8-9-2006

Mixing And Transport In The Kelvin-Stuart Cat Eyes Driven Flow Using The Topological Approximation Method

Stephen Michael Rodrigue
University of New Orleans

Mixing And Transport In The Kelvin-Stuart Cat Eyes Driven Flow Using The Topological Approximation Method

A Dissertation

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Engineering and Applied Science

by

Stephen Michael Rodrigue

B.S. Mathematics University of New Orleans, 1982
B.S. Physics (with Honors) University of New Orleans, 1982
M.S. Mathematics University of New Orleans, 1994
M.S. Physics University of New Orleans, 1994

August 2006

Acknowledgements

The research in this dissertation was funded in part by the National Oceanographic and Atmospheric Administration (NOAA) and the United States Geological Survey (USGS).

I would like to thank all my friends for the support they have given me over the years, and especially in the last year. The staff of the Physics department of the University of New Orleans, especially N.B. Day and Sandra Merz, have made the department an interesting and enjoyable workplace. The faculty members of the Physics department have been most supportive over the years, especially Dr. Ashok Puri, who has provided invaluable professional advice on many occasions.

My friends and colleagues in the Physics and Dual Degree Engineering department at Xavier University of Louisiana have shown great support, and have been instrumental in assisting me become a better teacher and physicist.

I am most grateful to my dissertation committee, Dr. Ralph Saxton, Dr. Jeffrey Falzarano, Dr. Juliette Ioup, and my co-chairs Dr. Elia Eschenazi of Xavier University, and Dr. George Ioup, who have been most generous with their time, and have demonstrated great flexibility in adapting to unusual circumstances in the course of this dissertation. I am most especially grateful to Dr. George Ioup, who always manages to find a way to make things work.

It has been a pleasure and a learning experience to work with my dissertation advisor Dr. Elia Eschenazi, who showed great patience, and faith in my abilities, when I struggled through some difficult periods during the course of this research. I have matured as a researcher under his direction, and if I am able to fully develop my potential, it will be due in large part to my association with Elia.

My family is the foundation of my life, and I am eternally grateful for all of their love and guidance.

All of my love and thanks go to Yuan-Jia, who has made so much possible, and enjoyable.

Table of Contents

List of Figures v
Abstract viii
Chapter 1 Introduction 1
Chapter 2 Kelvin-Stuart Cat Eyes Dynamical System 4
2.1 Streamfunctions 4
2.2 Previous Analytical Description 4
2.3 Numerical Studies 6
2.4 Dynamical Systems Analysis 8
Chapter 3 Melnikov Function, Whisker Map, and Lobe Dynamics Transport Theory 15
3.1 Melnikov Function 15
3.2 Whisker Map 18
3.3 Lobe Dynamics Transport Theory 20
Chapter 4 Topological Approximation Method (TAM) 25
4.1 Six Conditions for Applying the Topological Approximation Method 25
4.2 Outline of the Topological Approximation Method 26
4.3 Bifurcation Curves 27
4.4 Symbolic Dynamics, the Lobe Dynamics Transfer Matrix, and the Initial Vector 29
4.5 Lobe Intersection Areas and Weighting Factors 38
Chapter 5 Application of the TAM to the Kelvin-Stuart Cat Eyes Driven Flow 41
5.1 Kelvin-Stuart Cat Eyes Driven Flow Bifurcation Curves 41
5.2 Lobe Intersection Points 44
5.3 Lobe Dynamics Transfer Matrix and Initial Vector for L = 2 50
Chapter 6 Results, Conclusions and Further Research 53
6.1 Transport Rates for L = 2 Through L = 10 53
6.2 Comparison to Theoretical Flux Function 63
6.3 Conclusions 66
6.4 Suggestions for Further Research 67
6.5 Final Thought 68

References....69
Appendices....72
Appendix A: Analytical Calculations....72
A.1 Fixed Point Analysis – Eigenvalues and Eigenvectors....72
A.2 Heteroclinic Orbit (Separatrix), Including Parametric Form....73
A.3 Melnikov Function....75
A.4 Internal and External Periodic Functions....81
Appendix B: Mathematica® Programs....86
B.1 Module for Calculating and Plotting Bifurcation Curves....86
B.2 Topological Approximation Method Module....91
Vita....111

