The continuation of invariant circles and breakup due to noninvertibility

Joe Kell and Bruce Peckham

December 21, 2017

Abstract

In a 2003 paper titled "A route to computational chaos revisited: noninvertibility and the breakup of an invariant circle" the authors investigate the transition from an attracting fixed point to an invariant circle to "computational chaos" as the step size in a numerical integration scheme is increased [1]. The transition is anything but simple: there are plenty of complicated local and global bifurcations that occur. The current paper provides numerical evidence that transition from invariant circle to chaos can occur at a single point, but only if one follows an "irrational rotation arc" in a two-parameter space. We use and analyze three numerical methods to look for this transition point. The most successful method involves representing the invariant circle with truncated Fourier series. The breakup involves the interaction of the invariant circle with a critical curve in the phase space. Numerical continuation suggests that, at a key parameter value, the invariant circle loses its smoothness by forming cusps. Beyond this "cusp point", the invariant circle ceases to exist, and chaotic dynamics are possible.

1 Introduction

Discrete dynamical systems is the study of the eventual behavior of sets as they are iterated under specific maps. Some of the most important sets are ones that are *invariant*, or don't change when iterated. These invariant sets can come in all different shapes and sizes and are dependent on the particular map that is being discussed. This paper is going to focus on invariant circles (IC). It is well known from theory that invariant circles are born out of a Hopf bifurcation, but there is no standard way in which they break up. Numerical methods can be used to follow an invariant circle with a fixed irrational rotation number along a curve in a two-parameter space which continues away from the Hopf bifurcation where the invariant circle was born. This paper is focused on finding a point along this curve where the invariant circle ceases to be an invariant circle. This specific point is important because behaviors of circle maps are quite limited, but when the invariant circle breaks apart the dynamical behavior can become much more complicated. Under certain scenarios, this point can be on the boundary between tame circle map behavior, and chaotic dynamics. In fact, we believe that following this irrational arc in parameter space is a typical "route to chaos" for dynamical systems. The feature that is unique about this paper is that we are exploring the breakup of invariant circles in the context of noninvertibility. It is well known that near the Hopf bifurcation the map must be invertible (locally), but as parameters are varied, the invariant circle can interact with the critical set of the map. Noninvertible maps typically "fold" phase space along the critical curves. This folding causes "bumps" in the invariant circle. At a critical parameter value, the bumps can become cusps on the invariant circle. This "cusp point" could be the point of transition between circle map behavior and chaos. This paper provides numerical evidence that this "cusp" point might actually exist.

Circle map primer To give the reader a better understanding of how circle maps behave, examples will be given from the most elementary one-dimensional circle map, rigid rotation. This map $f : S \to S$ is defined by:

$$\theta_{n+1} = f(\theta_n) = \theta_n + \omega. \tag{1}$$

This concept is shown in figure 1 where $\omega = \frac{1.32}{2\pi} \approx .210085$ is an irrational number that is slightly larger than $\frac{1}{5}$. We view S as the unit circle in the plane. In this illustration, the starting point is taken to be $\theta_0 = 0$ which is $(1, 0)$ in the plane. This is indicated by the red dot. In (a) the reader can see that the point's first iterate (slightly farther than a fifth of the way around the circle) is shown in blue. (b) shows this iterate and the next three, each iterate a different color. (c) shows that the fifth iterate (green) goes beyond the starting point, this is because $\omega > \frac{1}{5}$. (d) shows the first 500 iterates which gives the idea that with more iterates, the orbit will be dense in the circle, which only occurs when the rotation number is irrational. This map behaves in a much "simpler" way when ω is rational and in reduced form, $\frac{p}{q}$. It is not difficult to see that iterating any point q times results in rotating around the circle exactly p times coming back to the same point. This makes any point on the circle a period-q point when $\omega = \frac{p}{q}$.

Nonlinear circle maps In more general nonlinear systems, circle maps with irrational rotation numbers are still (conjugate to) rigid rotations, but circle maps with rational rotation numbers are not. Rather than having all points on the circle being periodic, there are typically two period-q orbits, one attracting and one repelling, and all orbits but the repelling period-q orbit are attracted to the attracting period-q orbit. The simplest family exhibiting this behavior is the Arnold Circle Map family, defined by $f(\theta) = \theta + \omega + \alpha \sin(\theta)$. If $\alpha = 0$ we are back to rigid rotations. For an irrational number ω^*, there is an arc in the parameter plane extending from $(\omega, \alpha) = (\omega^*, 0)$ into the $\alpha > 0$ half-plane along which the maps continue to have rigid rotations with rotation number ω^*, and all orbits are dense. In contrast, for rational values p/q, emanating from $(p/q, 0)$ is an "Arnold tongue" which opens up into a horn-shaped region in the parameter plane. Inside the p/q Arnold tongue the corresponding maps have a pair of period-q orbits as described above. The sides of the Arnold tongues are period-q saddle-node bifurcation curves where the two period-q orbits come together and disappear.

Our investigation We investigate a two-parameter family of maps which we call the Lorenz-Kevrekidis map. This family undergoes a Hopf bifurcation, defined by parameter values where a fixed point is neutral, with complex eigenvalues of $e^{\pm 2\pi i\omega}$ on the unit circle. The set of Hopf

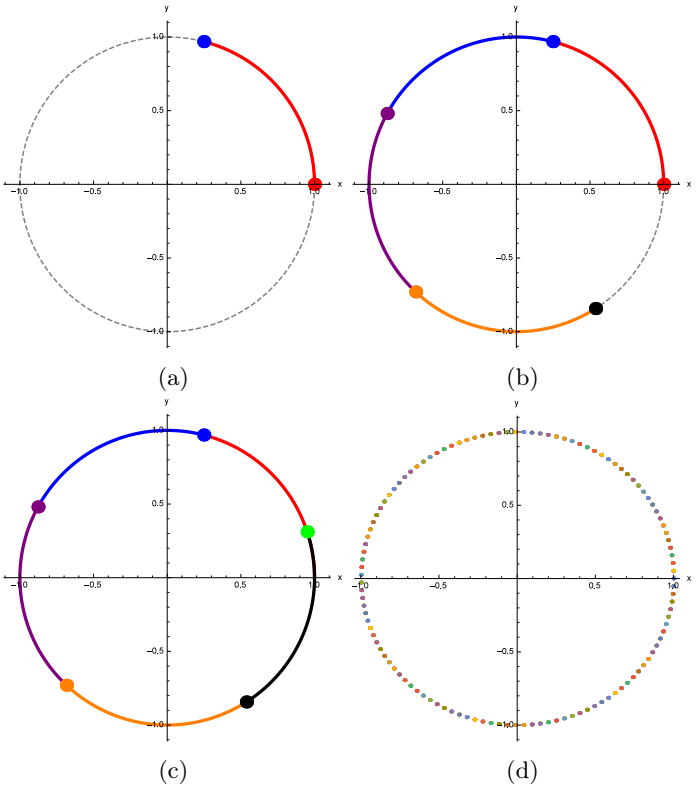


Figure 1: This illustration shows iterates of a point under the circle map with $\theta_0 = 0$ and ω is an irrational number slightly larger than $\frac{1}{5}$. Colored lines show the two-dimensional interpretation of iterating the point of the same color as the line. (a) shows θ_0 and θ_1. (b) shows θ_0, θ_1, θ_2, θ_3, and θ_4. (c) shows θ_1, θ_2, θ_3, θ_4, and θ_5. (d) shows θ_0, θ_1, $\ldots$, θ_{500} and the colored lines are omitted to avoid a mess.

bifurcation points forms a curve in the parameter plane. Along the Hopf curve, ω varies, and the eigenvalues travel along the unit circle. As the Hopf bifurcation curve is crossed from below to above, the fixed point changes from attracting to repelling. This change in stability is accompanied by the birth of an attracting invariant circle for parameter values above the Hopf curve.

The size of the invariant circle increases as the parameters move away from the Hopf curve. Emanating from the Hopf curve, from points with irrational rotations ω, on the side with the invariant circle, are curves of parameter values where the map has an invariant circle with the same irrational number ω. Emanating from the Hopf curve, from points with rational rotation p/q, on the side with the invariant circle, are horn-shaped regions of parameter values where the map has an invariant circle with an attracting and a repelling period-q orbit. The sides of the tongues (or horns) are period-q saddle-node bifurcations, as in the Arnold Circle Map family. Look ahead to Fig. 4, where we see a partial parameter plane bifurcation diagram for the Lorenz-Kevrekidis map that we study in this paper. Displayed are the Hopf bifurcation curve and two p/q tongues, for $p/q = 1/6$, and $1/5$. An "irrational arc" can be seen, many of the figures in this paper will display different approximations to the same irrational arc.

In this paper, we are most interested in the continuation of the irrational rotation arc. Local theory guaranties its existence near the Hopf bifurcation curve, but the invariant circle can, and does, break up when one follows the irrational rotation arc away from the Hopf bifurcation. The mechanism for the breakup involves the intersection of the invariant circle with the critical set of the map. Since the phase space is "folded" along the critical set, the intersection of the invariant circle with the critical set causes "bumps" to form on the invariant circle. The curvature of these bumps increases as the tangent to the invariant circle where it intersects the critical curve approaches the zero eigenvector at that same point. If these two vectors line up, then there is typically a cusp point at the image of this point. Beyond this point, the image of a curve becomes a loop. If this happens, then the invariant circle can no longer exist. The goal of this research is to provide numerical evidence that this special "cusp" point actually exists, and to develop algorithms to numerically locate it.

2 Dynamical Systems Definitions

Definition x is a *period-q* point for the map f, if x satisfies $f^q(x) = x$.

Definition Let $f : R^2 \to R^2$ be a smooth (C^∞) map. Let x be a period-q point for f. The *stable manifold* of x is $W^s(x) = \{y \in R^2 : f^{kq} \to x \text{ as } k \to \infty\}$. Since the inverse map may not be uniquely defined the *unstable manifold* of x is $W^u(x) = \{y \in R^2 : \text{there exists a biinfinite orbit } \{y_j\} \text{ with } y_{i+1} = f(y_i), y_0 = y \text{ and } y_{-kq} \to x \text{ as } k \to \infty\}$.

Definition In this paper the following definition for attractor will be used: A set, Λ, in phase space is called an *attractor* if it has the property that points in a neighborhood, B, of Λ, limit to Λ in iteration. I.e. $\Lambda = \cap_{n=1}^{\infty} f^n(B)$. The set of all points that are attracted to Λ in forward time is called the *basin of attraction of Λ*.

Definition A *rotation number* is the long term average proportion of the way around the IC that the iteration of a point falls from the point.

Definition In mathematics the golden ratio is a very well known irrational number, $\frac{1+\sqrt{5}}{2}$ and has a lot of interesting properties. Later in this paper will be references to *our* "golden ratio", which is simply $\frac{1+\sqrt{5}}{16}$.

Definition The *critical set* of a map, commonly referred to as J_0, is the set on which the determinant of the Jacobian matrix is equal to 0. The image of J_0 will be referred to as J_1 in this paper.

Definition An *invariant* set, S, has the property that $f(S) = S$.

Definition An *invariant circle* is an invariant set that is a topological circle, IC will frequently be used instead of invariant circle in this paper.

Definition A Hopf bifurcation (also called a Neimark-Sacker bifurcation) is a codimension-one bifurcation which occurs as a fixed point changes stability from attracting to repelling with complex eigenvalues passing through the unit circle.

Definition A saddle-node bifurcation is the birth (or death) of a pair of fixed points. At bifurcation, the fixed points coalesce, and there is an eigenvalue of one.

Definition A *period-q Arnold tongue* is a horn-shaped region in a two-parameter space that emanates from a point on the Hopf bifurcation curve where the eigenvalues are a rational multiple of 2π: $2\pi p/q$. Inside this region, the corresponding maps have an invariant circle on which there is at least one period-q orbit. Typically, one orbit is attracting, and the other is a saddle. The invariant circle is composed of the unstable manifold of the period-q saddle points.

Definition A *irrational arc* is a curve in parameter space along which the corresponding maps have an invariant circle with dynamics conjugate to a fixed irrational rotation.

Definition A *cusp point* is a parameter point on an irrational arc where the corresponding map has an invariant circle with cusps. Thus the invariant circle is not smooth.

Definition The *omega limit set* of a point x_0 is the intersection for $n \geq 1$ of the closure of the forward orbit of $f^n(x_0)$.

3 Motivation

In the 2003 paper "A route to computational chaos revisited: noninvertibility and the breakup of an invariant circle" Kevrekidis [1] further explored the work Lorenz did in the 1989 paper "Computational chaos-a prelude to computational instability" [2]. Kevrekidis investigated the transition from a smooth attracting IC to a chaotic attractor in the *noninvertible* setting of what will be referred to as the Lorenz-Kevrekidis map.

The motivation for the Lorenz-Kevrekidis map is quite straightforward. Start with the famous Lorenz system [3]:

$$\begin{aligned}\dot{x} &= -\sigma(x-y)\\ \dot{y} &= -xz+\rho x-y\\ \dot{z} &= xy-\beta z\end{aligned}$$

let $\sigma \to \infty$, and rescale the variables and the remaining parameters. This results in the following two coupled nonlinear ODEs:

$$\begin{aligned}\dot{x} &= ax-xy\\ \dot{y} &= -y+x^2\end{aligned} \tag{2}$$

Once a forward Euler integration scheme with time step τ is directly applied to these differential equations, the result is the equations:

$$\begin{aligned}x_{n+1} &= x_n+\tau(ax_n-x_ny_n)\\ y_{n+1} &= y_n+\tau(-y_n+x_n^2)\end{aligned}$$

When the terms are grouped together, the result is the standard form of the discrete Lorenz-Kevrekidis map:

$$L(a,\tau): R^2 \to R^2 = \begin{cases} x_{n+1} = (1+a\tau)x_n - \tau x_n y_n \\ y_{n+1} = (1-\tau)y_n + \tau x_n^2 \end{cases} \tag{3}$$

where a is the parameter that was fixed in the Lorenz study and τ was the parameter being varied.

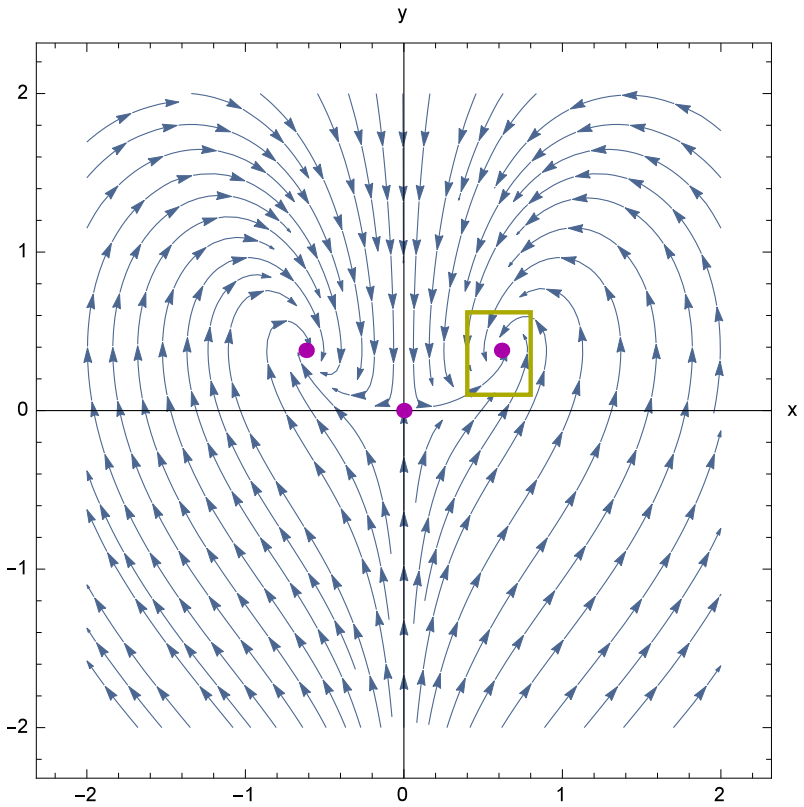


Figure 2: This shows a vector plot of the system defined by equations (2) when $a = .38$. Very simple dynamics in the system and the basins of attraction are clearly the left half plane, y-axis and the right half plane for the magenta fixed points from left to right in the picture. The yellow highlight focuses on the fixed point (.616,.38).

