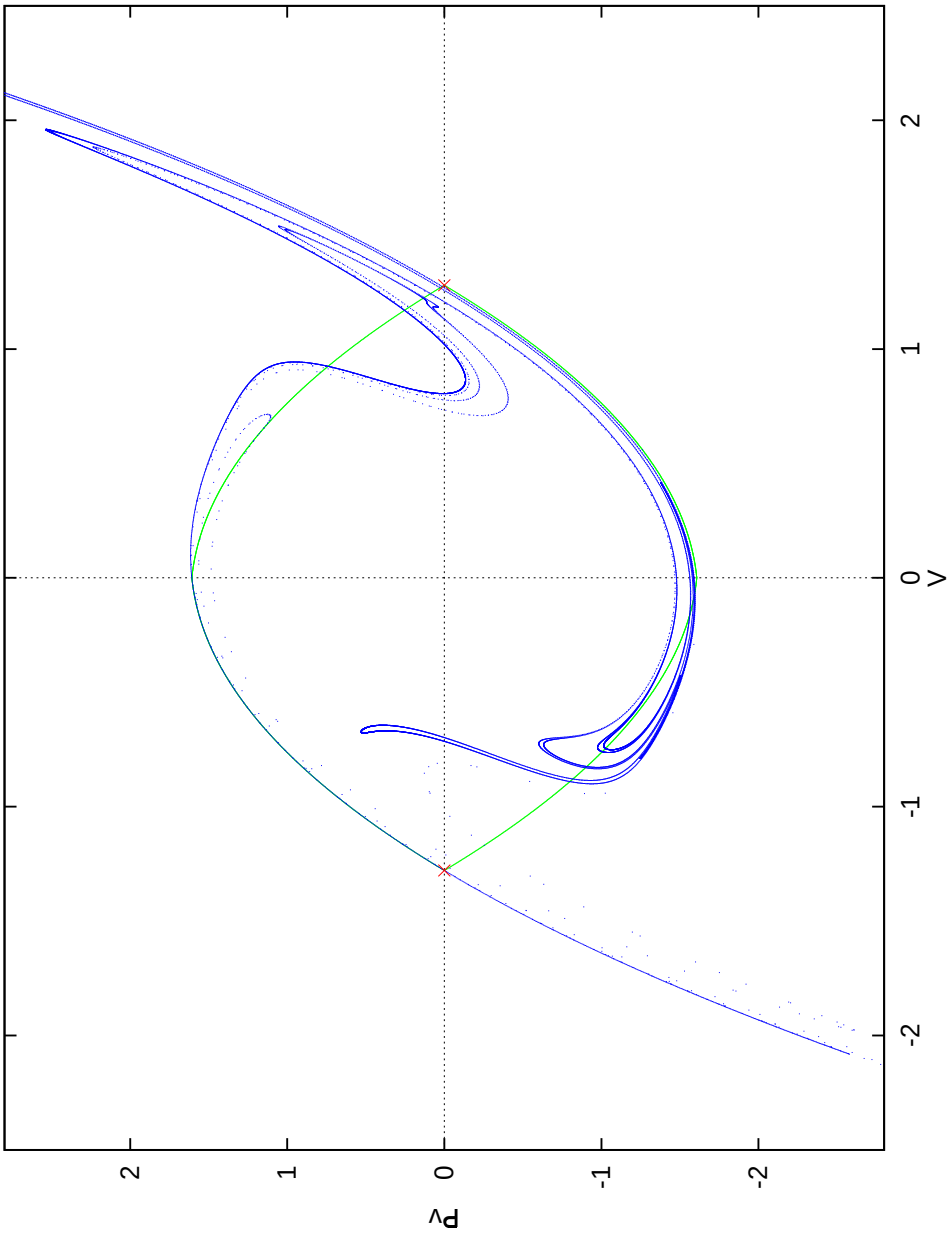

Figure 16: The sixth iterate of ℓ.

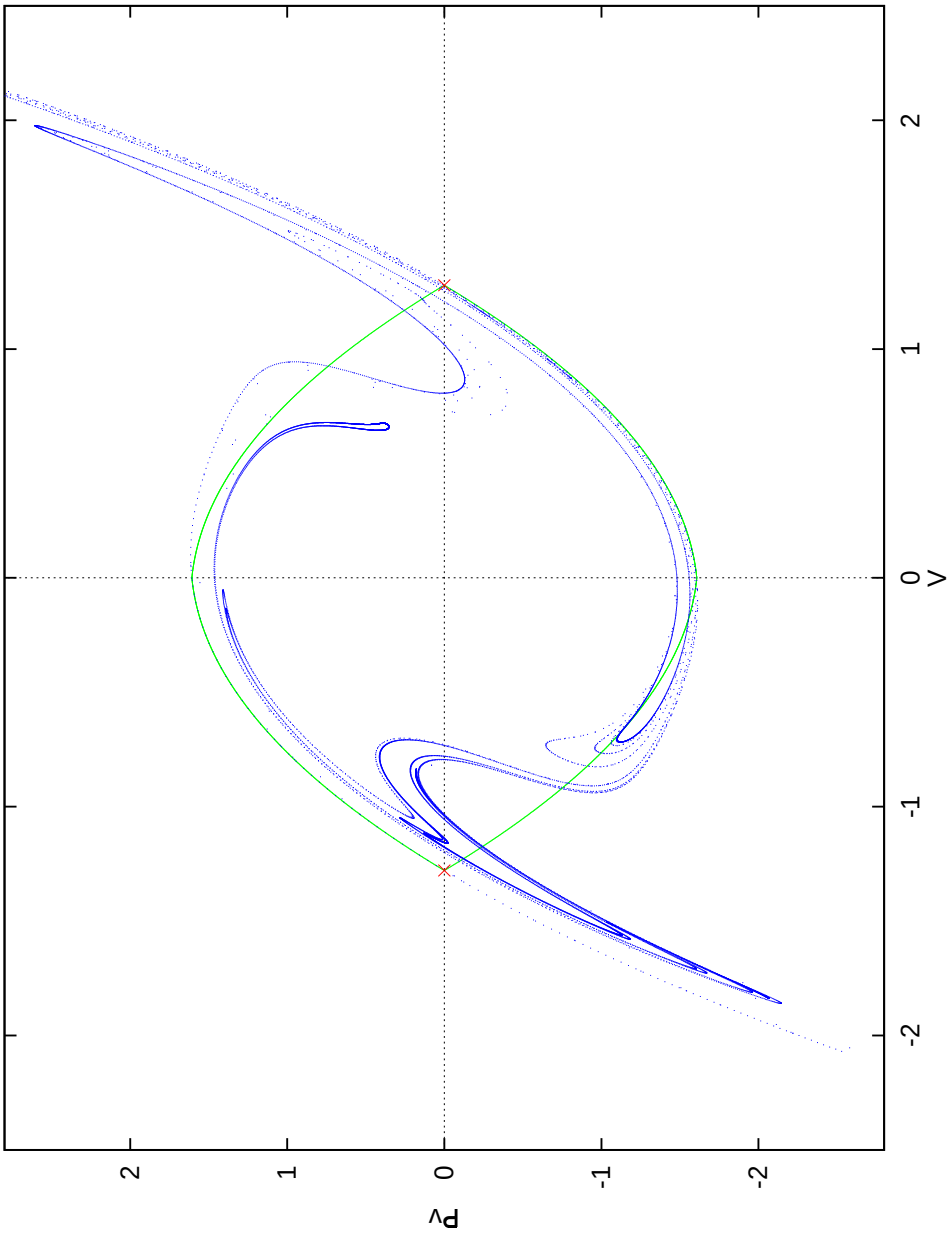

Figure 17: The seventh iterate of ℓ. First escape with $n_v = 4$.