List of Figures

Figure 2.2.1 Streamlines for the Kelvin-Stuart cat eyes dynamical system 5
Figure 2.3.1 Calculated Melnikov amplitude function F(ω) for the Kelvin-Stuart system corresponding to the graph found on p.257 in Ottino (1989) 7
Figure 2.4.1 Heteroclinic Orbit – Separatrix Map 10
Figure 2.4.2 The heteroclinic orbit, and representative internal and external orbits with the associated limits of integration for each case 10
Figure 2.4.3 Internal and external period functions, with the internal function for negative H, and the external function for positive H 12
Figure 2.4.4 Relative error in the approximate period functions 13
Figure 2.4.5 Relative error in the derivatives of the approximate period functions 14
Figure 3.1.1 Perturbed Manifolds and the Distance Between Them 15
Figure 3.1.2 Shifted Melnikov function, with zero at t = 0 17
Figure 3.1.3 Melnikov amplitude function C(ω) 17
Figure 3.2.1 Heteroclinic Orbit and h and t Poincaré Cross-sections for the Whisker Map.....19
Figure 3.2.2 Example of lobe intersection points and their time ordering 19
Figure 3.3.1 Illustrative examples of pips, lobes, turnstile lobes, and regions 21
Figure 3.3.2 The heteroclinic tangle and examples of lobe evolution under iteration of the driving perturbation 22
Figure 3.3.3 Lobe complexity for the Kelvin-Stuart cat eyes driven flow, with representative internal and external lobes.....23
Figure 3.3.4 The first pair of lobes for L = 1, $\varepsilon = 0.1$, and $\omega = 1.15$ 24
Figure 4.4.1 Example of lobe dynamics, using L = 2 for the cubic potential system.....30
Figure 4.4.2 Representative strips for the Kelvin-Stuart driven flow for L = 2 32
Figure 4.4.3 Lobe Dynamics Transfer Matrix for L = 2.....36
Figure 4.4.4 Initial Vector for L = 2 37
Figure 5.1.1 Kelvin-Stuart Cat Eyes Bifurcation Curves for L = 0 to L = 10 41
Figure 5.1.2 Zoom-in on L = 2 Bifurcation Curve 42
Figure 5.1.3 Variation of ω, with L = 2 and ε = 0.01 43

Figure 5.2.1 Intersection Points for Lobes L and L + 1 for L = 2, ε = 0.01, ω = 1.21971 45
Figure 5.2.2 Zoom-in on Lobe L + 1 intersections for L = 2, ε = 0.01, ω = 1.21971 45
Figure 5.2.3 Parametric Plot of Lobes L and L + 1 for L = 2, ε = 0.01, ω = 1.21971 46
Figure 5.2.4 Intersection Points for the inner part of lobe 2L + 1, for L = 2, ε = 0.01, ω = 1.21971 47
Figure 5.2.5 Zoom-in the inner part of Lobe 2L + 1 intersections for L = 2, ε = 0.01, ω = 1.21971 48
Figure 5.2.6 Parametric Plot of Lobes L, L + 1, L + 2, and 2L + 1, for L = 2, ε = 0.01, ω = 1.21971 49
Figure 5.3.1 Calculated Initial Vector for L = 2, ε = 0.01, ω = 1.21971 50
Figure 5.3.2 Calculated Lobe Dynamics Transfer Matrix for L = 2, ε = 0.01, ω = 1.21971 51
Figure 6.1.1 Table of ω values for L = 2 to L = 10 53
Figure 6.1.2 L = 2, ε = 0.01, ω = 1.21971 Area transferred per iteration 54
Figure 6.1.3 L = 2, ε = 0.01, ω = 1.21971 Cumulative area transferred 54
Figure 6.1.4 L = 3, ε = 0.01, ω = 1.55572 Area transferred per iteration 55
Figure 6.1.5 L = 3, ε = 0.01, ω = 1.55572 Cumulative area transferred 55
Figure 6.1.6 L = 4, ε = 0.01, ω = 1.85766 Area transferred per iteration 56
Figure 6.1.7 L = 4, ε = 0.01, ω = 1.85766 Cumulative area transferred 56
Figure 6.1.8 L = 5, ε = 0.01, ω = 2.13393 Area transferred per iteration 57
Figure 6.1.9 L = 5, ε = 0.01, ω = 2.13393 Cumulative area transferred 57
Figure 6.1.10 L = 6, ε = 0.01, ω = 2.39005 Area transferred per iteration 58
Figure 6.1.11 L = 6, ε = 0.01, ω = 2.39005 Cumulative area transferred 58
Figure 6.1.12 L = 7, ε = 0.01, ω = 2.62982 Area transferred per iteration 59
Figure 6.1.13 L = 7, ε = 0.01, ω = 2.62982 Cumulative area transferred 59
Figure 6.1.14 L = 8, ε = 0.01, ω = 2.85599 Area transferred per iteration 60
Figure 6.1.15 L = 8, ε = 0.01, ω = 2.85599 Cumulative area transferred 60
Figure 6.1.16 L = 9, ε = 0.01, ω = 3.07063 Area transferred per iteration 61
Figure 6.1.17 L = 9, ε = 0.01, ω = 3.07063 Cumulative area transferred 61
Figure 6.1.18 L = 10, ε = 0.01, ω = 3.27532 Area transferred per iteration 62