Since the Jacobian matrix for our map is

$$\begin{bmatrix} 1+\tau a-\tau y & -\tau x \\ 2\tau x & 1-\tau \end{bmatrix}$$

finding the determinant is an elementary exercise and it is not hard to believe that when the determinant of this matrix is set equal to 0, the equation can be algebraically manipulated to be $y = -\frac{2\tau}{\tau-1}x^2 + \frac{1+a\tau}{\tau}$. It is evident that the critical set is truly a critical curve, in particular, a parabola. This will be seen frequently in later figures.

When $a = .38$ the dynamics are very simple in the continuous map governed by the system of equations (2). There is a saddle point at the origin with y-axis as stable manifold and curves that branch out, originally parallel to the x-axis and leading into the attracting fixed points in a spiraling motion as unstable manifold. There are two attracting fixed points, (-.616,.38) and (.616,.38) with complex eigenvectors whose basins of attraction are the left and right half plane, respectively. A vector plot for the system is seen in figure 2. At the same value for the parameter a, the dynamics of the system (3) are quite similar as long as τ is "small" which is what would be expected from an Euler integration scheme with "small enough" time step. As τ is increased in size, the first thing that should be noticed is that this is, of course, going to become a very bad approximation to the continuous system and start taking on its own unique properties. Once $\tau = \frac{1}{2a}$ a Hopf bifurcation occurs and an invariant circle (IC) is born. In figure 3 we see two period-5 orbits overlaying a portion of the vector plot seen in figure 2. This figure shows just how bad the Euler approximation can be when the time step, τ is large, for example, $\tau = 1.4$ in the figure and the attractor is no longer a point, but a set of period-5 orbit that lives on an IC, and the original fixed point is repelling, confirming that the system has gone through a Hopf bifurcation.

A general picture of parameter space Given a pair of parameters in parameter space, (a, τ), the dynamics in phase space, (x, y), can vary quite a bit. Figure 4 shows the Hopf curve which

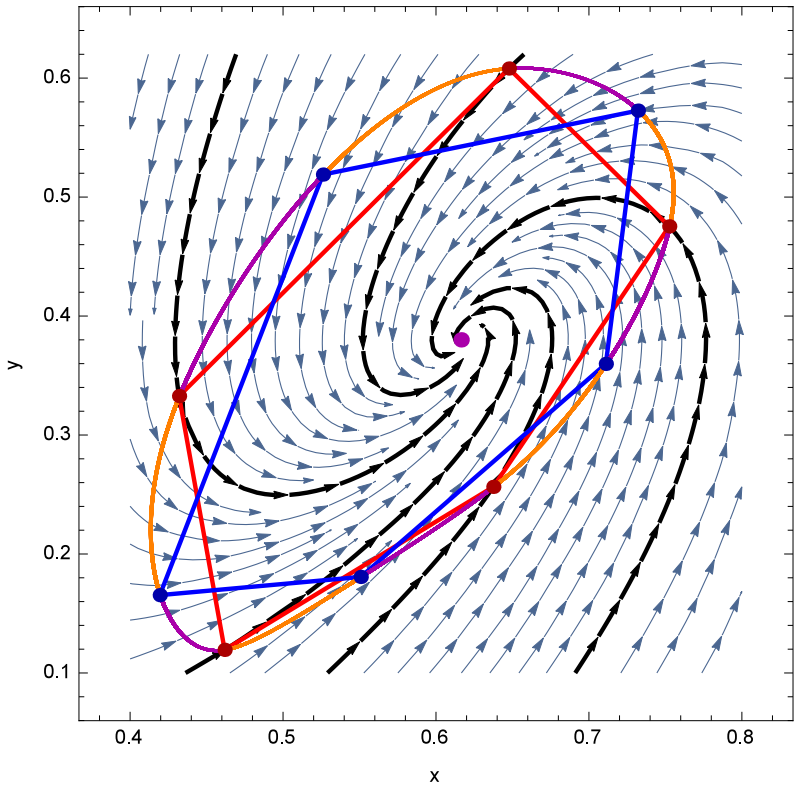


Figure 3: This figure shows only the area highlighted in yellow in figure 2. Overlayed are some features of the discrete map when $\tau = 1.4$. The five black vector streams go through the attracting period-5 points seen in red and connected with red lines to show to show where iteration takes each point (follow counterclockwise). The blue points are period-5 saddle points and are connected with blue lines to show where iteration takes each point (follow counterclockwise). In orange and magenta are the unstable manifolds of the blue points that create the invariant circle.

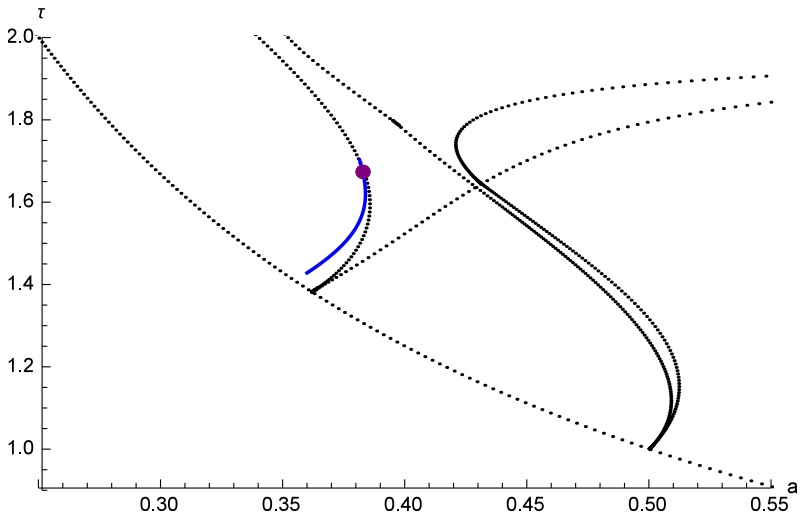


Figure 4: This figure in parameter space shows the Hopf curve that stretches from the upper left corner to the lower right corner. Stemming from the Hopf curve at approximately $(.37, 1.4)$ is the period-5 Arnold tongue. Stemming from the Hopf curve at approximately $(.5, 1.0)$ is the period-6 Arnold tongue. Most of the remainder of the paper focuses on parameter values slightly to the left of the period-5 tongue and above the Hopf curve. Moving along the curve from left to right decreases the rotation number of the IC that is born by crossing from below to above. In blue is the irrational arc that is computed in figure 29 and the purple point, found by extrapolation of the plots in figures 35 and 36 is the best approximation that has been found in this paper for the sought-after 'cusp' point.

is the set of (a, τ) values for which a Hopf bifurcation occurs. Below this curve, the dynamics of the discrete map are extremely similar to the dynamics of the continuous map. Above this curve are ICs in phase space with a rotation number that depends on where it is along the curve; as a increases, the rotation number, ω decreases and as τ increases the IC grows. The Arnold tongues are regions in which a period-q saddle orbit and a period-q node orbit exist. For parameter values sufficiently close to the Hopf bifurcation, these period-q orbits will live on the invariant circle, and the corresponding *necessarily rational* rotation number will be $\frac{p}{q}$. If this is the case, the IC will consist of the period-q saddle points, their unstable manifolds, and the period-q nodes.Between these tongues, arcs of parameter values occur that correspond to an IC in phase space with an irrational rotation number. For parameters on the irrational arcs, the corresponding maps have an IC on which the maps is conjugate to a rigid rotation with a *necessarily irrational* rotation number, ω. That is, it satisfies $f(\gamma(\theta)) = \gamma(\theta + \omega)$, where the image of γ is the IC. In this case, the IC is the ω-limit set of any one point that starts in the basin of the attractor.

Kevrekidis paper issues In the Kevrekidis paper, the authors conducted careful numerical study of the transition from an attracting IC to chaotic dynamics. Lorenz [2] had suggested that there might be a critical value of τ where the invariant circle developed cusps, and beyond which the maps no longer have simple circle map behavior.The authors of the Kevrekidis paper conducted a one-parameter computational study of the Lorenz-Kevrekidis map fixing a at 0.36 and varying τ and found that in a vertical cut, the transition from circle maps to chaotic dynamics was quite complicated. It involved multiple global bifurcations, including the interaction of the IC with periodic orbits (and their unstable manifolds) off the IC. In figure 5 it's not clear that as the parameter cut jumps in and out of Arnold tongues and irrational arcs, much more complicated bifurcation scenario occurs than a single cusp point. The hardest concept to grasp is that although the images in figures 5b and 5c look the same, (c) exhibits chaotic behavior when closely examined and (b) is still an IC. One of the conjectures that was then made in the paper is that if instead of fixing a, an irrational rotation number is fixed and the arc is followed in parameter space, one might be able to find the point in parameter space at which the cusp points occur on the IC becoming the start of a transition to chaos. This is the main approach that was taken in this paper.

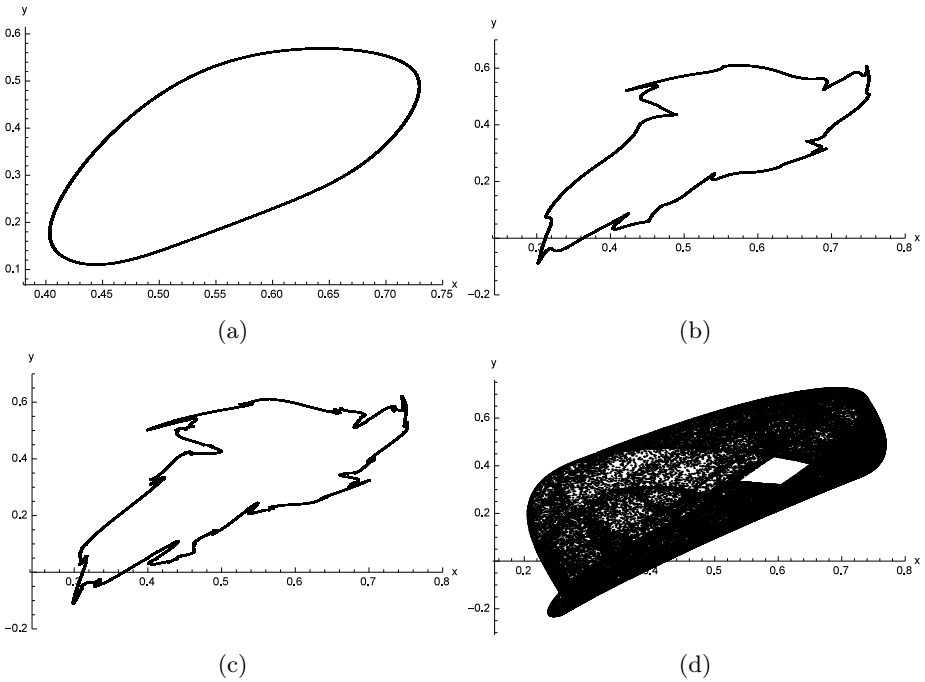


Figure 5: a=.36 for all 4 portraits. (a) τ=1.55 and we see a clear circle as our attractor, (b) τ=1.775 where it isn't as clear but we still have an invariant circle as the attractor, (c) τ=1.785 where we no longer have a circle because there are self intersections in the curve (not clear without zooming in), and (d) τ=1.91 where we clearly have a chaotic attractor.

Why look for cusps? A cusp occurring on the IC in a noninvertible setting is the simplest possible transition from smooth IC to chaotic behavior. The Kevrekidis paper already thoroughly investigated a complicated transition between the two, the purpose here is to see if the simplest possible transition actually does happen. If $\gamma(t)$ is a parametrization in terms of arc length and the IC is smooth, we know that $\gamma'(t) \neq \vec{0}$. Generically (as long as certain nondegeneracy conditions hold), a cusp occurs in the image, $f(\gamma(t))$, if $\frac{d}{dt}[f(\gamma(t))] = \vec{0}$. Since f is in two dimensions, $\frac{d}{dt}[f(\gamma(t))] = D[f(\gamma(t))] * \gamma'(t)$. Since both $D[f(\gamma(t))] \neq \vec{0}$ and $\gamma'(t) \neq \vec{0}$ the only way that the product can be $\vec{0}$ is if $\gamma'(t) \in ker(D[f(\gamma(t))])$. This is the criterion that we look for to determine where the cusp occurs.

4 Numerical results

Using a software entitled TBC (or To Be Continued) [4], the 3 methods (discussed in [5]) we coded for following irrational arcs in parameter space were: linear interpolation, Fourier series, and Fourier series with an anchor point on J_0. The algorithm used is the pseudo arclength method. Typically, one uses known information about the system to find two points on the irrational arc. Once the two initial points are located, the algorithm makes an initial guess for a third point on the arc by moving a predetermined distance along the line connecting the two initial points. Newton's method is then used to converge to this third point, restricting convergence to be perpendicular to the original line. By iterating this process, the software is able to numerically follow the arc in parameter space. The goal of this paper is to find the parameter values where cusps occur on the IC. It is interesting to note that the continuation methods don't necessarily stop working after the IC no longer exists. Each method provided unique challenges and information.

The main observation is that any method should work with enough points or modes but the computation time grows too large to be reasonable as when the number of points or modes is large enough for an accurate approximation of the IC. In section 6 I propose a method the optimizes the amount of time taken versus the accuracy of the approximation.

4.1 Linear Interpolation Method

When using the linear interpolation method, the first thing that needs to be determined is how many points to use to approximate the IC. This number will be referred to as n. The next step is to connect the points with line segments and this serves as the approximation. Theoretically this makes no sense because infinite points would be necessary to replicate the curve but computationally all that is needed is a "small enough" error. Since the desired rotation number and the number of points used to approximate the IC are selected, it is a simple coding task to determine between which two points the iterate of one point should fall so we don't need a general equation, just an equation for each line segment.

Setup To find a first point on the irrational arc, the method needs to start when the system is at parameter values such that there is an IC is fairly close to the shape of an ellipse in phase space. The method starts by selecting points at the center of the ellipse, the major axis and the minor axis. Then an ellipse is parametrized so that those three clicked points are the center, major axis and minor axis. The point (x_0, y_0) is what will be referred to as the base point and is taken to be the selected major axis point. The other $n-1$ points are placed on the parametrized ellipse, equally spaced in θ, where θ is used to represent the proportion around the ellipse in its standard parametrization.

Variables With this method, the variables are pretty straight forward. The first set is the points, $x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}$ in phase space that are connected with the linear interpolation. If we are using n points, this will be $2n$ variables. The remaining variables are the parameter values, a and τ. This gives a total of $2n+2$ variables that the Newton's method needs to converge to. We let $\vec{z}_k = \{x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}, a, \tau\}_k$ be the k^{th} point on the continuation of the arc. We also let $\vec{z}_{k_i} = \{x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}, a, \tau\}_{k_i}$ be the i^{th} step in Newton's method as it approximates $\vec{z}_k$. In other words, $\vec{z}_{k_i} \to \vec{z}_k$ through Newton's method. $\vec{z}_{k+1_0}$ is found by moving a prescribed distance in the direction of the vector $\vec{z}_k - \vec{z}_{k-1}$ from the point $\vec{z}_k$.