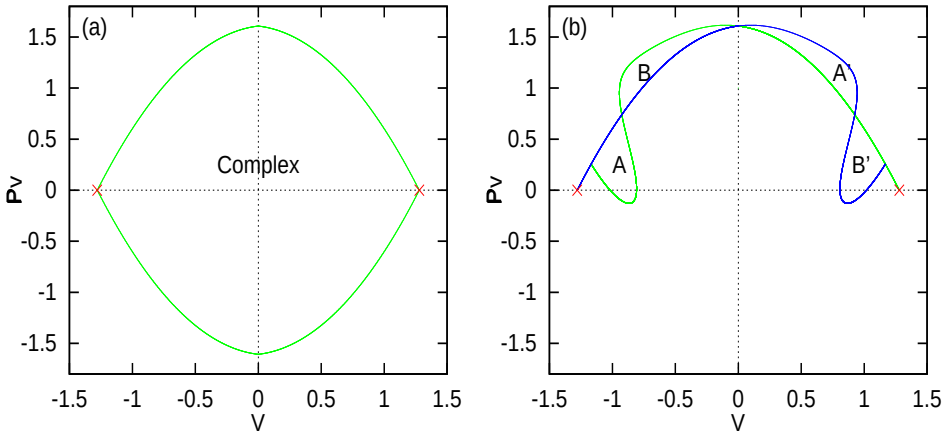

Figure 18: The complex (a) and the upper-half plane turnstile (b). Points in the region A arrive in A' after one iteration of the map; furthermore this is the only way out of the complex in the upper-half plane. Scattering orbits enter the complex by going from B to B'. There exists a conjugate turnstile in the lower-half plane.

condition, $\hat{v}^2 - \hat{u}^2 \geq 2$, and it will never return to the complex. Therefore we redefine the "bound" region of phase space to be the points within the complex, and we say the electron has escaped if it goes from inside the complex to outside.

Shown in Figure 18 is the first pair of lobes of both the upper inset and outset; there is an equivalent structure in the lower half-plane (the system is symmetric about $\hat{v}$ and $\hat{p}_v$). This structure is called the turnstile because points moving from inside to outside the complex, or vice versa, must move through it[6]. As we are interested in dynamics as time moves forward, we focus attention on the regions A and A'. Trajectories inside the complex which will pass outside of it and escape with $\hat{v} > 0$ all go from A to A', thus making region A the "doorway to escape. Note that trajectories passing through the upper-right escape region will have n_v odd, while those passing through the lower left will have n_v even.

4.2 Recasting the escape time function

To better identify features in the escape time function suggested by study of the heteroclinic tangle, we recast our depiction of the function. Previous representations have been for a range of initial conditions having $\hat{v} = 0$ and some range in $\hat{p}_v$. This is the most physical choice of initial conditions, and corresponds to what would be seen in experiment. Anticipating escape times which can be more easily understood in terms of the structure of the heteroclinic tangle, we choose our initial conditions to be along the outset of the left fixed point from $\hat{v} = 1.2788$ to $\hat{v} = 0$. This corresponds to the the portion of the upper-left outset (the blue curve) that bounds regions A and B in Figure 10. Figure 19 shows escape times along this curve, as well as the dependence of n_v and n_i, the number of iterates before a particular initial condition left the complex. This final graph shows strong Cantor-like qualities.

5 Scaling

5.1 The scaling parameter alpha

It was stated previously that chaotic trajectories which start close to each other diverge exponentially. That is, the distance between them should scale as [3] [8] [9]

$$\Delta\hat{v} = \Delta\hat{v}_0\alpha^i$$

where $\Delta\hat{v}_0$ is the initial distance between the trajectories, α is a scaling parameter, and i is the number of iterations of the map. We anticipate that the scaling parameter should play a role in the structure of the escape time function, and determine its value

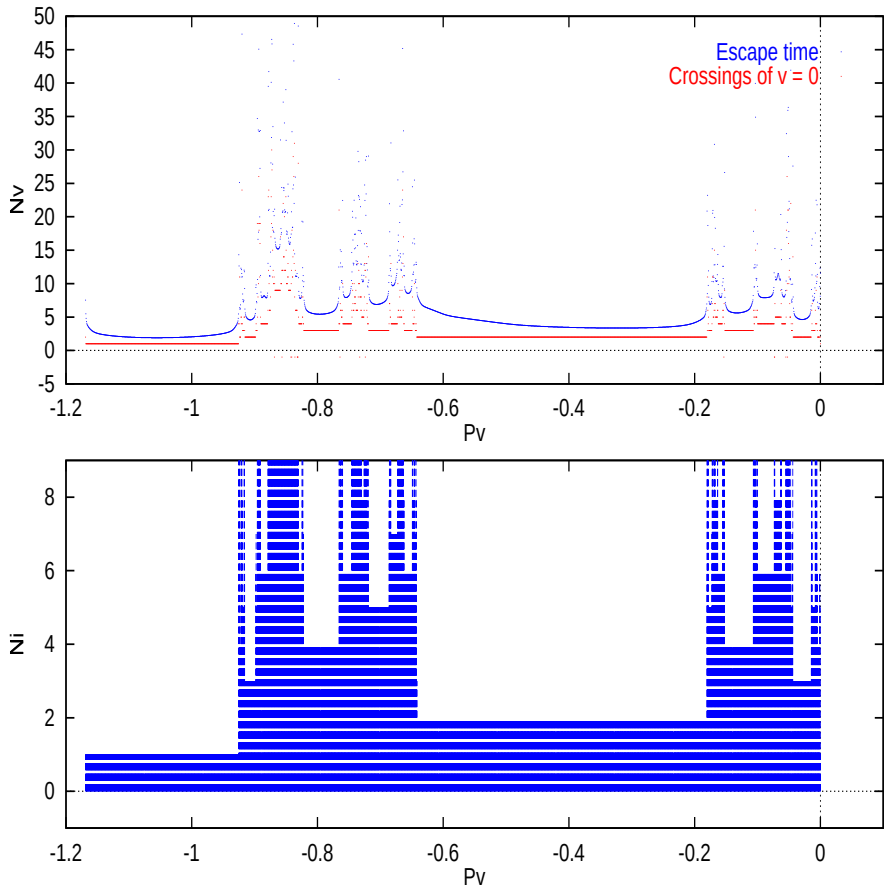

Figure 19: Three representations of escape along the outset of the left fixed point. Note the Cantor-like features of the third graph, showing middle-third escape regions, and some even-fifths as well.

below.

We measure the divergence in the distance between two near trajectories on the outset from the left-hand fixed point. Figure 20 shows the evolution of these distances. A linear fit to these points gives a $\alpha = 9.0085 \pm 0.0083$. This is a fairly large value; points diverge quickly in this system. Because of this quick divergence, it is often possible to calculate only a handful of points before running into numerical limitations when measuring such divergences.