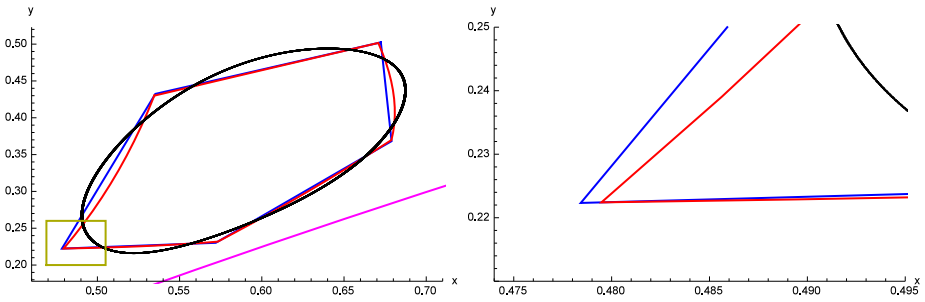
Equations For each of the n mesh points the iterate of the mesh point needs to be on the IC created by linear interpolation. Instead of parameterizing the whole curve, it was easy enough to determine the pair of points that the iterate of a point should fall between and the proportion between those points that it should be. The following equations and specifications are used for Newton's method to converge to the IC:

$$L(x_0, y_0)_x - (\alpha * x_{j \pmod n} + (1-\alpha) * x_{j+1 \pmod n}) = 0 \tag{4}$$
$$L(x_0, y_0)_y - (\alpha * y_{j \pmod n} + (1-\alpha) * y_{j+1 \pmod n}) = 0 \tag{5}$$
$$L(x_1, y_1)_x - (\alpha * x_{1+j \pmod n} + (1-\alpha) * x_{1+j+1 \pmod n}) = 0 \tag{6}$$
$$L(x_1, y_1)_y - (\alpha * y_{1+j \pmod n} + (1-\alpha) * y_{1+j+1 \pmod n}) = 0 \tag{7}$$

$$\vdots$$

$$L(x_{n-1}, y_{n-1})_x - (\alpha * x_{n+j-1 \pmod n} + (1-\alpha) * x_{n+j \pmod n}) = 0 \tag{8}$$
$$L(x_{n-1}, y_{n-1})_y - (\alpha * y_{n+j-1 \pmod n} + (1-\alpha) * y_{n+j \pmod n}) = 0 \tag{9}$$
$$arg(L(x_0, y_0) - (x_0, y_0)) - target = 0 \tag{10}$$
$$(\vec{z}_k - \vec{z}_{k-1}) \cdot (\vec{z}_{k+1_i} - \vec{z}_{k+1_0}) = 0 \tag{11}$$

where $j = \lfloor n * \omega \rfloor \pmod n$, $\alpha = n * \omega - j$, and $target$ is the angle computed as $arg(L(x_0, y_0)_0 - (x_0, y_0)_0)$ in the original approximation of the IC. The last equation ensures that the method converges to a point on the irrational arc that is on the hyperplane perpendicular to $(\vec{z}_k - \vec{z}_{k-1})$ and goes through the point $\vec{z}_{k+1_0}$. This gives a total of $2n+2$ equations. The subscripts x and y refer to the x and y components of the function.



(a) (b)

Figure 6: The initial approximation to the IC using 5 points in the linear interpolation method in phase space. The parameters are (.36, 1.43908). Blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, and magenta is J_1. (a) shows the whole approximation and (b) shows the window that is highlighted in yellow in (a). This image also shows that the iterate of the blue point in the upper left corner in (a) lands in the lower left corner but not on the next blue point, instead it lands on the line that connects that blue point with the next blue point.

4.1.1 5 points

5 points The experiments with the linear interpolation method using a small number of points in the approximation were - not surprisingly - really poor. Figure 6 is evidence of this. It is clear that the points that the method converged to are all outside of the IC. However, the reader should have expected that a pentagon wouldn't be a great approximation for an ellipse so there must be large differences between the actual IC and the approximation if the method does converge to anything. Figure 6b gives a glimpse of how the algorithm introduced by the equations 4-11 work. In 6a it appears that the red points line up with the blue points. It appears this way because the rotation number is so close to $\frac{1}{5}$ and the number of points being used in the approximation is 5 so although the points appear to line up, figure 6b shows that the red points merely lie on the line that connects the blue points, which is exactly what the algorithm is set up to do.

4.1.2 50

50 points The next number of points in an approximation that was investigated was 50. The approximation to the initial ellipse isn't nearly as poor as the 5 point approximation but still isn't great which is evident in 7a on the major axis of the ellipse. However, when compared to the 5 point approximation with similar parameter values (6) there is clear improvement in the approximation. Figures 7b and 7c show that once zoomed in far enough, the approximation that looks pretty good from afar, isn't that great, especially in areas of high curvature. In figure 7d the arc in parameter space has some strange behavior which will be a trend in the linear interpolation method, that strange behavior being a "corner" in the arc. I believe that the actual arc will be smooth, even up to the point in parameter space where cusps occur on the IC in phase space. An obvious thought would be that perhaps the "cusp point" occurs at the corner, however the corner just appears to be where the continuation breaks down and the attracting IC at that parameter value does not have cusps. With the linear interpolation method, the IC still appears quite smooth at the corners in most cases.

4.1.3 150

Two arcs next to each other (150 vs 50) The next number of points considered was 150. In figure 8 the arcs computed using 50 points versus 150 points shown for side by side comparisons. Figure 8a shows both arcs plotted "on top" of each other. It would be expected that the arcs line up with each other until the corner appears on the 50 point arc but this is not the case. The 150 point arc lies below and to the right of the 50 point arc (farther along the Hopf curve with a lower rotation). This phenomenon is a trend with the linear interpolation method and will be discussed in further detail in section 5.

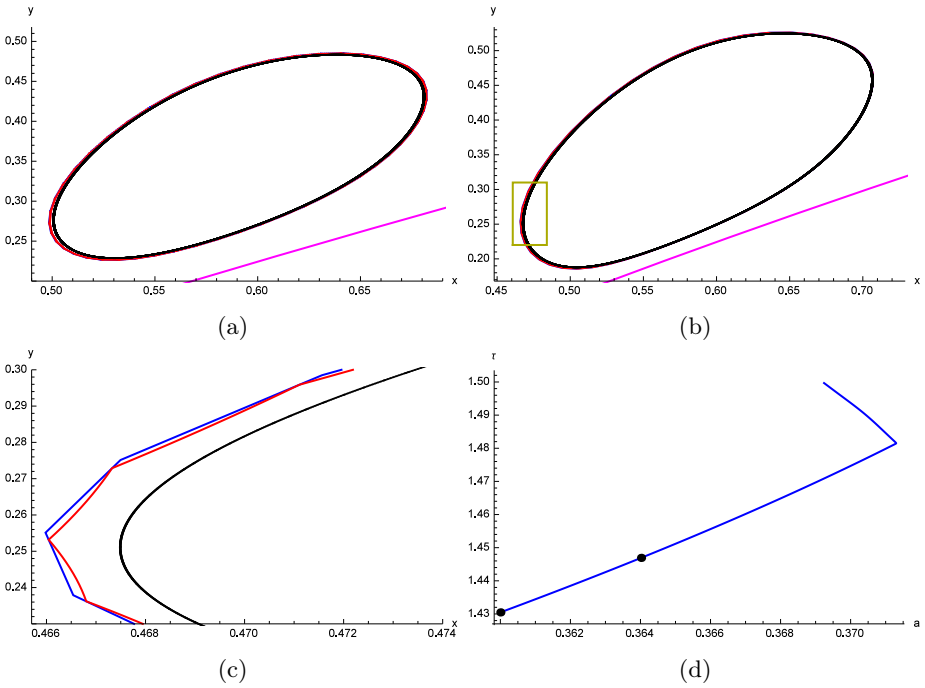


Figure 7: (a), (b), and (c) are 50 point approximations to an IC using the linear interpolation method in phase space. In those images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, and magenta is J_1. In (a) the parameters are (0.36, 1.43047). In (b) the parameters are (0.364017, 1.44692). (c) is a closer look at the yellow window highlighted in (b). (d) shows the arc in parameter space that was generated through continuation of the algorithm and the black dots indicate the points in parameter space that correspond to the phase space images in (a) and (b).

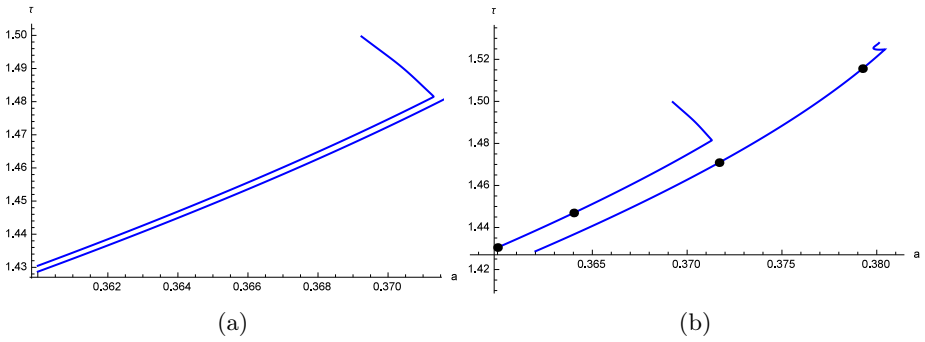


Figure 8: These images show the arc computed using a 50 point approximation to the IC versus the arc computed using 150 points. (a) shows the two arcs plotted together and the 50 point arc appears to be above the 150 point arc in parameter space. (b) shows the two arcs plotted with the 150 point arc shifted to the right for a smoother visual comparison. The black points on the 150 point arc correspond to parameter values that pair with phase space images seen in figures 9a and 9c.

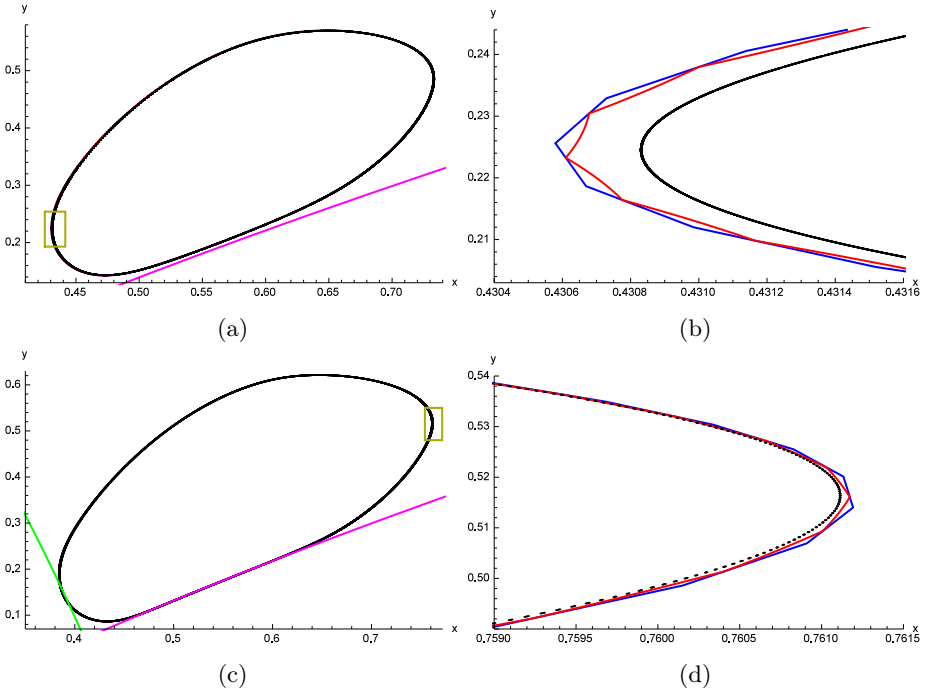


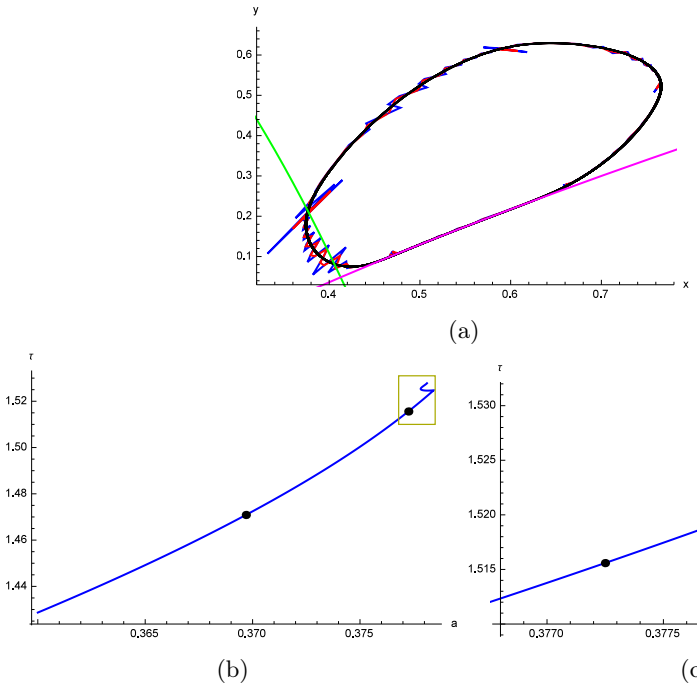
Figure 9: (a), (b), (c), and (d) are 150 point approximations to an IC using the linear interpolation method in phase space. In those images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.369685, 1.47086) which corresponds with the black point in the center of figure 10b. (b) shows the window that is highlighted in yellow in (a). In (c) the parameters are (0.377249, 1.51558) which corresponds with the black point in the upper right corner of figure 10b and the IC appears to have just intersected J_0. (d) shows the window that is highlighted in yellow in (c).

150 points compared to 50 When comparing the figures 7b, 7c, 8b, 9a, and 9b at first glance it appears that although the approximation shown in figures 9a and 9b is much farther along the arc in parameter space, the accuracy isn't as high as in figures 7b and 7c. This, however, is not the case; when looking at the scaling on the axes in figure 9b one can see that the approximation is about 9 times closer to the IC compared to the distance in figure 7c so there is apparent increase in accuracy as the number of points used in the approximation increases.

Continuing the 150 point discussion Figures 9c and 9d show the approximation still holding about the same accuracy farther up on the arc, pretty close to the point where the corner occurs on the arc (see figure 10b). This will be useful when comparing the 150 point approximation with the 300 point approximation. The reader may be wondering about the sudden switch of zoom window location. The window was simply moved to a point on the IC with higher curvature as this appears to be where the most inaccuracy occurs. The most interesting picture so far is figure 10a. This approximation as well as all of the approximations after the corner in the arc seem to exhibit the "zig-zag" behavior that is seen in the image and this tends to lead to an awful approximation of the IC. It isn't quite clear why this happens but this will be discussed further in section 5 in more detail. Figure 10c shows the arc having a highly unexpected hook shape in it after the corner point.

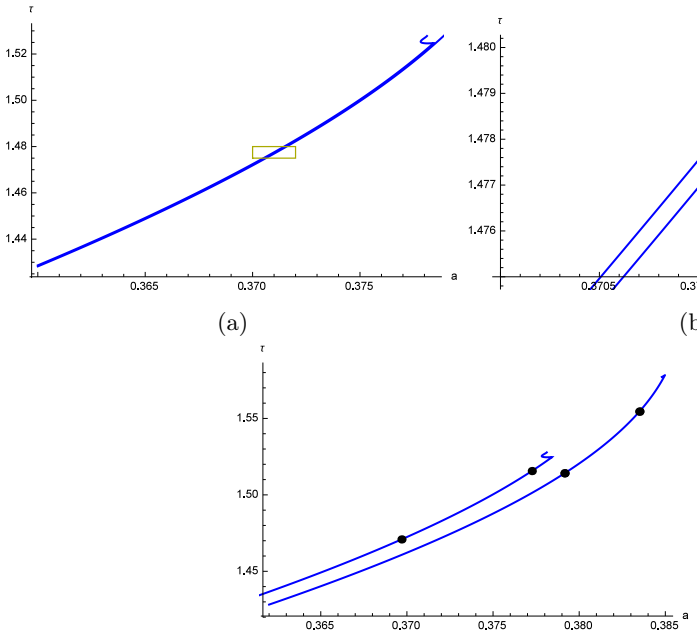
4.1.4 300

Two arcs next to each other (300 vs 150) Figure 11 shows how the arc created by continuation of the algorithm for the 300 point method compares to the arc created by the 300 150 point method. After zooming in, it is clear that there is a gap between the two arcs once again but it is a much smaller gap. The gap is about 10 times smaller than the gap seen in figure 8a which is what would be expected as the approximations get closer to the actual IC.



(a) (b) (c)

Figure 10: (a) is a 150 point approximation to an IC using the linear interpolation method in phase space. In this image blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. The parameters are (0.37812, 1.5283) which corresponds with the red point at the top of (c). (b) shows the same arc as the one on the right side of figure 8b but isn't shifted to the right so that the numbers on the axes are accurate for the arc. (c) shows the window that is highlighted in yellow in (b) and has the added red point that represents the parameters corresponding to the phase image (a).



(a) (b) (c)

Figure 11: These images show the arc computed using a 150 point approximation to the IC versus the arc computed using 300 points. (a) shows the two arcs plotted together. (b) shows the window that is highlighted in yellow in (a) which reveals that the arcs don't line up exactly but are pretty close together. (c) shows the two arcs plotted with the 300 point arc shifted to the right for a smoother visual comparison. The black points on the 300 point arc correspond to parameter values that pair with phase space images seen in figures 12a and 12c.

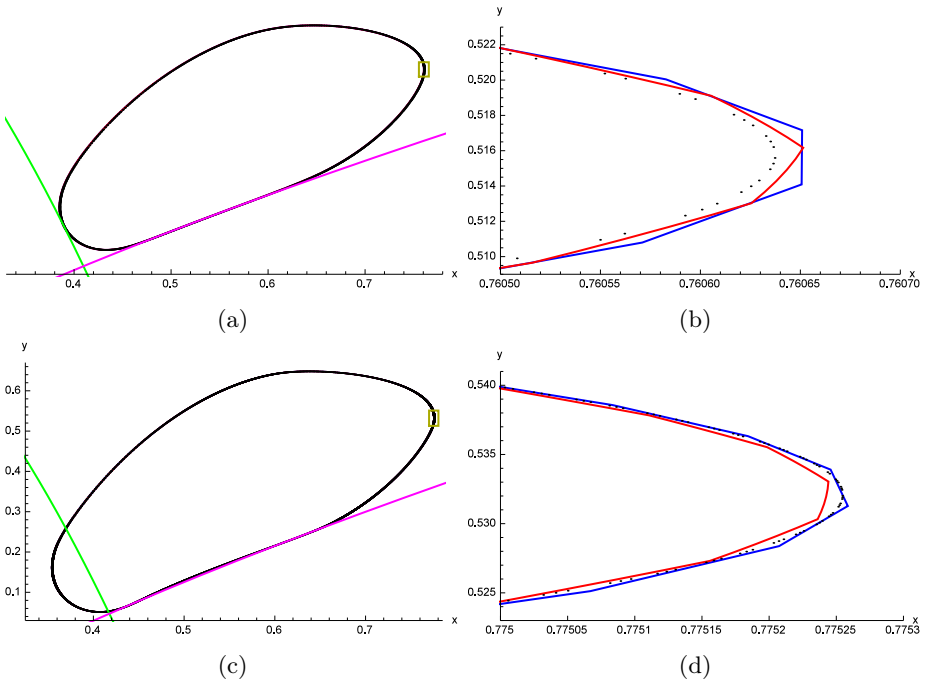


Figure 12: (a), (b), (c), and (d) are 300 point approximations to an IC using the linear interpolation method in phase space. In those images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.377149, 1.51411) which corresponds with the black point in the center of figure 13b and the IC appears to have just intersected J_0. (b) shows the window that is highlighted in yellow in (a). In (c) the parameters are (0.381494, 1.55448) which corresponds with the black point in the upper right corner of figure 13b. (d) shows the window that is highlighted in yellow in (c).

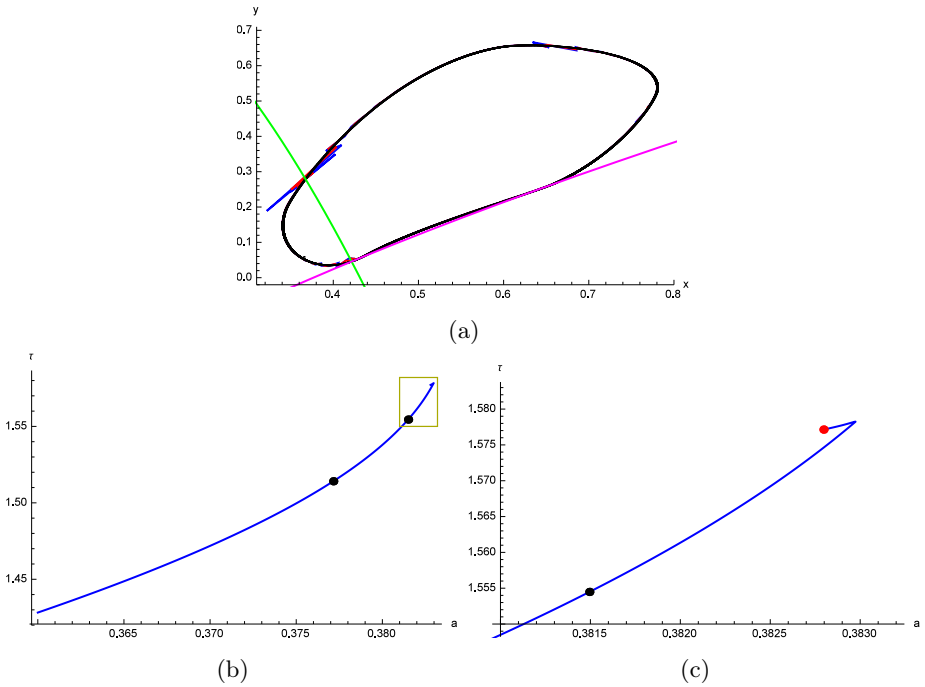
300 points compared to 150 We appear to have an improvement of about 9 times more accuracy when using the 300 point approximation as opposed to the 150 point approximation. This can be seen in the comparison of figures 9d and 12b, which are approximations to the IC at very close parameter values (see figure 11c). This gives more evidence that this method converges to approximating the IC accurately as the number of points used in the approximation increases.

Continuing the 300 point discussion The figures 12c and 12d show the approximation still holding about the same accuracy farther up on the arc, pretty close to the point where the corner occurs on the arc (see figure 13b). This will be useful when comparing the 150 point approximation with the 300 point approximation. The zig-zag behavior is seen in figure 13a with parameters beyond the corner for this arc seen in figure 13c. Since the continuation isn't as far past the corner point as it was in the 150 point examples, the zig-zags don't seem as extreme. Ff this curve were continued as far as the 150 point continuation was, the zig-zags would be just as extreme. Something that can be seen in figures 11c and 8b is that with more points the arc in parameter space seems to continue smoothly further each time before hitting the mysterious corner point.

4.1.5 600

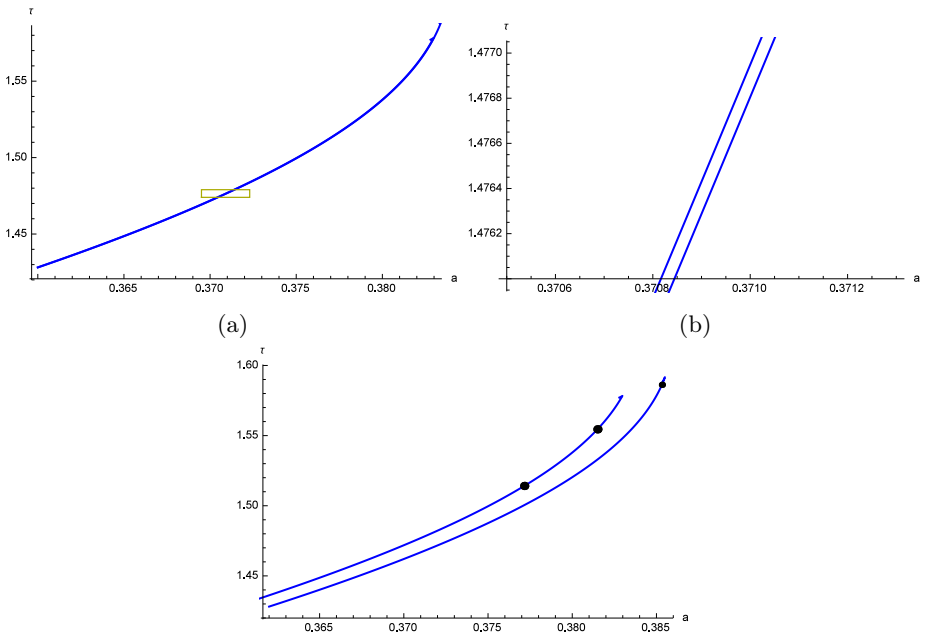
Two arcs next to each other (600 vs 300) Figure 14 shows how the arc created by continuation of the algorithm for the 600 point method compares to the arc created by the 300 point method. After zooming in, there is a small gap between the arcs like the one between the 300 and 150 point arcs. The gap is just a little smaller than the gap seen in figure 11b which is still showing the behavior that is expected.

600 points compared to 300 Figure 15 shows a 600 point approximation to the IC with accuracy about the same as in figure 12c but with parameter values that are even farther on the arc than the parameters values of the IC in figure 13a. This gives even more evidence that as



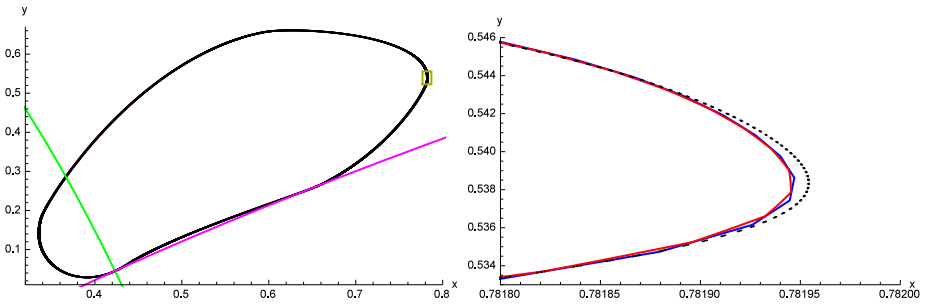
(a) (b) (c)

Figure 13: (a) is a 300 point approximation to an IC using the linear interpolation method in phase space. In this image blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. The parameters are (0.382796, 1.57714) which corresponds with the red point at the top of (c). (b) shows the same arc as the one on the right side of figure 11c but isn't shifted to the right so that the numbers on the axes are accurate for the arc. (c) shows the window that is highlighted in yellow in (b) and has the added red point that represents the parameters corresponding to the phase image (a).



(a) (b) (c)

Figure 14: These images show the arc computed using a 300 point approximation to the IC versus the arc computed using 600 points. (a) shows the two arcs plotted together. (b) shows the window that is highlighted in yellow in (a) which reveals that the arcs don't line up exactly but are pretty close together. (c) shows the two arcs plotted with the 600 point arc shifted to the right for a smoother visual comparison. The black point on the 600 point arc correspond to parameter values that pair with phase space images seen in figure 15a.



(a) (b)

Figure 15: (a) and (b) are 600 point approximations to an IC using the linear interpolation method in phase space. In those images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.383324, 1.58621) which corresponds with the black point in the center of figure 13b. (b) shows the window that is highlighted in yellow in (a).

the number of points used in the approximation is increased the continuation of the algorithm will create a more reliable arc in parameter space with points that correspond to an accurate approximation of the IC farther up than seen in previous attempts with less points.

Continuing the 600 point discussion Figure 15 is showing great improvement so there is hope that with even more points in an approximation there will be even more improvement. The next number of points to be used in the approximation investigated will be 1200 points. Just like in the previous sections, the zig-zag behavior is clear when beyond the corner in the arc. This can be seen in figure 16. Again, figure 14c shows the arc continuing farther than any of the previous arcs had.

4.1.6 1200

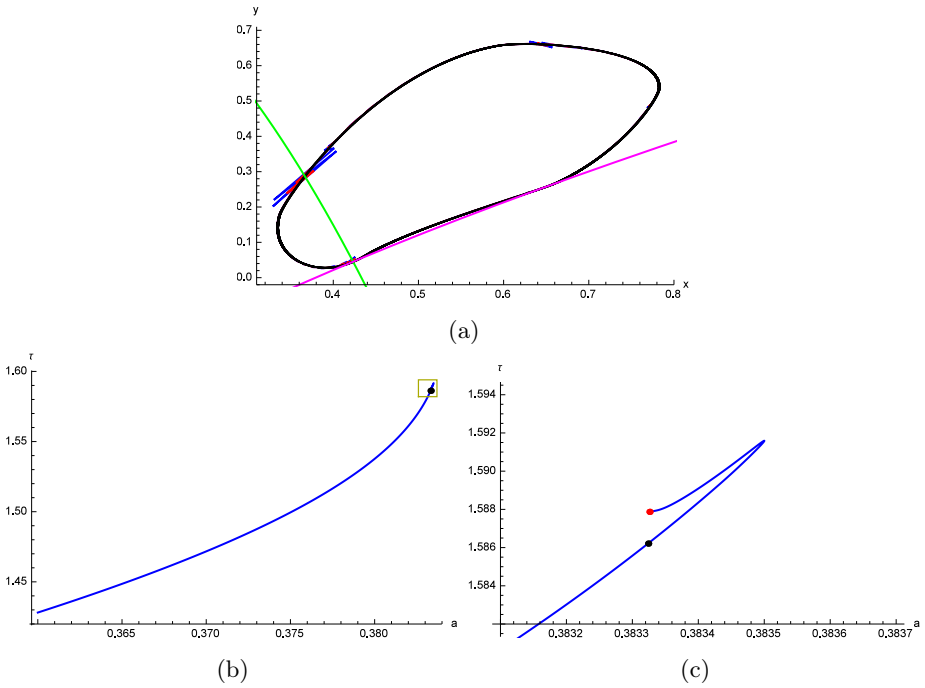
Two arcs next to each other (1200 vs 600) Figure 17 shows how the arc created by continuation of the algorithm for the 1200 point method compares to the arc created by the 600 point method. After zooming in, there is a small gap between the arcs like the one between the 300 and 150 point arcs. The gap is just a little smaller than the gap seen in figure 14b which is still showing the behavior that is expected. It would be reasonable at this point to assume that these are converging to a location in parameter space that actually corresponds to the arc for this rotation number.

1200 points compared to 600 Figure 18 shows a 1200 point approximation to the IC with accuracy about the same as in figure 15a but with parameter values that are even farther on the arc than the parameters values of the IC in figure 16a. The next number of points attempted was 2500. This gave no results at all after running for a month. So we are left with 1200 points being the most used in the linear interpolation method.

Continuing the 1200 point discussion Figure 18 shows that this method is still promising and would be expected to eventually converge but computation time is too large to do experiments with more points. Just like in the previous sections, the zig-zag behavior is clear when beyond the corner in the arc. This can be seen in figure 19. Again, figure 14c shows the arc continuing farther than any of the previous arcs had.

4.1.7 Linear Interpolation Recap

Discussing the pictures Figures 20 and 21 show the progress that the algorithm made when the number of points was increased. This give evidence that as the number of points used approaches infinity, the algorithm is bound to eventually converge to the entire curve. However, as the number of points used increases so does the amount of time that it takes for the algorithm to give output.



(a) (b) (c)

Figure 16: (a) is a 600 point approximation to an IC using the linear interpolation method in phase space. In this image blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. The parameters are (0.383326, 1.58787) which corresponds with the red point at the top of (c). (b) shows the same arc as the one on the right side of figure 14c but isn't shifted to the right so that the numbers on the axes are accurate for the arc. (c) shows the window that is highlighted in yellow in (b) and has the added red point that represents the parameters corresponding to the phase image (a).

Although 2500 points would have been a great approximation to the IC, over a month is just not a realistic amount of time to wait when the Fourier method may do just as well in less time.

4.2 Fourier Series Method

Instead of considering points on the curve, in this method the assumption that the dynamics on the invariant curve are conjugate to rigid rotation of the IC. A truncated Fourier series should be a great approximation for a periodic curve, as long as the curvature is not higher than what is possible by the level of truncation. Another big advantage in this method is that the coefficients are used to give an entire curve, not just n points that are on the curve like linear interpolation.