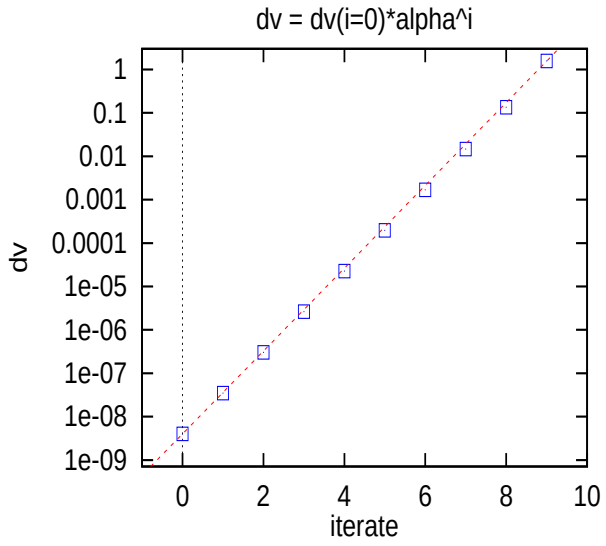

Figure 20: Two trajectories, both very near the left fixed point, were started on its outset. Plotted above are the separations in v of successive iterates of these trajectories. A fit of the form $\Delta v = \Delta v_0 \alpha^i$ gives $\alpha = 9.0085 \pm 0.0083$.

Such exponential divergence may be understood in terms of the insets and outsets, manifolds in the Poincaré plane departing or approaching an unstable fixed point. Trajectories iterating along the separatrices approach or recede from the unstable fixed point; and only in the limit of infinite time do they reach the X point. In this context we may understand that, given any window in time of fixed duration, the distance the between successive crossings of the Poincaré plane in each progressive window decreases (if on an inset) and increases (if on an outset) exponentially. The parameter α is a measure of this exponential change.

5.2 Scaling, escape, and the heteroclinic tangle

We have focused our attention on the escape times of points on the outset of the left fixed point. Specifically, we have chosen that region of the outset which forms

the turnstile; we identify this segment by finding the endpoints of the lobes of the turnstile and taking them one iterate backwards in time. We subdivide this segment into two smaller pieces, corresponding to the inner and outer lobes. Points on the outer lobe escape after the next crossing of the line $\hat{v} = 0$, while points on the inner lobe remain bound at least for the next crossing of the line $\hat{v} = 0$. This process of back-iteration of the turnstile may be repeated infinitely, creating an alternating pattern of escaping and surviving (in the near-future sense) segments the lengths of which scale as α. We will see that a Cantor-like escape structure is generated by mapping this segmenting pattern onto sub-segments of the heteroclinic tangle in a recursive process. The manner of this mapping is specific to the system, but it may be described as a process of stretching and folding. The first few steps are made explicit below.

We already stated that points in the complex escape to the right only by entering region A'; they escape to the left only by entering a corresponding region in the lower-left quadrant. Those escaping to the right or to the left have crossed $v = 0$ an odd or even number of times respectively. Therefore we rename the escape zone $A' \equiv E^{odd}$, and its conjugate region in the lower-left quadrant is E^{even}.

Any section of the tangle intersecting with the escape lobes E^{odd} and E^{even} will escape. Let $\ell(0)$ represent the curve of initial conditions (the portion of the blue curve in Figure 18 bounding regions A and B). Let $\ell(i)$ be the i^{th} iterate of ℓ. The portions of $\ell(i)$ that escape on the i^{th} iterate have $n_i = i$ and some n_v value; these escaping portions are called $\lambda(n_i; n_v)$. These are the segments of the one-dimensional curve $\ell(i)$ which lie within the two-dimensional region in phase space E^{odd} or E^{even}.

This may be succinctly stated as follows:

$$\ell(i) \bigcap E^{even} = \lambda(n_i, n_v = even)$$

$$\ell(i) \bigcap E^{odd} = \lambda(n_i, n_v = odd)$$

Note that several subsections of ℓ can be specified at once by this notation. A single iterate of ℓ may, and indeed for higher i often does, overlap multiply both E^{odd} and E^{even}.

We need not view the escape lobes as static entities. Letting $E^{even}(0)$ represent an escape lobe in the A' turnstile position, $E^{even}(-j)$ represents its j^{th} back-iterate. We know that any portion of ℓ intersecting with the i^{th} back-iterate of E^{even} will escape j iterations in the future. Such a segment will simply iterate forward in time in such a way as to overlap $E^{even}(0)$. That is, the back iterates of $E^{even,odd}(0)$ are equally valid escape conditions, or

$$\lambda(n_i, n_v) = \ell(n_i) \bigcap E^{even,odd}(0) = \ell(n_i - j) \bigcap E(-j)$$

By viewing a surface of section showing some $\ell(i)$ and the back-iterates of an escape region (say $E^{even}(-1, -2, -3)$, we can identify the $\lambda(n_i; n_v)$ before they actually reach $E^{even}(0)$. Consider the particular segment of $\ell(2)$ depicted in Figure 21. It intersects with back iterates $E^{even}(i)$ $i = 0 \ldots \infty$. The overlaps with each lobe will escape with the same n_v (in this depiction $n_v = 2$) and different n_i. These lobes are anchored on the manifold defining the boundary of the complex; the lengths between their anchor points are exactly the same sort of lengthss the scaling of which was used to determine α. In this way the escape/survive segmenting pattern, characterized by

α, is mapped onto the portion of ℓ which survived its first pass of E^{odd}; that is, all of ℓ except that attributable to $\lambda(1;1)$. As ℓ iterates it wraps around the complex and the pattern will continue to be mapped onto surviving portions of ℓ. The mapping takes the form of an exponential grid imposed on the Poincaré plane by back-iterates of the escape region in the vicinity of an inset. Figures 21 and 22 depict this scenario.

5.3 Non-generic traits

A first glance at Figure 19 shows Cantor like features, but it is clear there is additional complexity present. To better understand this it is useful to distinguish what will be referred to as generic and non-generic behavior of the system. Generic behavior constitutes all escape segments the lengths of which correspond to a α^{-i} scaling of the length of a "base" escape segment. Non-generic behavior constitutes the set of all such base escape segments. We shall see that base segments pertain to the tips of lobes in the heteroclinic tangle.

The heteroclinic tangle develops as a series of lobes, formed by the outset, which extend off of the inset. Though there may be much folding in a given lobe, observation indicates that there is always a central tip to the lobe (there is an observed symmetry in the ordering of n_v and n_i values of escape segments for any given iterate of ℓ). We define a tip as an escape segment λ^{ter} or pair of escape segments ${\lambda_{a,b}}^{quint}$ having $n_v = n_v^{tip}$ and $n_i = n_i^{tip}$ for which *both* of the nearest neighboring escape segments (one on either side, in the sense of $\ell(0)$) of equal n_v have $n_i = n_i^{tip} + 1$. Though the tip of each new lobe does not always broach a new n_v value, any new n_v value will