Variables If our parametric equation that represents the IC in terms of θ is $\gamma(\theta)$, then the x component is $\gamma(\theta)_x$ and the y component is $\gamma(\theta)_y$. Each one of these equations is represented as a truncated Fourier series in this method, so each one has $2n+1$ Fourier coefficients.

The following are the equations:

$$\gamma(\theta)_x = a_{0x} + a_{1x} * \cos(\theta) + b_{1x} * \sin(\theta) + \cdots + a_{nx} * \cos(n*\theta) + b_{nx} * \sin(n*\theta)$$
$$\gamma(\theta)_y = a_{0y} + a_{1y} * \cos(\theta) + b_{1y} * \sin(\theta) + \cdots + a_{ny} * \cos(n*\theta) + b_{ny} * \sin(n*\theta)$$

These Fourier coefficients are what the system will be converging to. Combining the coefficients needed for $\gamma(\theta)_x$ and $\gamma(\theta)_y$ gives us $4n+2$ variables. We also need to converge to values for a and τ in parameter space because the goal is for the IC to have parameter values on the correct curve in parameter space. This gives a total of $4n+4$ variables so the Newton's method will need to be a system of $4n+4$ equations. We let $\vec{z}_k = \{a_{0x}, a_{1x}, \ldots, a_{nx}, b_{1x}, \ldots, b_{nx}, a_{0y}, a_{1y}, \ldots, a_{ny}, b_{1y}, \ldots, b_{ny}, a, \tau\}_k$ be the k^{th} point on the continuation of the arc. We also let

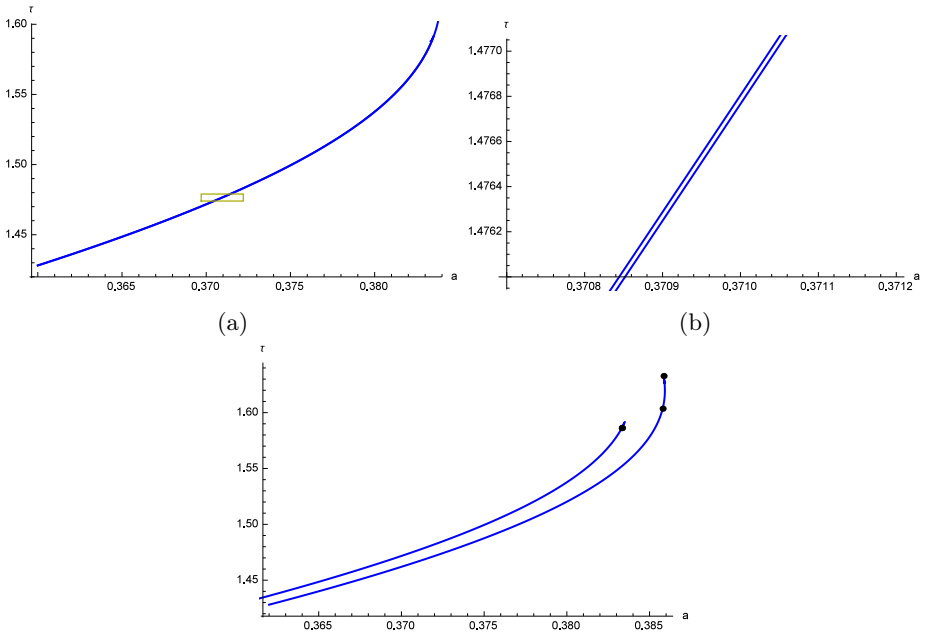


Figure 17: These images show the arc computed using a 600 point approximation to the IC versus the arc computed using 1200 points both using the linear interpolation method. (a) shows the two arcs plotted together. (b) shows the window that is highlighted in yellow in (a) which reveals that the arcs don't line up exactly but are pretty close together (about .00001 apart). (c) shows the two arcs plotted with the 1200 point arc shifted to the right for a smoother visual comparison. The black point on the 600 point arc correspond to parameter values that pair with phase cross parameter space images seen in figure 15a, while the points on the 1200 point arc correspond to the phase space images seen in figure 18.

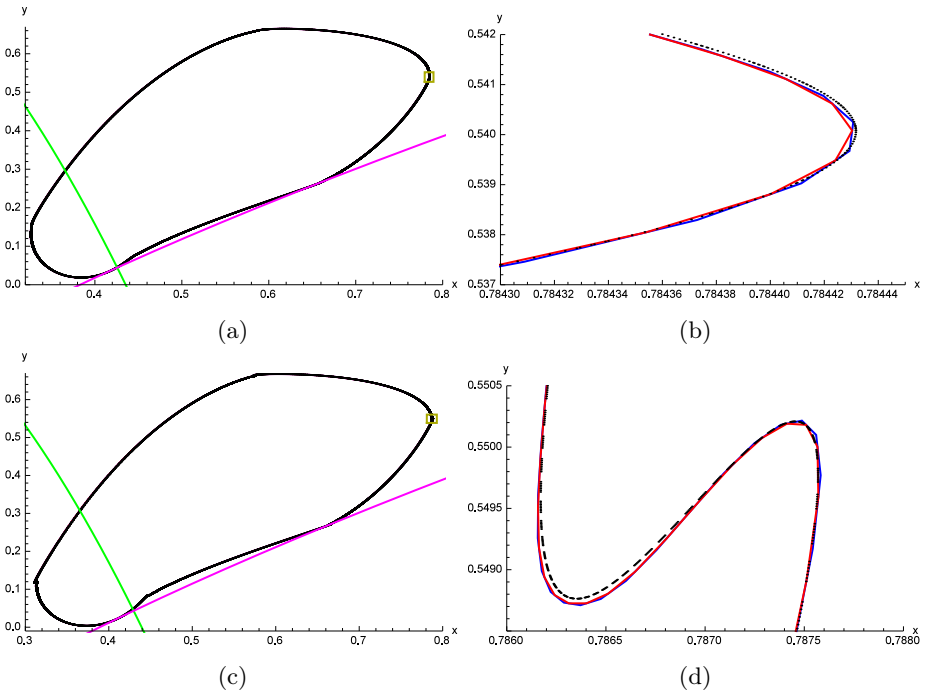


Figure 18: (a), (b), (c), and (d) are 1200 point approximations to an IC using the linear interpolation method in phase cross parameter space. In those images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.38379,1.60351). (b) shows the window that is highlighted in yellow in (a). In (c) the parameters are (0.383848,1.63277). (d) shows the window that is highlighted in yellow in (c).

$\vec{z}_{k_i} = \{a_{0x}, a_{1x}, \ldots, a_{nx}, b_{1x}, \ldots, b_{nx}, a_{0y}, a_{1y}, \ldots, a_{ny}, b_{1y}, \ldots, b_{ny}, a, \tau\}_{k_i}$ be the i^{th} step in Newton's method as it approximates $\vec{z}_k$. In other words, $\vec{z}_{k_i} \to \vec{z}_k$ through Newton's method. $\vec{z}_{k+1_0}$ is found by moving a prescribed distance in the direction of the vector $\vec{z}_k - \vec{z}_{k-1}$ from the point $\vec{z}_k$.

Equations The following equations and specifications are used for Newton's method to converge to the IC:

$$L(\gamma(\theta_0))_x - \gamma(\theta_0 + \omega)_x = 0 \tag{12}$$
$$L(\gamma(\theta_0))_y - \gamma(\theta_0 + \omega)_y = 0 \tag{13}$$
$$L(\gamma(\theta_1))_x - \gamma(\theta_1 + \omega)_x = 0 \tag{14}$$
$$L(\gamma(\theta_1))_y - \gamma(\theta_1 + \omega)_y = 0 \tag{15}$$
$$\vdots$$
$$L(\gamma(\theta_{2n}))_x - \gamma(\theta_{2n} + \omega)_x = 0 \tag{16}$$
$$L(\gamma(\theta_{2n}))_y - \gamma(\theta_{2n} + \omega)_y = 0 \tag{17}$$
$$arg(L(\gamma(\theta_0)) - \gamma(\theta_0)) - target = 0 \tag{18}$$
$$(\vec{z}_k - \vec{z}_{k-1}) \cdot (\vec{z}_{k+1_i} - \vec{z}_{k+1_0}) = 0 \tag{19}$$

where $\theta_i = 2\pi * \frac{i}{2n}$ and $target$ is the angle computed as $arg(L(\gamma(\theta_0))_0 - \gamma(\theta_0)_0)$ in the original approximation of the IC, as described in the linear interpolation method. The last equation ensures that the method converges to a point on the irrational arc that is on the hyperplane perpendicular to $(\vec{z}_k - \vec{z}_{k-1})$ and goes through the point $\vec{z}_{k+1_0}$. This gives a total of $4n + 4$ equations.

Compared to linear interpolation When looking at this method compared to the capabilities of the linear interpolation method, there are pros and cons. This method is definitely going to be

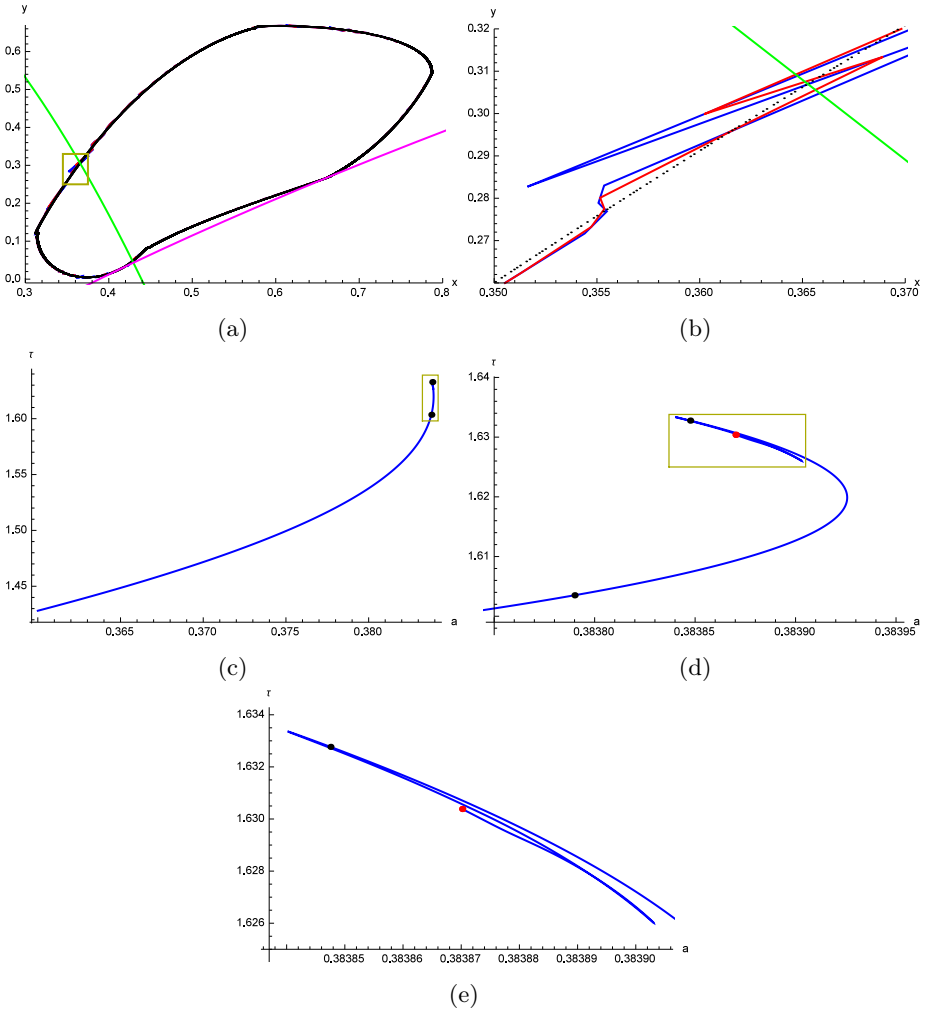


Figure 19: (a) and (b) are 1200 point approximations to an IC using the linear interpolation method in phase cross parameter space. In this image blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. The parameters are (0.38387,1.63038) which corresponds with the red point seen at the end of the curve in (e). (b) shows the window that is highlighted in yellow in (a). (c) shows the same arc as the one on the right side of figure 17c but isn't shifted to the right so that the numbers on the axes are accurate for the arc. (d) shows the window that is highlighted in yellow in (c) and has the added red point that represents the parameters corresponding to the phase image (a). (e) shows the window that is highlighted in yellow in (d) which gives evidence of 2 corners that occur on this arc.

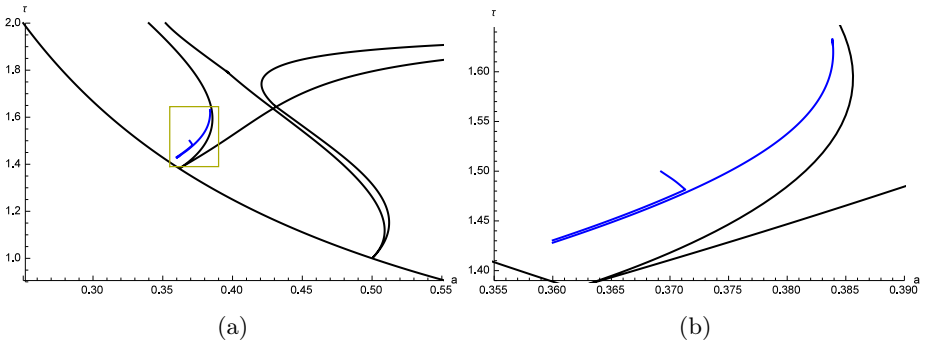


Figure 20: (a) shows the 50 point arc along with the 1200 point arc generated using the linear interpolation method in blue in parameter space along with the hopf curve, period 5 Arnold tongue, and period 6 Arnold tongue in black. (b) shows the region highlighted in yellow in (a).

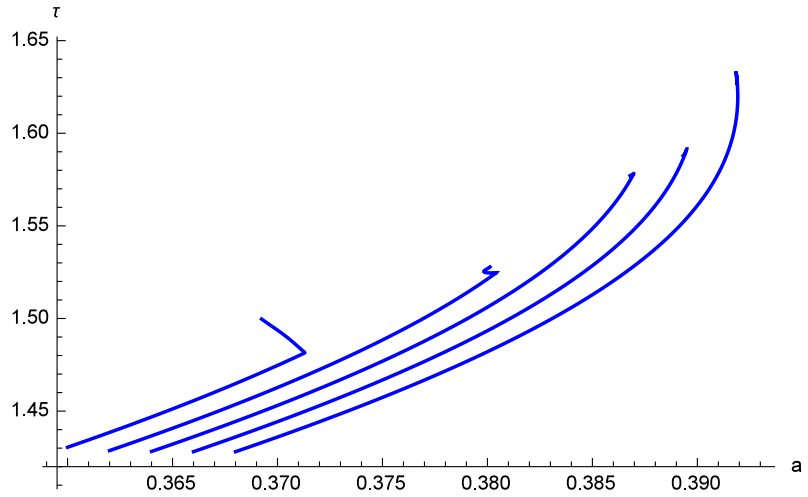


Figure 21: This parameter space figure shows the five arcs that have been generated using the linear interpolation method and discussed up to this point in the paper. From left to right the arcs are 50 points, 150 points, 300 points, 600 points, and 1200 points. The 50 point arc is shown in its true location while the others have been shifted so that they can be compared easily side by side.

better farther away from the cusp point because the only limiting factor here is curvature. So as long as the curvature is reasonable on the IC it doesn't matter how many times the IC curves left versus right. Linear interpolation isn't a great method even when the IC is as basic of a shape as an ellipse, but the Fourier series method can approximate and ellipse accurately with just 1 mode. One of the cons of this method is that each calculation is extremely complex.

4.2.1 5 modes

5 modes A 5 mode Fourier series approximation is definitely going to be better than 5 points when trying to represent the IC. Figure 22a shows that the approximation of the original ellipse is basically perfect. Figure 22b shows that this approximation is still decent at the point where J_0 intersects the IC. In figure 22c, we don't see a corner in the computed arc even beyond the point where the approximation isn't accurate. Figure 22d gives a little information on how the most basic experiments using this method compare to the most advanced experiment of the other method.

4.2.2 48 modes

Two arcs next to each other (48 vs 5) Figure 23 shows how the arc created by continuation of the algorithm for the 48 mode method compares to the arc created by the 5 mode method. The curve on this arc appears to be smoother than on the other one, which makes sense because we are expecting cusps to occur before the arc intersects with an Arnold tongue with such a low period.

48 modes Figure 24 shows us that even this small number of modes computes a much more realistic arc than the one that was computed with the highest amount of points used in the linear interpolation method. This is a sign that this method will be much more useful in future works. Figure 25a shows a lot of improvement, the approximation is much more accurate than the 5 mode approximation was and it is much farther on the arc. Figure 25b appears to be extremely accurate when looking from afar, however as we zoom in, there are clear inaccuracies.

Continuing the 48 mode discussion Figure 26 shows that the arc ventured into the period 5 Arnold tongue, and it appears that the attractor is period 5 and the algorithm still found something that isn't invariant because the blue doesn't match up with the red in figure 26a. Crossing into the Arnold tongue wasn't anything that was seen using the linear interpolation method, but that method didn't produce arcs this long in a realistic amount of time either. The only thing to do is see if this still happens with more modes. One issue may be that the curvature of the arc gets so high at some points that inaccuracy compounds in the continuation algorithm.

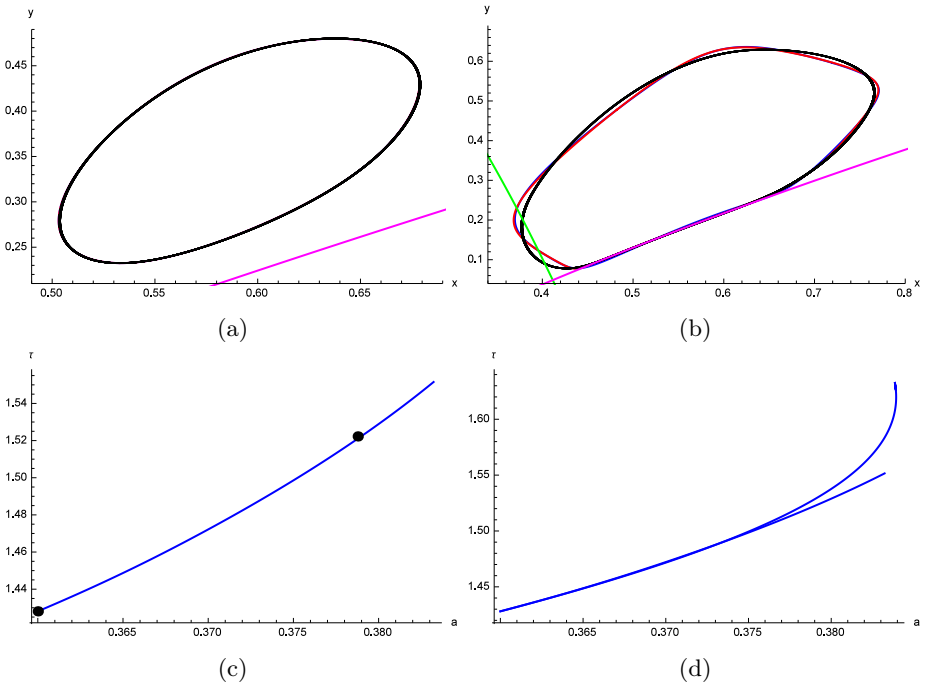


Figure 22: (a) and (b) are 5 mode approximations to an IC using the Fourier series method in phase space. In this image blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.36, 1.42808) which corresponds with the point at the bottom left of (c). In (b) the parameters are (0.378775, 1.52225) which corresponds with the point at the top right of (c). (c) shows the arc in parameter space that was generated through continuation of the algorithm and the black dots indicate the points in parameter space that correspond to the phase space images in (a) and (b). (d) shows the arc in (c) overlaid with the 1200 point arc from the linear interpolation method. The 1200 point arc is the one that reaches the upper right corner.

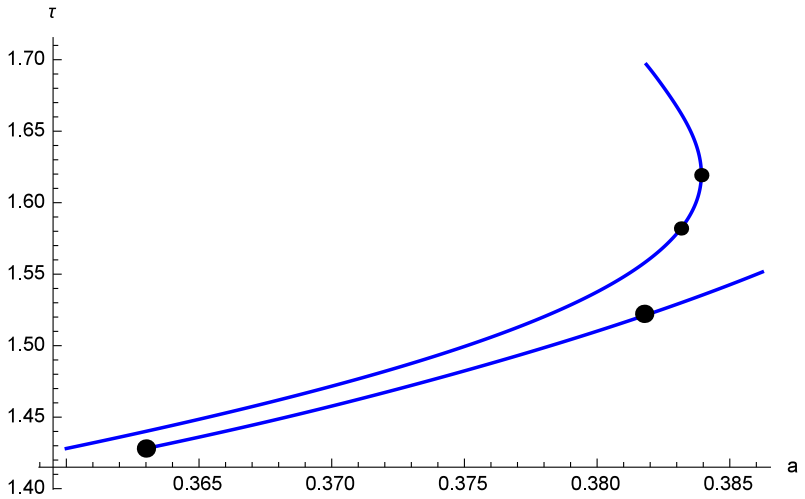


Figure 23: This image in parameter space shows the arc computed using 5 modes and the arc computed using 48 modes. The 5 mode arc is shifted slightly to the right for an easier comparison. The black points on the 48 mode curve correspond to the parameter values for the phase space images in figure 25.

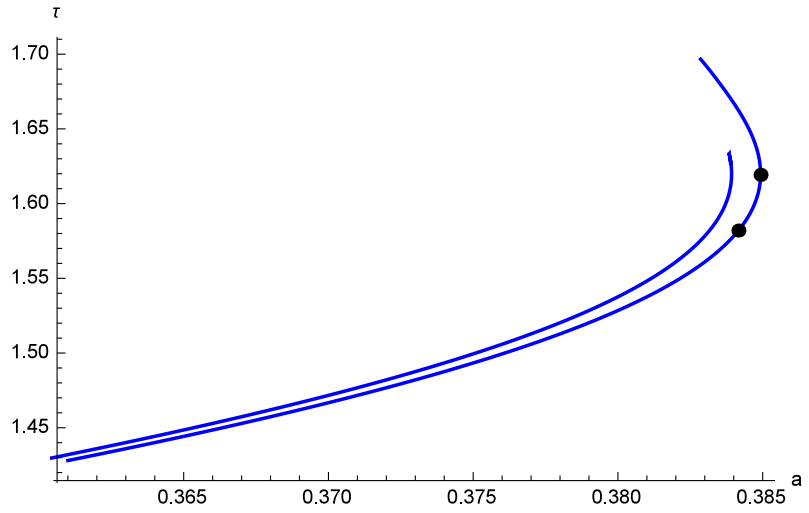


Figure 24: This image in parameter space shows the arc computed using 48 modes and the arc computed using 1200 points with the linear interpolation method. The 48 mode arc is shifted slightly to the right for an easier comparison.

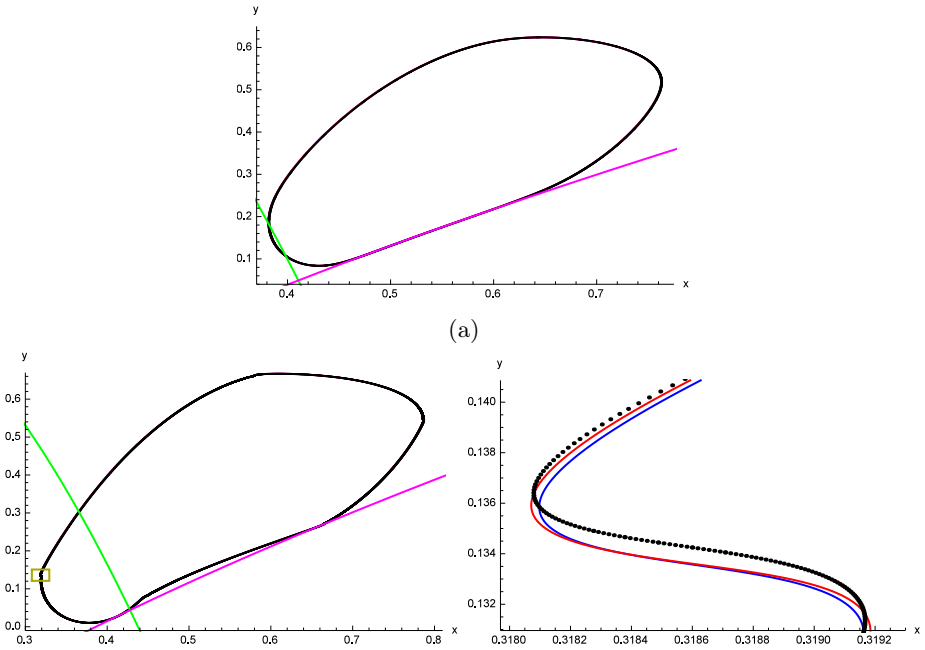
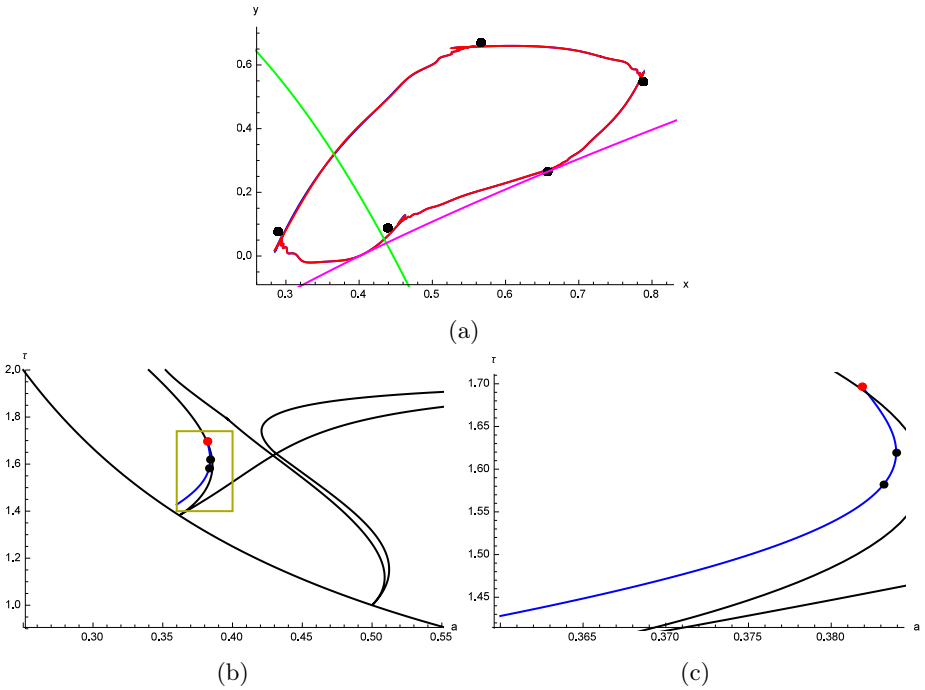


Figure 25: (a), (b), and (c) are 48 mode approximations to an IC using the Fourier series method in phase space. In these images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.37771, 1.51803) which corresponds with the lowest point on the 48 mode curve in figure 23. In (b) the parameters are (0.383927, 1.61923) which corresponds with the highest point on the 48 mode curve in figure 23. (c) shows the area highlighted in yellow in (b).



(a)

(b)

(c)

Figure 26: (a) is a 48 mode approximation to an IC using the Fourier series method in phase space. In this image blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting set found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.381856, 1.69665) which corresponds with the red point at the top of the arc in (b). (b) Shows the 48 mode arc in paramter space along with the hopf curve, period 5 Arnold tongue, and period 6 Arnold tongue. The red point at the top added that represents the parameter values for the IC in (a) and is inside the Arnold tongue. This tells us that a globally attracting invariant circle can no longer exist because it would have to have the irrational rotation number but also contain a period 5 orbit. (c) shows the region highlighted in yellow in (b).

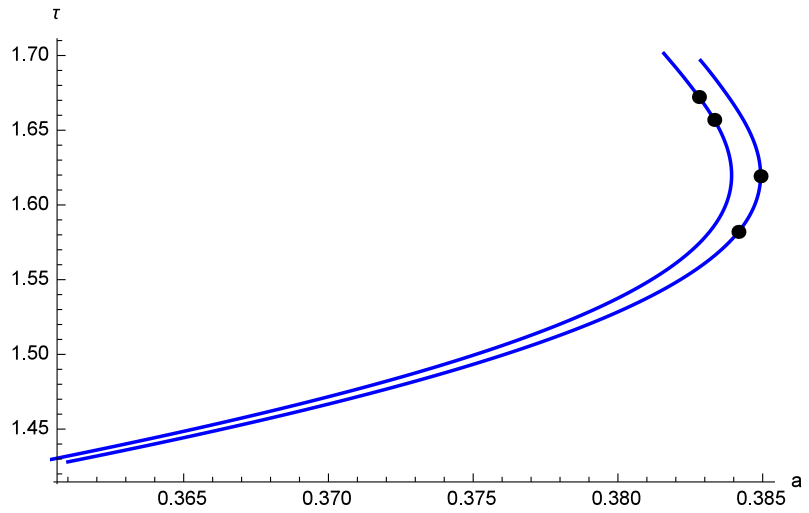


Figure 27: This image in parameter space shows the arc computed using 48 modes and the arc computed using 600 modes. The 48 mode arc is shifted slightly to the right for an easier comparison. The black points on the 600 mode curve correspond to the parameter values for the phase space images in figure 28a and 28c.

4.2.3 600 modes

Two arcs next to each other (600 vs 48) Figure 27 shows how the arc created by continuation of the algorithm with 600 modes compares to the arc created with 48 modes. These curves have the same general shape and are about the same length. It's clear that the black points on the 600 mode curve are farther along than the points on the 48 mode curve.

600 modes Figure 28 shows a lot of improvement, the parameters are much higher on the arc than the 48 mode phase space pictures were. (a) and (b) show great accuracy while (c), (d), and (e) show that the approximation is off by a bit close to the cusp point.

Continuing the 600 mode discussion The figures discussed here might be the most important in the whole paper. Figure 29a shows that continuing farther on the curve resulted in a high periodic attractor (presumably a multiple of 5 but too high to count). The found "IC" appears to display cusps. It is an "IC" because the image in blue is not the same as the IC found in red. The next investigation was to see where a cusp might occur. Figure 29d shows the blue points of the continuation that makes up the curve seen in (c) and the purple point corresponds to the closest parameter values to the cusp point on that curve. Figure 30b shows just how close this point in parameter space is to having an attracting IC with cusps in phase space. Figure 30c shows gives an easy visual for comparing the tangent line on the point of the IC that intersects with J_0 is at an angle very close to the angle of the 0-eigenvector for J_0 which is the criteria for a cusp point to occur where the IC intersects with J_1. The next step to this method will be adding something to the software to compare these angles so that instead of converging to the irrational arc in parameter space, the software converges to the cusp point.

4.2.4 1000 modes

1000 mode discussion With 1000 modes it was only practical to compute the arc because of the computation time needed. It was easy to set the algorithm to go for a long time and let it work for a month. The biggest thing here is that the arc doesn't continue as far as the 600 mode arc, so the initial thought was that the continuation stopped at the cusp point. This, however, was not the case; instead the algorithm stopped short of the cusp point.