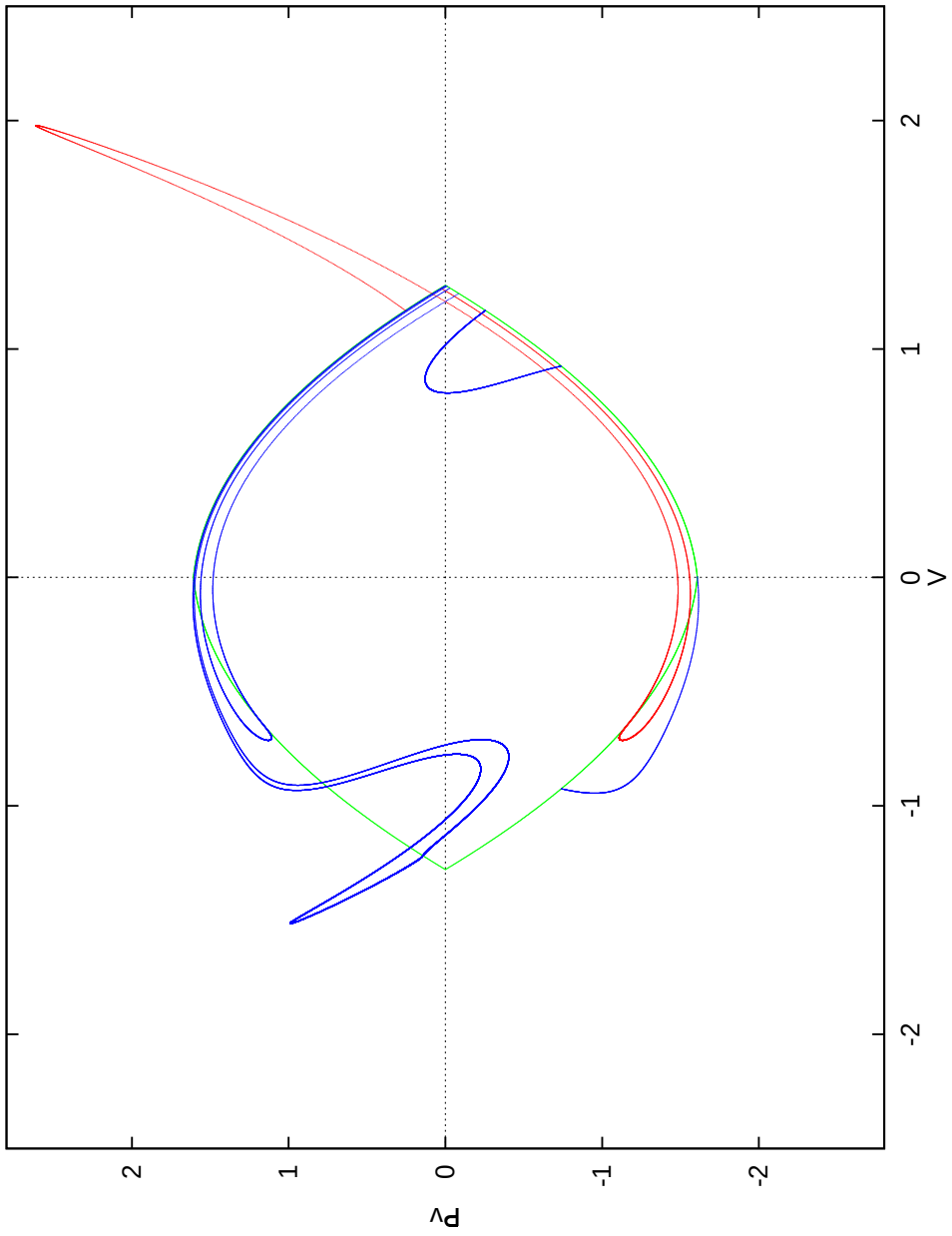

Figure 21: Intersections of $\ell(2)$ with back iterates of E^{even}. The tertiary tip is escaping through $E^{even}(0)$, while the first three generic segments intersect $E^{even}(-1)$, $E^{even}(-2)$, and $E^{even}(-3)$, respectively.

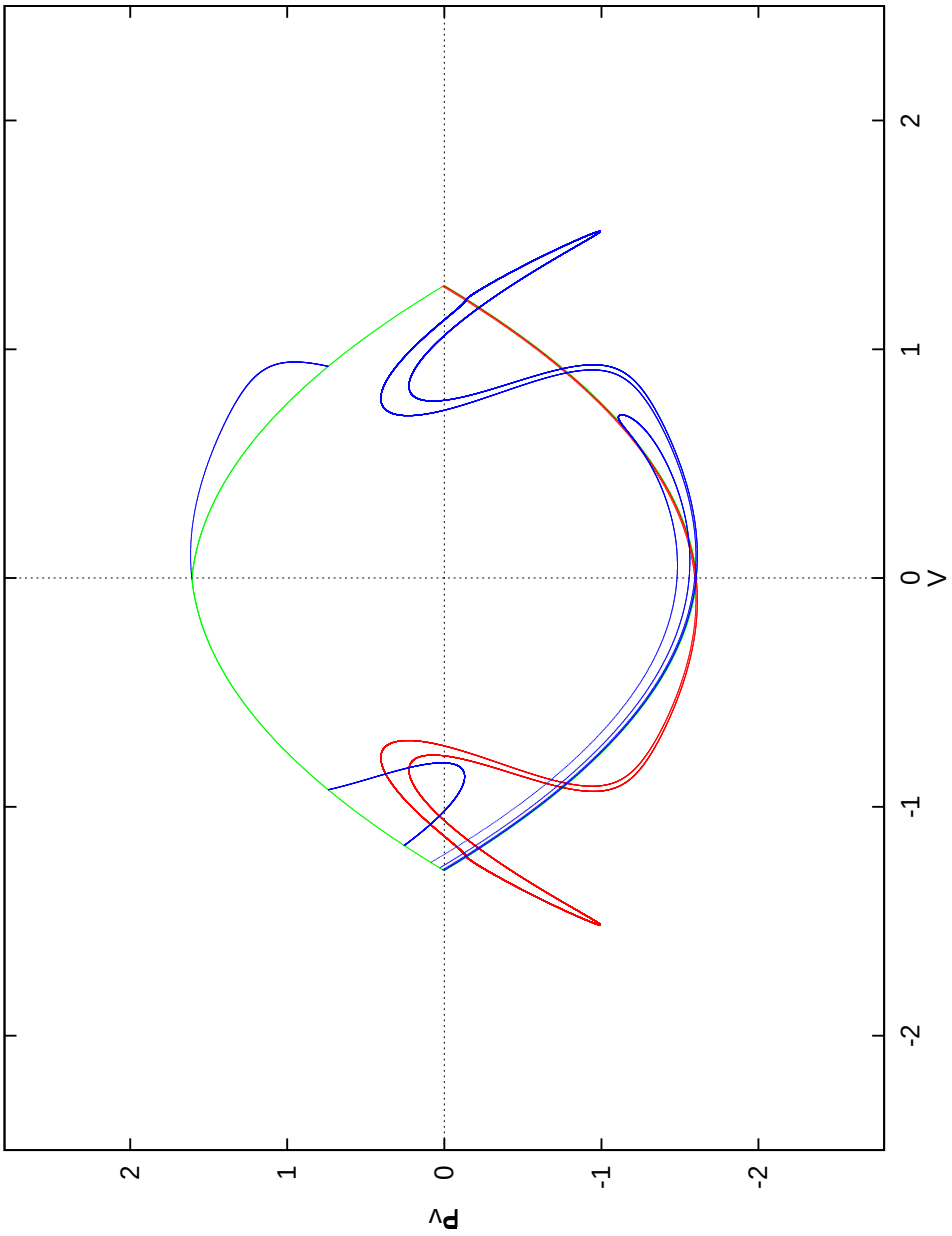

Figure 22: Intersections of $\ell(4)$ (red curve) with back iterates of E^{odd} (blue curves). There are two new base escape segments (non-generic segments which are the base of a generic α-scaling sequence of escape segments) overlapping each back-iterate of E^{odd}, always in what may be informally recognized as a tip of the lobe formed by the back-iterate.