4.2.5 Fourier Series Recap

Fourier series seems to far surpass the linear interpolation method in approximating the IC, the only concern is that even with 1000 modes, the approximation doesn't seem to make it to the cusp

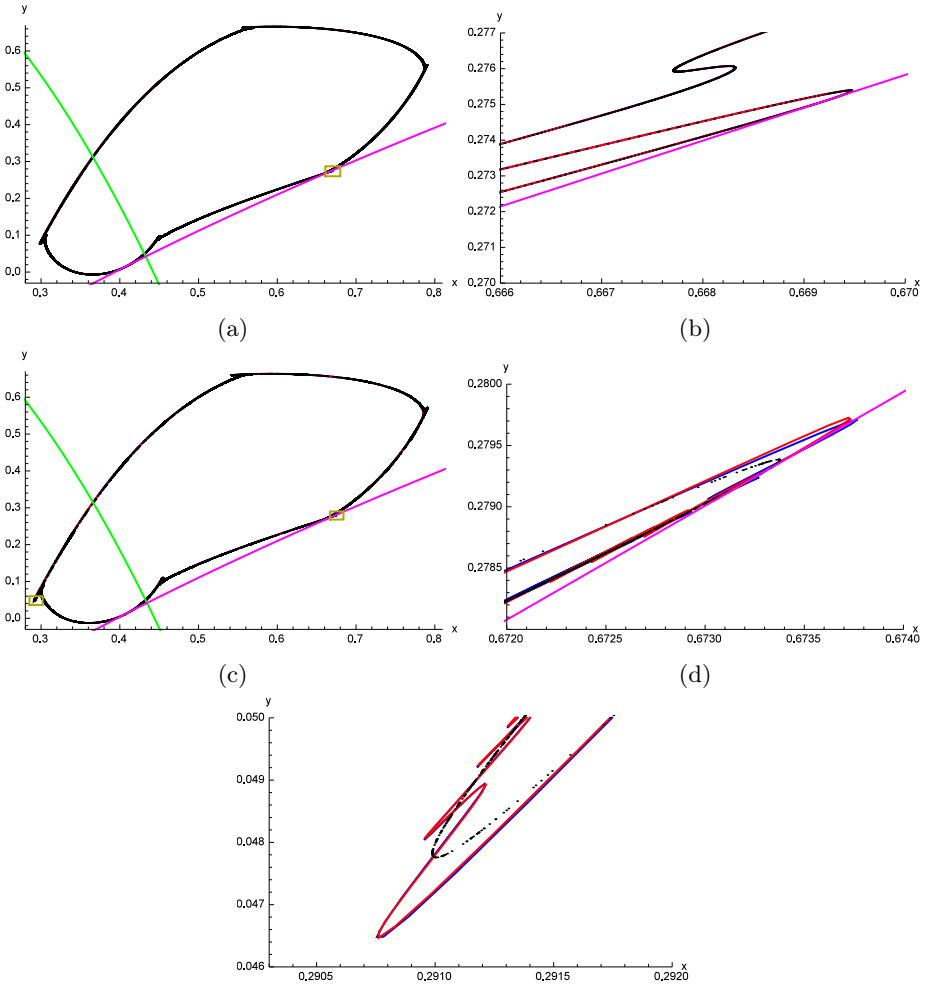
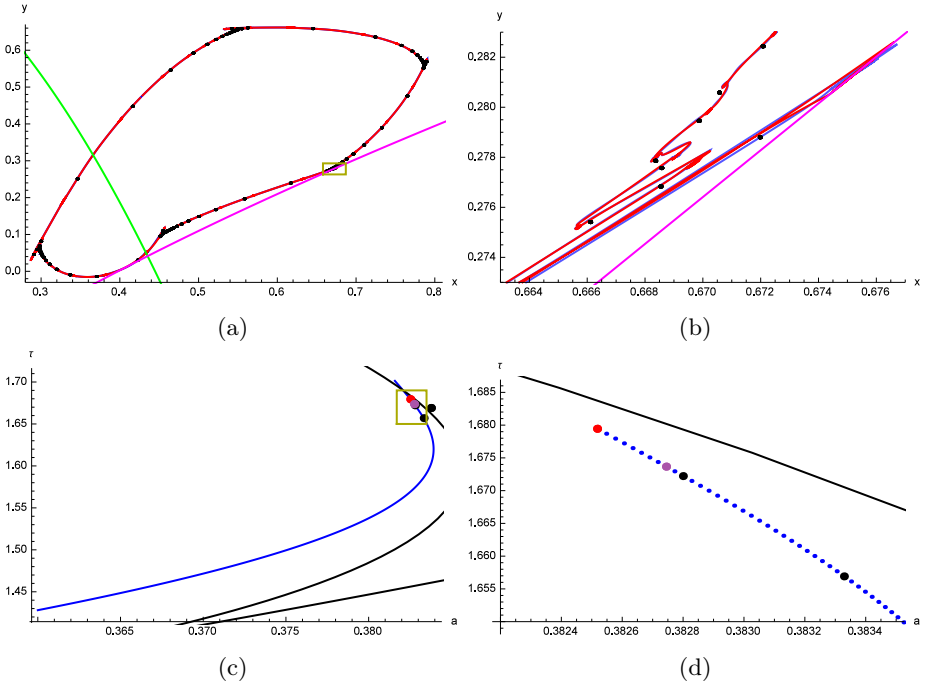


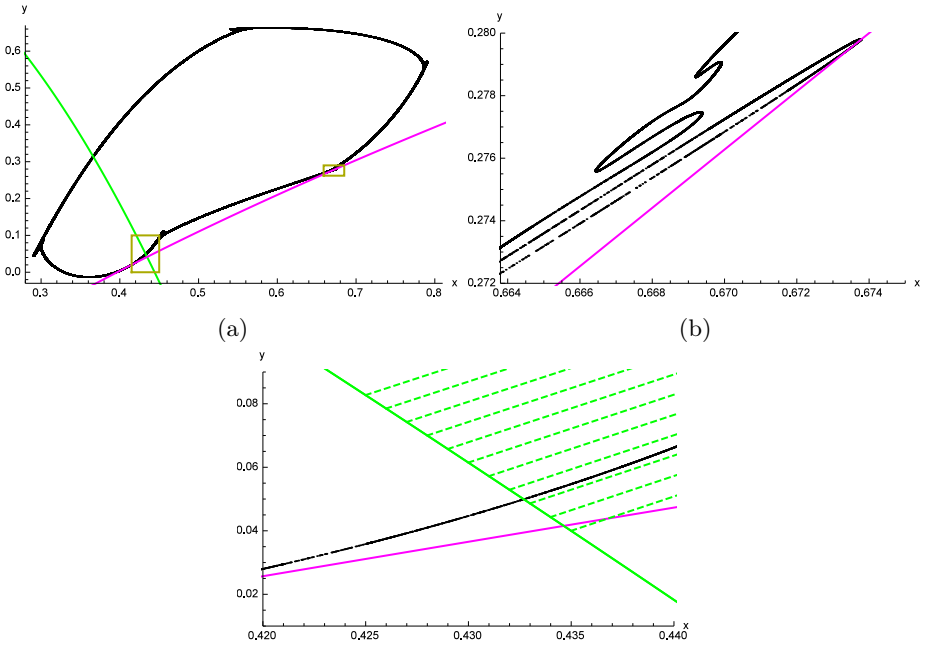
Figure 28: (a), (b), (c), (d), and (e) are 600 mode approximations to an IC using the Fourier series method in phase space. In these images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting set found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.3833284, 1.656912) which corresponds with the lowest point on the 600 mode curve in figure 27. (b) shows the are highlighted in yellow in (a). In (c) the parameters are (0.3827993, 1.672221) which corresponds with the highest point on the 600 mode curve in figure 27. (d) shows the area highlighted in yellow in the middle of the right side of (c). (e) shows the area highlighted in yellow in the lower left corner of (c).



(a) (b)

(c) (d)

Figure 29: (a) and (b) are 600 mode approximations to an IC using the Fourier series method in phase space. In these images blue is the IC approximation, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameters are (0.3825174, 1.679442) which corresponds with the red point at the top of the arc in (c). (b) shows the area highlighed in yellow in (a). (c) Shows the 600 mode arc in paramter space along with the period 5 Arnold tongue with the red point at the top added that represents the parameter values for the IC in (a). A purple point is added that represents the parameter values for the IC in 30a. (d) shows the are highlighed in yellow in (c).



(a) (b)

(c)

Figure 30: In (a) black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1 at the parameter values (0.3827439, 1.673663) which corresponds with the purple point on the arc in figure 29c. (b) shows the area highlighted in yellow in the middle of the right side of (a). The black IC appears to be close to having a cusp at the point where it touches the magenta J_1 curve. (c) shows the area highlighted in yellow ont he left on the bottom of (a). The tangent to the black IC appears to be close to the slope of the green eigenvectors at the point where the IC crosses J_0. Dashed green vectors corresponding to the 0-eigenvector on J_0 are added in (c) for emphasis.

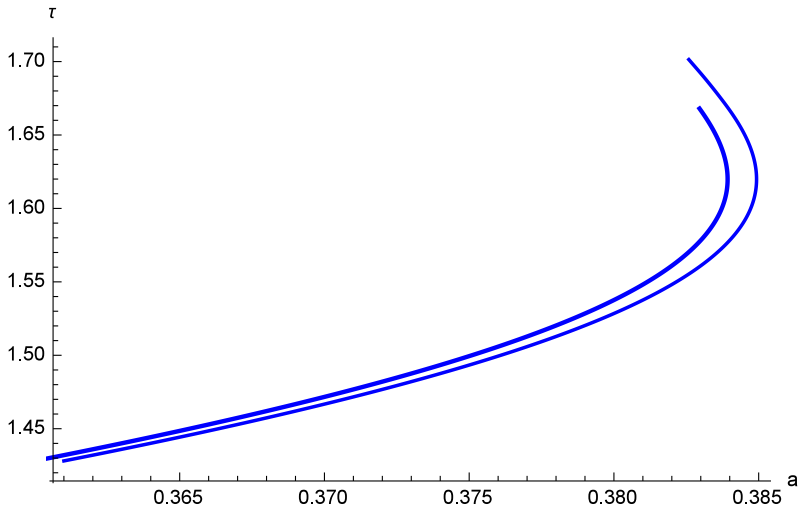


Figure 31: This image in parameter space shows the arc computed using 1000 modes and 600 modes with the Fourier series method. The 600 mode arc is shifted slightly to the right for an easier comparison.

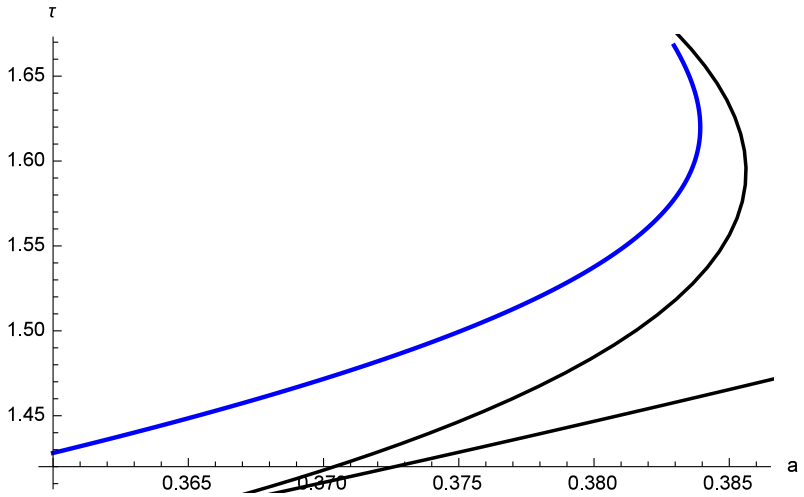


Figure 32: This image in parameter space shows the arc computed using 1000 modes with the Fourier series method and the period 5 Arnold tongue.

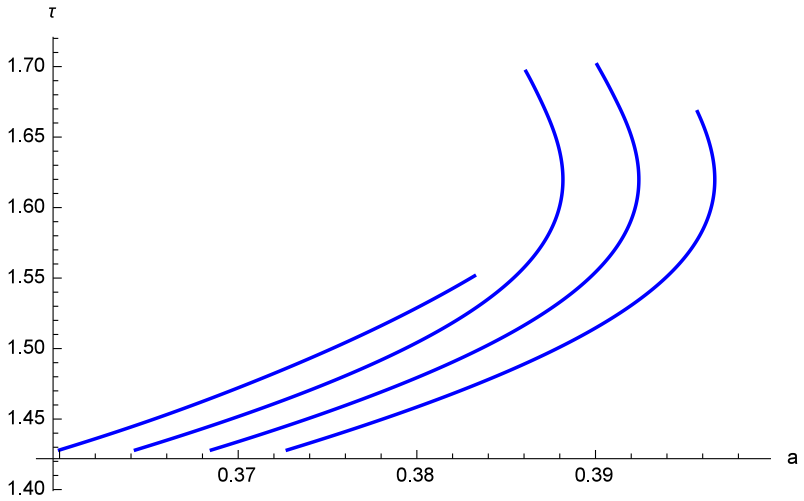


Figure 33: This parameter space figure shows the four arcs that have been generated using the Fourier series method and discussed in this section of the paper. From left to right the arcs are 5 modes, 48 modes, 600 modes, and 1000 modes. The 5 mode arc is shown in its true location while the others have been shifted so that they can be compared easily side by side.

point that appears to exist. If we wanted an algorithm that checked if we were at a cusp point, we have to keep in mind that for this to happen, the tangent vector for the IC on J_0 must be parallel to the 0-eigenvector of J_0 at that point. It is much easier to keep track of these two vectors, if one of the variables that we are storing is the point that is on J_0. This will be the only difference introduced in the next method.

4.3 Fourier Series with anchor point on J_0 Method

The difference between this method and the previous method is the one equation determining the phase condition. Instead of matching the target angle to determine which point is the base point we take the base point to be one of the points on the IC that has Jacobian determinant zero. The added benefit of this method is that the software can now track the angle of the zero-eigenvector and tangent vector at the point where the IC intersects J_0 so that it is possible to add a stopping criterion for when those angles are the same because this is when the cusp point should occur. It isn't an issue with the method, but it should be noted that this method is no better than the previous method for computing the irrational arc, however code can be added to this method to stop following the arc once a cusp point is reached.

Converting Coefficients Changing the location the base point can be done by shifting θ in the equations. The first step of this process is to determine the value for θ at which the IC intersects J_0, call this θ_0. This part is pretty straightforward using the fact that the Jacobian determinant changes signs when crossing J_0. If the old curve was paramtrized by $\gamma(\theta)$, then the new curve is paramtrized by $\gamma(\theta+\theta_0)$. Since the method uses a truncated Fourier series with n modes, the approximation of $\gamma(\theta)$ is as follows:

$$\gamma(\theta)_x = a_{0x} + a_{1x} * \cos(\theta) + b_{1x} * \sin(\theta) + \cdots + a_{nx} * \cos(n*\theta) + b_{nx} * \sin(n*\theta)$$
$$\gamma(\theta)_y = a_{0y} + a_{1y} * \cos(\theta) + b_{1y} * \sin(\theta) + \cdots + a_{ny} * \cos(n*\theta) + b_{ny} * \sin(n*\theta)$$

and when the shift of θ_0 is added the new equations are:

$$\tilde{\gamma}(\theta)_x = a_{0x} + a_{1x} * \cos(\theta+\theta_0) + b_{1x} * \sin(\theta+\theta_0) + \cdots + a_{nx} * \cos(n*(\theta+\theta_0)) + b_{nx} * \sin(n*(\theta+\theta_0))$$
$$\tilde{\gamma}(\theta)_y = a_{0y} + a_{1y} * \cos(\theta+\theta_0) + b_{1y} * \sin(\theta+\theta_0) + \cdots + a_{ny} * \cos(n*(\theta+\theta_0)) + b_{ny} * \sin(n*(\theta+\theta_0)).$$

To keep this in a form that works for the software, these Fourier coefficients need to be changed so that the equations can be written in standard form. Using the basic angle sum trig identities:

$$\cos(i*(\theta+\theta_0)) = \cos(i*\theta) * \cos(i*\theta_0) - \sin(i*\theta) * \sin(i*\theta_0)$$
$$\sin(i*(\theta+\theta_0)) = \sin(i*\theta) * \cos(i*\theta_0) + \cos(i*\theta) * \sin(i*\theta_0)$$

it is a quick calculation to get the new Fourier coefficients:

$$\tilde{a}_{ix} = a_{ix} * \cos(i*\theta_0) + b_{ix} * \sin(i*\theta_0)$$
$$\tilde{b}_{ix} = b_{ix} * \cos(i*\theta_0) - a_{ix} * \sin(i*\theta_0)$$

and now the curve can be written in this form:

$$\tilde{\gamma}(\theta)_x = a_{0x} + \tilde{a}_{1x} * \cos(\theta) + \tilde{b}_{1x} * \sin(\theta) + \cdots + \tilde{a}_{nx} * \cos(n*\theta) + \tilde{b}_{nx} * \sin(n*\theta)$$
$$\tilde{\gamma}(\theta)_y = a_{0y} + \tilde{a}_{1y} * \cos(\theta) + \tilde{b}_{1y} * \sin(\theta) + \cdots + \tilde{a}_{ny} * \cos(n*\theta) + \tilde{b}_{ny} * \sin(n*\theta)$$

with a base point on J_0.