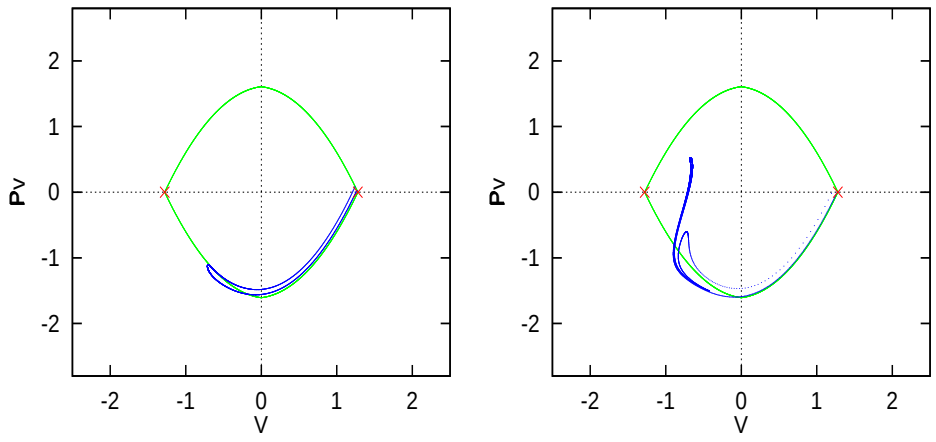

Figure 23: Tertiary and quintenary tips. Tertiary tips are composed of three sub-segments which survive beyond n_i, escape at n_i, and survive beyond n_i, respectively. Quintenary tips are composed of five sub-segments which survive, escape, survive, escape, and survive, respectively.

first be broached by the tip of a lobe. In this sense, the tips of lobes are the source of non-generic behavior. There are fundamentally two circumstances in which the tip of a lobe may be found; they are shown in Figure 23.

In the first, the tip is found to be in an escape region, and thus outside the complex; this is referred to as a tertiary tip. In the second, the tip is inside the complex, creating a quintenary tip. A tertiary tip creates a middle thirds escape pattern: there is a single region that escapes at n_i with some n_v; on either side of it is a region that remains bound until some larger n_v; next to those are two regions which escape at $(n_i + 1, n_v)$, and so on. A quintenary tip creates an even fifths escape pattern; the four crossings of the escape region boundary divide the segment into fifths, with the tip representing the middle fifth, which remains bound. Instead, the sub-segments adjacent to the tip (the even fifths, ${\lambda_a}^{quint}$ and ${\lambda_b}^{quint}$) escape on the n_i

iteration.

Note that the lengths of each escaping sub-segment of the lobe tip need not be equal; this is due both to the non-constant density of points along the manifold and the different positions the sub-segments occupy within the escape region. These single (tertiary) and double (quintenary) escape segments are the base escape segments referred to above (given that they belong to the first lobe entering the escape region with a particular value of n_v). All future segments escaping with that n_v are mapping near one of these base segments; if the base segment escaped with $n_i = n_{i_0}$, escape segments with the same n_v and $n_i = n_{i_0+j}$ will scale as $\alpha^{n_{i_0} - n_{i_0+j}}$. We have not been able to see any long-time, long-term pattern in the base escape segments in this system. Also we are sure that if there were any such long-time pattern in this system, the pattern would be different in other systems. Therefore we say that the type and length of base segments at each n_i and n_v are non-generic properties.

Note that, according to the definition of a tip given above, a tip is not uniquely what is often meant in the colloquial sense of the word (as in, for example, the tip of a finger). As ℓ iterates forward, it approaches closely and occasionally wraps around previous iterates of itself. These previous iterates had tips (in the colloquial sense), and a portion of ℓ wrapping around one of the old tips qualifies as a tip itself. Thus, although each lobe has a central tip, it may have multiple tips according to the above definition, each corresponding to a base escape segment.

5.4 Generic traits

We can show that there are generic traits which may be understood after the non-generic properties have been observed. Figure 24 does just this. As discussed above, escape segments which come from lobe tips in ℓ constitute non-generic features of the system. They are followed by pairs of escape segments $\lambda(n_i + j, n_v)$ where $j = 1, 2, 3, \ldots$ such that $\lambda(n_i + j, n_v) = \alpha^{-j}\lambda(n_i, n_v)$. Escape segments from tertiary and quintenary tips alike possess α-scaling sequences composed of pairs of escape segments, despite the fact that tertiary tips incorporate a single escape segment, while quintenary tips incorporate two.

We suggest that the escape-time curve of the parallel fields represents a novel sort of fractal dependence which we term epistrophic. The generic scaling described above is typical of fractal structures; however, the presence of multiple base segments is a complication. Despite these multiple bases, all lengths scale as a single parameter α. Multifractals, by contrast, present a single basic structure which scales as multiple parameters, or a spectrum of parameters. Although it is possible that there is a pattern to the appearance of base escape segments which is yet to be identified, that pattern would be destroyed under a change of parameters. Instead, we propose that the escape-time dependence is a fractal variant having multiple bases and a single scaling parameter, as opposed to the single basis and multiple parameters of a multifractal. We name this an epistrophic fractal, after the rhetorical device of epistrophe, in which multiple sentences end in the repetition of a single phrase. This is intended to draw analogy to the introduction of new base segments followed by

scalings thereof.

6 Summary

The dependence of ionization times on initial conditions may be a fractal function in the parallel fields system; that it is has been explored but not conclusively demonstrated. We have given observational motivation for the presence of structure-within-structure, a feature of the system more readily apparent for regions of relatively quick ionization. The ionization time function looks something like the result of an iterative escape process, such as the evolution of the Cantor set which occurs under repeated applications of the tent map. We have sought a similar iterative escape mechanism in the stretches and folds of the system's heteroclinic tangle. This complex structure in phase space results from the intersection of separatrices of two unstable fixed points and generates chaotic motion in the system. As chaotic orbits diverge exponentially, it is not surprising that the ionization time of two neighboring orbits should show little correlation.

The heteroclinic tangle is composed of two manifolds in phase space which cross each other repeatedly in a manner so as to preserve the area enclosed by successive crossings. Any single trajectory iterates discretely from point to point on one of the manifolds (or separatrices or inset/outsets). The smooth curve in phase space which is the manifold is composed of a family of such trajectories. As trajectory points iterate along the manifold, the distance between points along the manifold increases or decreases exponentially, depending on whether the manifold is an outset

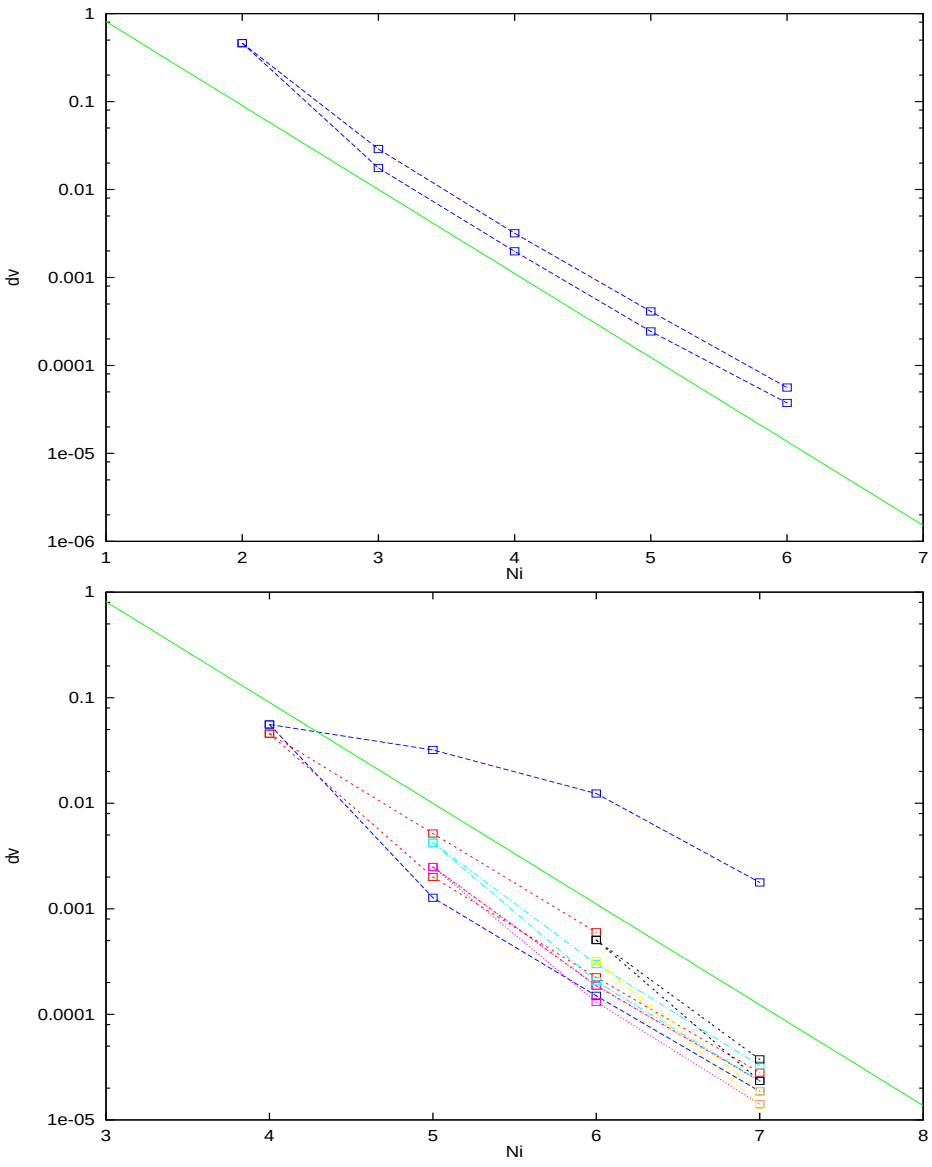

Figure 24: The evolution of escape segments $\lambda(n_i; 2)$ (first plot) and $\lambda(n_i; 3)$ (second plot). In both cases we see only bases generated by tertiary tips, which split to produce two chains of generic escape segments. Colors are used to affiliate base segments and their scaled shadows. The plotted line is of slope α to indicate that escape segment lengths do, indeed, scale as α.

or an inset. The length between any two successive iterates is a factor of α larger or smaller than the lengths between neighboring iterates.

This is significant because the points at which inset and outset cross, like any other points on a manifold, scale in this way. Lobes of the tangle extend off of the crossing points. The spacing and width of these lobes scale as α, and near the manifold off of which they extend they are like logarithmic demarcations of the surface of section. Further away from this manifold, they develop in a system-specific fashion, but locally every tangle will impose this grid pattern on the Poincaré plane.

In a qualitative sense, ionization may be understood in terms of this localized grid pattern. Back-iterates of a lobe which is an escape region establish the grid. Portions of the tangle which intersect the back-iterates away from the gridded region of the surface of section, near the tip of the escape lobe, constitute what we have termed base escape segments. As they iterate forward, they are drawn increasingly closer to the inset and, correspondingly, the gridded region. Thus the tails of tips scale as α, and this scaling improves with successive iterations, as the base segment is "pulled" closer to the inset.

The exponential grid in phase space establishes a comfortable qualitative understanding of the ionization process. We have given evidence that the dependence of escape time on initial conditions has a pattern which we call an epistrophic fractal. At each (n_i, n_v) there is a set of "base segments" that escape at that (n_i, n_v). We find no pattern to these base segments. Each base segment is followed by a sequence of segments that escape at (n_{i+j}, n_v). The lengths of these segments scales as α^{-j}; more precisely $\lim_{j\to\infty} |\lambda(n_{i+j+1}, n_v)/\lambda(n_{i+j}, n_v)| = \alpha$.

Acknowledgements

I would like to thank John Delos for his instruction, patience, and enthusiasm, each in their appropriate time. I also thank Dongmei Wang, who has been an exuberant and much appreciated source of encouragement.

References

[1] C. Jaffé, D. Farrelly, and T. Uzer, "Transition State in Atomic Physics," *Phys. Rev. A.* **60** (5), 3833-3850 (1999).

[2] A. A. Flower, "The Excited Hydrogen Atom in Parallel Electric and Magnetic Fields," (College of William & Mary Undergraduate Thesis, 1999).

[3] H. Hastings and G. Sugihara, *Fractals: A User's Guide for the Natural Sciences*, (Oxford Science Publications, 1993).

[4] M. L. Du and J. B. Delos, "Effect of Closed Classical Orbits on Quantum Spectra: Ionization of Atoms in a Magnetic Field," *Phys. Rev. Lett.* **58**, 173-33 (1987).

[5] L. D. Landau and E. M. Lifshitz, *Course of Theoretical Physics: Mechanics*, 1st ed. (Addison-Wesley, Reading, MA, 1960).

[6] R. C. Hilborn, *Chaos and Nonlinear Dynamics*, (Oxford University Press, 1994).

[7] M. Tabor, *Chaos and Integrability in Nonlinear Dynamics: An Introduction*, (John Wiley & Sons, Inc., 1989).

[8] A. Tiyapan and C. Jaffé, "Classical Atom-Diatom Scattering: Self-Similarity, Scaling Laws, and Renormalization," *J. Chem. Phys.* **99** (4), 2765-2780 (1993).

[9] J. L. McCauley, *Chaos, Dynamics, and Fractals*, (Cambridge University Press, 1993).

[10] R. Blumel and W.P. Reinhardt, *Chaos in Atomic Physics*, (Cambridge University Press, 1997).

Appendix A: Code

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C escape.f                                                                    C
C AUTHOR:   Brian Tighe                                                       C
C PURPOSE:  escape numerically integrates classical trajectories of the       C
C           Hydrogen atom in parallel electric and magnetic fields.           C
C           It outputs u-v space representations of those trajectories,       C
C           surfaces of section in the v-pv Poincare plane, and               C
C           escape times, iterates, and crossings of the line v = 0 as a      C
C           function of an initial condition.  These initial conditions       C
C           may be any user-specified, single-valued function in phase        C
C           space.                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      PROGRAM escape
      implicit none

         real*8      Hamiltonian
         real*8      PVFIT

         integer*4   neqn, nw, i
         integer*4   numSosMax, numSosMin
         parameter   (neqn = 5, nw = 100 + 21*neqn)
         real*8      energy, B, Pi, tMax
         logical     plotTraj, plotSoS, plotSep, recordIts
         common/phys/energy, B
         common/num/Pi
         common/logic/plotTraj, plotSoS, plotSep, recordIts
         common/end/tMax
         common/SoS/numSosMax, numSosMin

         real*8         y0(neqn), y1(neqn)
         real*8         u, v, v0, pv, pu, pv0, pvMax, pvMin, puSq
         real*8         vMin, vMax, vtest
         real*8         h
         real*8         t, tau, tau0, tau1, tauStep, tauBaby
c         real*8         c0, c1, c2, c3
         integer*4      numTraj, currSos
         integer*4      iflag
         integer*4      nv   !# of times traj crosses v = 0
         logical        recordTime, recordLoops
         logical        finish
         logical        changeV, changePv

         open (11, file = 'sos.d')     ! file 11 -> SoS data
         open (21, file = 'ni.d')      ! file 21 -> other SoS data
         open (31, file = 'orbit.d')   ! file 31 -> trajectory data
         open (41, file = 'nv.d')      ! file 41 -> times across v = 0
         open (51, file = 'tescape.d') ! file 51 -> escape times
         open (61, file = 'info.d')    ! file 61 -> misc. info
         open (71, file = 'input.d')   ! file 71 -> input data
```