Variables The variables for this method are the same as the variables in the Fourier series method. We let $\vec{z}_k = \{\tilde{a}_{0x}, \tilde{a}_{1x}, \ldots, \tilde{a}_{nx}, \tilde{b}_{1x}, \ldots, \tilde{b}_{nx}, \tilde{a}_{0y}, \tilde{a}_{1y}, \ldots, \tilde{a}_{ny}, \tilde{b}_{1y}, \ldots, \tilde{b}_{ny}, a, \tau\}_k$ be the k^{th} point on the continuation of the arc. We also let $\vec{z}_{k_i} = \{\tilde{a}_{0x}, \tilde{a}_{1x}, \ldots, \tilde{a}_{nx}, \tilde{b}_{1x}, \ldots, \tilde{b}_{nx}, \tilde{a}_{0y}, \tilde{a}_{1y}, \ldots, \tilde{a}_{ny}, \tilde{b}_{1y}, \ldots, \tilde{b}_{ny}, a, \tau\}_{k_i}$ be the i^{th} step in Newton's method as it approximates $\vec{z}_k$. In other words, $\vec{z}_{k_i} \to \vec{z}_k$ through Newton's method. $\vec{z}_{k+1_0}$ is found by moving a prescribed distance in the direction of the vector $\vec{z}_k - \vec{z}_{k-1}$ from the point $\vec{z}_k$.

Equations The following equations and specifications are used for Newton's method to converge to the IC:

$$L(\tilde{\gamma}(\theta_0))_x - \tilde{\gamma}(\theta_0+\omega)_x = 0 \tag{20}$$
$$L(\tilde{\gamma}(\theta_0))_y - \tilde{\gamma}(\theta_0+\omega)_y = 0 \tag{21}$$
$$L(\tilde{\gamma}(\theta_1))_x - \tilde{\gamma}(\theta_1+\omega)_x = 0 \tag{22}$$
$$L(\tilde{\gamma}(\theta_1))_y - \tilde{\gamma}(\theta_1+\omega)_y = 0 \tag{23}$$
$$\vdots$$
$$L(\tilde{\gamma}(\theta_{2n}))_x - \tilde{\gamma}(\theta_{2n}+\omega)_x = 0 \tag{24}$$
$$L(\tilde{\gamma}(\theta_{2n}))_y - \tilde{\gamma}(\theta_{2n}+\omega)_y = 0 \tag{25}$$
$$\tilde{\gamma}(\theta_0)_y + \frac{2\tau}{\tau-1}\tilde{\gamma}(\theta_0)_x^2 - \frac{1+a\tau}{\tau} = 0 \tag{26}$$
$$(\vec{z}_k - \vec{z}_{k-1}) \cdot (\vec{z}_{k+1_i} - \vec{z}_{k+1_0}) = 0 \tag{27}$$

These $4n+4$ equations are the same as the equations 12 except that the second to last equation is now fixing the base point on J_0.

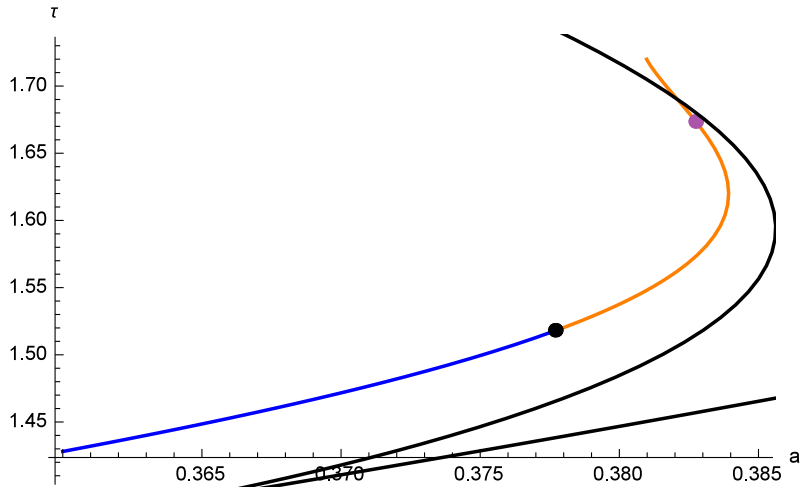


Figure 34: This image in parameter space shows an arc computed using 200 modes with the Fourier series method (blue) until it gets to the point where the IC intersects J_0 (black) and then is continued with the Fourier series with base point on J_0 method using 200 modes (orange). The purple point represents the cusp point that was found on the 600 mode arc in the previous section. The black curves show the period-5 Arnold tongue.

An immediate issue arises that there isn't necessarily a point on the IC that intersects J_0. While this is true, for a cusp point to occur there must be a point on J_0 so the solution to this problem is to start the approximation farther from the Hopf curve. The second issue that comes up is that J_0 intersects the IC in two places when we are close to the cusp point. The advantage here is that in this specific map, the tangent vector on the IC is only close to being parallel to the 0-eigenvector at one of those two points (the one with the lowest y value in phase space).

To make this work, the routine is to follow the arc using the standard Fourier series method until a little bit beyond when J_0 intersects the IC. Then at this point switch to the new Fourier series method and specify where the base point should be. This also requires changing the Fourier coefficients that are being used to parameterize the curve which is explained in more detail in an earlier section. Using this method we were able to better monitor at the angles of the two vectors: the tangent to the IC at the base point and the 0-eigenvector of the base point.

Figures 35 and 36 are the strongest numerical evidence we have to support the claim that a cusp point will occur. Using linear extrapolation with the last 3 points in each figure, we find that the cusp will occur when $a = 0.3827405$ and $\tau = 1.673721$. The distance between this point and the cusp point found in section 4.2.3 is only .000054. The idea of making plots of this type comes from the paper [6]. Note that this is the purple point in figure 4.

5 Discussion

Golden Ratio The "golden ratio", $\frac{1+\sqrt{5}}{16}$. used in this paper was chosen for specific reasons. There has been a lot of work done to show that the golden ratio is the "most irrational" number and some theory in Dynamical Systems gives evidence that the more irrational a number is, the further the corresponding arc in parameter space should go. We took the usual golden ratio and divided it by 8 so that it was closer to $\frac{1}{5}$ and thus closer to the period 5 Arnold tongue. One of this biggest issues with this is that we are using computers to simulate and computers use rational numbers with finite decimal expansions to approximate irrational numbers. An irrational number is by definition *more irrational* than another number if it harder to approximate with rational numbers. So the trade off is taking a number that is harder to simulate on a computer for the hopes that it has the nicer dynamics that are desired in this paper.

Linear interpolation arcs moving to the right In section 4.1 a phenomenon was noticed that as the number of points used to approximate the IC, the irrational arc in parameter space moved

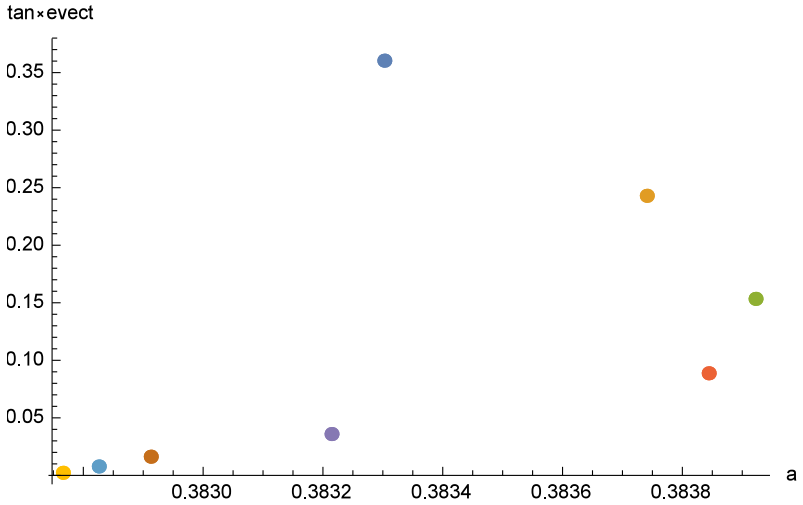


Figure 35: This shows the cross product of the unit tangent vector from the base point on J_0 with the unit 0-eigenvector at the base point on J_0 on the y-axis and a on the x-axis. These values were found using 200 mode approximations to the IC.

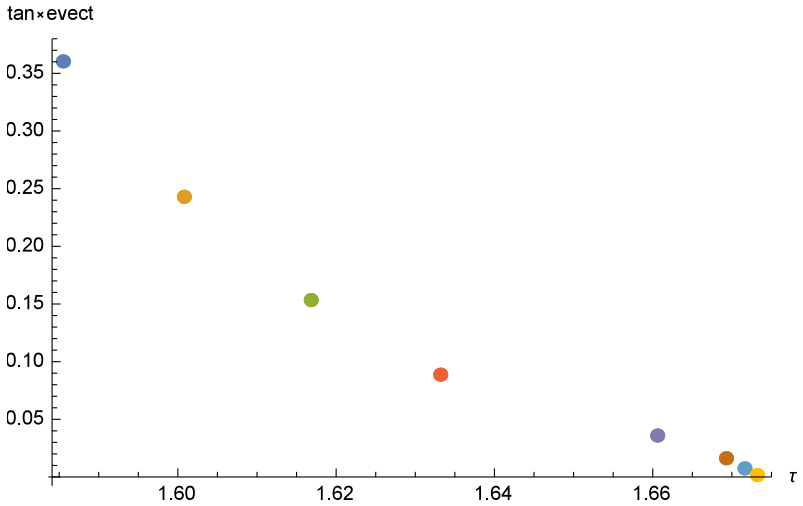
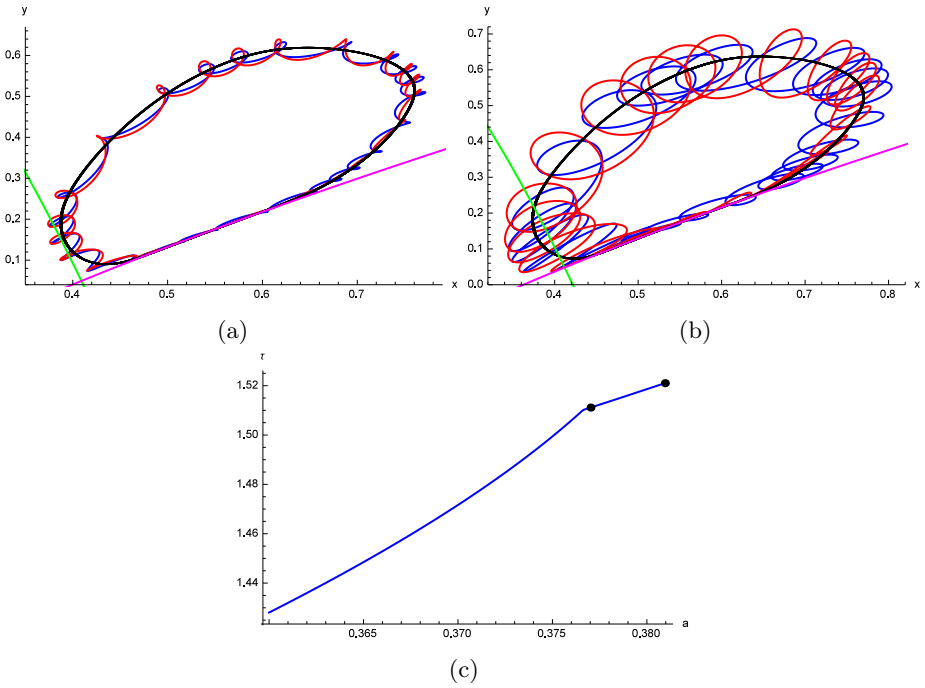


Figure 36: This shows the cross product of the unit tangent vector from the base point on J_0 with the unit 0-eigenvector at the base point on J_0 on the y-axis and τ on the x-axis. These values were found using 200 mode approximations to the IC.



(a) (b)

(c)

Figure 37: In (a) and (b) blue is a 24 mode IC approximation using the Fourier series method, red is the first iterate of the IC approximation, black is the locally attracting IC found by iteration, green is J_0, and magenta is J_1. In (a) the parameter values are (0.377017, 1.51113). In (b) the parameter values are (0.380953, 1.521). (c) shows the irrational arc computed. (a) and (b) correspond the black points seen after the corner in the arc.

to the right - corresponding to having a smaller rotation number. The difference was extremely small when the number of modes was large. The general understanding of this behavior is that the approximations to the IC tends to be larger than the actual IC, because the iterations of the mesh points need to land on the same IC that the mesh points are on. In reality, this bigger IC corresponds to a larger rotation number, this is why it appears farther to the left in parameter space. It would certainly be worthwhile to investigate this further.

Bizarre 24 Fourier mode behavior Several numbers of points or modes were used with each method discussed in this paper, but only a few for each method made it into the paper. One of the most peculiar things that happened during numerical experimentation was loops occurring so early in the approximation when using 24 modes with the Fourier series method. The continuation of the arc in parameter space was considerably shorter than the curve found using as little as 12 modes. Figure 37 gives an example of this odd behavior. Although this is interesting, it isn't the focus of the paper, further work on this would be great.

6 Future Works

Although the objective of this paper is to use numerical experimentation to find a "cusp point" on the irrational arcs in parameter space for which the corresponding map has an attracting invariant topological circle, but with cusps, there is no proof that these points actually exist. Even if some of the experiments show results from simulations that appear to back the hypotheses, the numerical error involved in the work that doesn't allow us to guarantee anything. In the experiments done for this paper it appears that the 600 mode Fourier series curve continuation method did the best job of approaching the cusp point on our "golden ratio" rotation number arc.

One of the next steps in this research is testing these methods with more irrational arcs. The biggest challenge in taking this next step is the massive computation time.

If looking to go even further on this topic, find reasons why the methods in this paper fall apart or converge when we know IC's necessarily aren't there. My personal suggestion to optimize accuracy and time needed is to use the Fourier series method with 250 modes until J_0 intersects the IC and then use the Fourier series with base point on J_0 method from there on at 500 modes. With the current software, computation of each irrational arc should take about a weekend.

References

[1] Christos E. Frouzakis, Ioannis G. Kevrekidis, Bruce B. Peckham *A route to computational chaos revisited: noninvertibility and the breakup of an invariant circle* 2003: Physica D.

[2] Edward N. Lorenz *Computational chaos-a prelude to computational instability* 1989: Physica D.

[3] E.N. Lorenz *Deterministic nonperiodic flow* J Atmos,. Sci. 20 (1963) 130.

[4] Peckham B B [1988-2017], *To Be Continued ...*, a continuation software package for discrete dynamical systems (continually under development).

[5] I. G. Kevrekidis, R. Aris, L. D. Schmidt, S. Pelikan *Numerical computation of invariant circles of maps* 1985: Physica D.

[6] Marta Canadell, Alex Haro *Computation of Quasi-Periodic Normally Hyperbolic Invariant Tori: Algorithms, Numerical Explorations and Mechanisms of Breakdown* 2017: Nonlinear Science.