```
110      format(1h , 4(e15.8, 2x))
111      format(1h , 4(e17.10, 2x), 2(i7))
310      format(1h , 4(e12.5, 2x))
510      format(1h , 3(e14.7, 2x), i5)
511      format(1h , 3(e13.6, 2x), i3, (e14.6, 2x), i6)
410      format(1h , (e14.7, 2x), 2(i5), (e15.7, 2x), i6)
610      format(' i=', i6, ' pv=', g12.5, ' outside allowed region')
611      format(' failed on trajectory',i5,' iflag=',i3,
     +                  ' pv0=',g12.5,' tau1-tau0=',g12.5)

         Pi = 4.0d0*datan(1.0d0)
c    c0 =  1.6067
c          c1 =  0.17313
c          c2 = -0.76589
c          c3 =  0.063494

C        Initialize from input file

         read (71,*) energy            ! read energy
         read (71,*) B                 ! B-field
         read (71,*) tMax              ! max integration time
         read (71,*) tauStep
         read (71,*) numTraj           ! # of trajectories to compute
         read (71,*) plotSep
         read (71,*) changeV
         read (71,*) vMin
         read (71,*) vMax
         read (71,*) changePv
         read (71,*) pvMin
         read (71,*) pvMax
         read (71,*) plotTraj
         read (71,*) plotSoS
         read (71,*) numSosMin         ! first saved SoS
         read (71,*) numSosMax         ! max # of passes through the SoS
         read (71,*) recordTime
         read (71,*) recordLoops
         read (71,*) recordIts



         if( plotTraj ) then
           tauStep = tauStep / 10.0   ! take baby steps if plotting
         endif
         tauBaby = tauStep / 100.0

         do i = 1 , numTraj !trajectory loop

            iflag   = -10
            nv      =   0

C        Set trajectory initial conditions

            t    = 0.0d0
```

```
            tau  = 0.0d0
            tau0 = 0.0d0
            u    = 0.0d0
            if (plotSep) then
               v   = vmin + ((vmax - vmin)/(numTraj-1.))*(i-1.)
c               write(*,*) v - vtest
               vtest = v
               v0  = v
               pv  = PVFIT(v)
               pv0 = pv
            else
               if (.not. changeV) v = 0.0d0
               if (changeV .and. numTraj .ne. 1)
     +             v = vmin + ((vmax - vmin)/(numTraj-1.))*(i-1.)
               if (changeV .and. numTraj .eq. 1)
     +             v = vmin
               v0   = v
               if (.not. changePv) pv = 0.0d0
               if (changePv .and. numTraj .ne. 1)
     +             pv = pvmin + ((pvmax-pvmin)/(numTraj-1.))*(i-1.)
               if (changePv .and. numTraj .eq. 1)
     +             pv = pvmin
               pv0  = pv
            endif

            pu   = 0.0d0
            puSq = -2.0 * Hamiltonian( u, v, pu, pv )
            if( puSq .lt. 0.0d0 ) then
               write( 61, 610) i, pv
               continue
            endif
            pu = dsqrt( puSq )
            h  = Hamiltonian( u, v, pu, pv )

            if( plotTraj ) then
               write( 31, 310 ) u, v
            endif

            tau0  = 0.0
            Y0(1) = u
            Y0(2) = v
            Y0(3) = pu
            Y0(4) = pv
            Y0(5) = t

            currSos = 0
            finish  = .false.

            if (plotSos .and. currSos .ge. numSosMin)
     +          write(11,111) v, pv, v0, pv0, nv, i

            do while (currSos + 1 .le. numSosMax
     +                .and. finish .eq. .false.)   !SoS loop
```

```
c               if (plotSos .and. currSos .ge. numSosMin)
c     +             write(11,111) v, pv
                currSos = currSos + 1

                CALL TRAJ( tau0, y0, tauStep, tauBaby, nv,
     +                tau1, y1, iflag)
c               write(*,111) tauBaby,  tau1-tau0
C               Test for success:

C               If it fails:
                if( iflag .ne. 7 .or.
     +            ( abs(tau1 - tau0) .lt. tauBaby )) then
                   write(61,611)  i, iflag, pv0, tau1 - tau0
                   write(*,611)   i, iflag, pv0, tau1 - tau0
                   write(*,*) v0, currSos
                   return

C               Otherwise, it succeeded; prepare to record data:
                else

                   u   = y1(1)
                   v   = y1(2)
                   pu  = y1(3)
                   pv  = y1(4)
                   t   = y1(5)
                   h   = Hamiltonian( u, v, pu, pv )

C                  If escaped, record original pv and escape time

                   if (iflag .eq. 7) then
c                      if (dabs(v*v - u*u - 2.0) .lt. 1.0d-05) then
                      if (v*v - u*u - 2.0 .gt. -1.0d-05
     +                    .and. .not. recordIts) then
c                         if (recordTime) then
c                            if (v .gt. 0) write(51, 510) pv0,  t
c                            if (v .lt. 0) write(51, 510) pv0, -t
c                         endif
                          if (recordTime) then
                             if (changeV) write(51, 510) v0, t
                             if (changePv) write(51, 510) pv0, t
                          endif
                          if (recordLoops)
     +                       write(41, 410) v0, nv, currSos
c                         if (recordIts)   write(21, 410) v0, currSos
                          write(*, 511) pv0, v0, t, nv, h, i
                          if (plotSos) finish = .true.
                       endif
c                      if (dabs(t - tMax) .lt. 1.0d-05 ) then
                       if (tMax - t .lt. 1.0d-05
     +                     .and. .not. recordIts) then
                          if (recordTime)  write(51, 510) pv0, t + 2.0
                          if (recordLoops) write(41, 410) v0, -1, -1
c                         if (recordIts)   write(21, 410) v0, -1, -1
                          write(*, 511) pv0, v0, t, -1, h, i
```

```
                     if (plotSos) finish = .true.
                  endif
                  if (plotSoS) then
                     if (currSos .ge. numSosMin)
     +                   write(11, 111) v, pv, v0, pv0, nv, i
                     if (currSos .eq. numSosMax
     +                   .and. finish .eq. .false.) then
                        write(*, 511) pv0, v0, t, nv, h, i
                        if (recordTime)  write(51, 510) v0, t
                        if (recordLoops)
     +                     write(41, 410) v0, nv, currSos
c                        if (recordIts)   write(21, 410) v0, currSos
                     endif
                  endif
                  if (recordIts) then
                     if (dabs(v) .gt. 1.278827859) then
                         write(21, 410) v0, currSos, nv
                         write(*, 511) pv0, v0, t, nv, h, i
                         finish = .true.
                     endif
                     if (dabs(v) .le. 1.278827859) then
                        if (v .le. 0.0 .and.
     +                        dabs(pv) .gt. PVFIT(v)) then
                           write(21, 410) v0, currSos, nv
                           write(*, 511) pv0, v0, t, nv, h, i
                           finish = .true.
                         endif
                         if(v .gt. 0.0 .and.
     +                       dabs(pv) .gt. PVFIT(-v)) then
                           write(21, 410) v0, currSos, nv
                           write(*, 511) pv0, v0, t, nv, h, i
                           finish = .true.
                         endif
                     endif
                     if (currSos .eq. numSosMax
     +                   .and. finish .eq. .false.) then
                        write(21, 410) v0, currSos, nv
                        write(*, 511) pv0, v0, t, nv, h, i
                     endif
                  endif
               endif

               tau0  = tau1
               Y0(1) = u
               Y0(2) = v
               Y0(3) = pu
               Y0(4) = pv
               Y0(5) = t

            enddo               !SoS loop

         enddo                  !trajectory loop
```

```
         stop

         end                     !escape program



      real*8 FUNCTION Hamiltonian( u, v, pu, pv )
         implicit none

         real*8 u, v, pu, pv, energy, B
         common/phys/energy, B

         Hamiltonian = (pu**2 + pv**2)/2.0d0 - energy*(u**2 + v**2) +
     +                 (B**2/8.0d0)*(u**4*v**2 + u**2*v**4) +
     +                 (u**4 - v**4)/2.0d0 - 2.0d0

         return

      end !Hamiltonian function


      real*8 FUNCTION PVFIT( v )
         implicit none
         real*8  v
         real*8 c0, c1, c2, c3, c4, c5, c6, c7, c8, c9

 c0 =  1.6077      !1.6067
         c1 =  0.15465     ! 0.17313
         c2 = -0.78232     !-0.76589
         c3 =  0.10549     !0.063494
         c4 =  0.047889
 c5 =  0.011052
                           !c6 =  0.010355
                           !c7 =  0.051993
                           !c8 =  0.040297
                           !c9 =  0.010101



         PVFIT = c0 + c1*v + c2*v**2 + c3*v**3 + c4*v**4 + c5*v**5 ! +
c     +             c6*v**6 + c7*v**7 + c8*v**8 + c9*v**9

         return

      end !PVFIT function




      SUBROUTINE TRAJ( tau0, y0, tauStep, tauBaby, nv, tau1, y1, iflag)
      implicit none

         real*8         Hamiltonian
```

```
      external       DP0DRT
      external       tderiv, g, gb
      integer*4      neqn, nw                      ! currSos
      parameter      (neqn = 5, nw = 100 + 21*neqn)
      real*8         tau, tau0, tau1, tauStep,
     +               tauBaby, tau0ut, tau0utBaby
      real*8         y0(neqn), y1(neqn), y(neqn), v0, v1
      real*8         work(nw)
      integer*4      iwork(6)
      integer*4      iflag, i, nv
      real*8         relErr, absErr, absErrRoot, relErrRoot
      real*8         h

      logical        plotTraj, plotSoS, plotSep, recordIts
      common/logic/plotTraj, plotSoS, plotSep, recordIts
      integer*4      numSosMax, numSosMin
      common/SoS/numSosMax, numSosMin


311   format (1h , 4(e12.5, 2x))
910   format (' tau=',g12.5,' minus tau0=',g12.5,' less than',g12.5)

      relErr      = 1.0d-13
      absErr      = 1.0d-13
      absErrRoot  = 100.0d0*absErr
      relErrRoot  = 100.0d0*relErr
      iflag       = 1

      tau         = tau0

      do i = 1, neqn
         y(i) = y0(i)
      enddo

C     Take a baby step

      tau0utBaby = tau + tauBaby/2.0
      v0 = y(2)
      CALL DP0DRT( tderiv, neqn, y, tau, tau0utBaby,
     +             relErr, absErr, iflag, work, iwork,
     +             GB, relErrRoot, absErrRoot )
      v1 = y(2)
      if (iflag .ne. 2) then
         return                ! problem
      endif

      if ((v0 .lt. 0.0d0 .and. v1 .gt. 0.0d0) .or.
     +     (v0 .gt. 0.0d0 .and. v1 .lt. 0.0d0)     ) then
         nv = nv + 1
      endif

      h = Hamiltonian( y(1), y(2), y(3), y(4) )

      if( plotTraj ) then
```

```
            write( 31, 311 ) y(1), y(2)
         endif

C        Take a big step

         do while (iflag .eq. 2) !integration loop

            tauOut = tau + tauStep
            v0 = y(2)
            CALL DPODRT( tderiv, neqn, y, tau, tauOut,
     +                   relErr, absErr, iflag, work,
     +                   iwork, G, relErrRoot, absErrRoot )
            v1 = y(2)
            tau1 = tau
            do i = 1, neqn
               y1(i) = y(i)
            enddo

            if ((v0 .lt. 0.0d0 .and. v1 .gt. 0.0d0) .or.
     +          (v0 .gt. 0.0d0 .and. v1 .lt. 0.0d0)     ) then
               nv = nv + 1
            endif

            h = Hamiltonian( y(1), y(2), y(3), y(4) )
            if( plotTraj ) then
               write( 31, 311 ) y(1), y(2)
            endif

         enddo !integration loop

C        If successful up to next SoS, output data to plotting file and return

         if( (iflag .eq. 7) .and.
     +       (abs(tau - tau0) .gt. tauBaby) ) then
            h = Hamiltonian( y(1), y(2), y(3), y(4) )
            if( plotTraj ) then
               write(31, 311 ) y(1), y(2)
            endif
            return
         endif

C        If returned to original SoS, cry for help

         if( (iflag .eq. 7) .and.
     +       (abs(tau - tau0) .le. tauBaby) ) then
             return
         endif

C        Otherwise, something funky happened

         return   !I surrender!

       end !subroutine Traj
```

```
      SUBROUTINE TDeriv( tau, yy,    dery)
      implicit none
C     for use by the diff. eq. solver; evaluates time derivatives
C     for the Hamiltonian eq's of motion
C
C     yy(1): u  yy(2): v  yy(3): pu  yy(4): pv  yy(5): t
C     tau  -> independent variable
C     yy   -> dependent variables (a vector)
C     dery -> value of the rhs of the first-order diff. eq.

         integer*4   neqn
         parameter   (neqn = 5)
         real*8      yy(neqn), dery(neqn)
         real*8      u, v, pu, pv
         real*8      tau, t

         real*8      energy, B, Pi
         common/phys/energy, B
         common/num/Pi

         u  = yy(1)
         v  = yy(2)
         pu = yy(3)
         pv = yy(4)
         t  = yy(5)

         dery(1) = pu
         dery(2) = pv
         dery(3) = 2.0*energy*u - 2.0*u**3 -
     +             (B**2/8.0)*(4.0*u**3*v**2 + 2.0*u*v**4)
         dery(4) = 2.0*energy*v + 2.0*v**3 -
     +             (B**2/8.0)*(2.0*v*u**4 + 4.0*v**3*u**2)
         dery(5) = u**2 + v**2

         return

      end !subroutine TDeriv



      real*8 FUNCTION G( tau, yy, dery )
      implicit none
C     When G = 0, DPODRT returns iflag = 7, the success condition (woo-hoo!)


         integer*4   neqn
         parameter   (neqn = 5)
         real*8      yy(neqn), dery(neqn)
         real*8      u, v, tau, t
         real*8      g1, g2, g3

         real*8      energy, B, tMax
```

```
      logical     plotTraj, plotSoS, plotSep, recordIts
      common/phys/energy, B
      common/logic/plotTraj, plotSoS, plotSep, recordIts
      common/end/tMax

      u   = yy(1)
      v   = yy(2)
      t   = yy(5)

      g1  = v*v - u*u - 2.0
      g2  = t - tMax
      g3  = u

      if (.not. plotSos .and. .not. recordIts) G = g1*g2
c                      .eq. 0 if either stop condition met
      if (plotSos .or. recordIts) G = g3

      return

    end !function G


    real*8 FUNCTION GB ( tau, yy, dery )
    implicit none
      integer*4 neqn
      parameter (neqn = 5)
      real*8  tau, yy(neqn), dery(neqn)

      GB = 1.0d0
      return
    end
```