Figure 6.1.19 L = 10, ε = 0.01, ω = 3.27532 Cumulative area transferred 62
Figure 6.2.1 Lobe area as a function of ω 63
Figure 6.2.2 Theoretical flux function for n = 100 iterations 64
Figure 6.2.3 Comparison of TAM calculated values and the theoretical flux function 65
Figure A.3.1 Contour used for contour integration in the complex plane 77
Figure A.3.2 Melnikov Function M (t) vs. ω and t, for A = 0.8, with ε = 1 79
Figure A.3.3 Melnikov Function as a function of t with fixed ω, and ε = 1 79
Figure A.3.4 Simplified Melnikov Function as a function of t with fixed ω, and ε = 1 80
Figure A.3.5 Amplitude function for the simplified Melnikov function 81
Figure A.4.1 Kelvin-Stuart Cat Eyes Dynamical System for A = 0.8, showing foliated internal periodic orbits, and external orbits exhibiting the periodicity of the system 82
Figure A.4.2 The heteroclinic orbit, and representative internal and external orbits with the associated limits of integration for each case 83

Abstract

Transport rates for the Kelvin-Stuart Cat Eyes driven flow are calculated using the lobe transport theory of Rom-Kedar and Wiggins through application of the Topological Approximation Method (TAM) developed by Rom-Kedar. Numerical studies by Ottino (1989) and Tsega, Michaelides, and Eschenazi (2001) of the driven or perturbed flow indicated frequency dependence of the transport. One goal of the present research is to derive an analytical expression for the transport and to study its dependence upon the perturbation frequency ω. The Kelvin-Stuart Cat Eyes dynamical system consists of an infinite string of equivalent vortices exhibiting a 2π spatial periodicity in x with an unperturbed streamfunction of $H(x, y) = \ln(\cosh y + A \cos x) - \ln(1+A)$. The driven flow has perturbation terms of $\varepsilon \sin(\omega t)$ in both the x and y directions. Lobe dynamics transport theory states that transport occurs through the transfer of turnstile lobes, and that transport rates are equal to the area of the lobes transferred. Lobes may intersect, necessitating the calculation and removal of lobe intersection areas. The TAM requires the use of a Melnikov integral function, the zeroes of which locate the lobes, and a Whisker map (Chirikov 1979), which locates lobe intersection points. An analytical expression for the Melnikov integral function is derived for the Kelvin-Stuart Cat Eyes driven flow. Using the derived analytical Melnikov integral function, derived expressions for the periods of internal and external orbits as functions of H, and the Whisker map, the Topological Approximation Method is applied to the Kelvin-Stuart driven flow to calculate transport rates for a range of frequencies from $\omega = 1.21971$ to $\omega = 3.27532$ as the structure index L is varied from L = 2 to L = 10. Transport rates per iteration, and cumulative transport per iteration, are calculated for 100 iterations for both internal and external lobes. The transport rates exhibit strong frequency dependence in the frequency range investigated, decreasing rapidly with increase in frequency.

Chapter 1
Introduction

The mathematical field of nonlinear dynamics experienced rapid development during the last decades of the 20th century, as evidenced by the large number of texts written to introduce and familiarize the scientific, engineering, and mathematical communities with the concepts and techniques of nonlinear dynamics (a sampling of such texts is Abraham and Shaw 1988, Arrowsmith and Place 1990, Guckenheimer and Holmes 1983, Hilborn 1994, Nayfeh and Balachandram 1995, Wiggins 1990, and Wiggins 1992). One area of study where the use of nonlinear techniques has been quite useful has been mixing and transport in fluid dynamics, with nonlinear dynamical methods applied to a wide range of mixing problems in fluid dynamical systems, from microfluidics (Ottino and Wiggins 2004, Wiggins and Ottino 2004), to mesoscale flow (Duan and Wiggins 2004), and to such large scale systems as atmospheric flow (Coulliette and Wiggins 2000) and oceanic flow (Wiggins 2005). Vered Rom-Kedar and Stephen Wiggins developed a theory of mixing and transport, based on what they called lobe dynamics, in a series of papers in the early 1990s (Rom-Kedar, Leonard, and Wiggins 1990, Rom-Kedar and Wiggins 1990, Rom-Kedar and Wiggins 1991, Wiggins 1992). Rom-Kedar further refined this theory into a method she called the Topological Approximation Method or TAM, and applied it to the cubic potential dynamical system (Rom-Kedar 1990, Rom-Kedar 1993, Rom-Kedar 1994, Rom-Kedar 1995). Anna Litvak-Hinenzon, a student of Rom-Kedar, applied the TAM to an asymmetrical Duffing oscillator (Litvak-Hinenzon 1996, Litvak-Hinenzon and Rom-Kedar 1997).

One fluid system of interest is the Kelvin-Stuart Cat Eyes dynamical system, which consists of an infinite sequence of vortices or eddies located at periodic intervals. In the unperturbed system, the interiors of the eddies are separated from the exteriors, with no possibility of transport across the boundary, called the separatrix map or heteroclinic orbit. In the driven or perturbed flow, the boundary undergoes radical change, creating a heteroclinic tangle consisting of an infinite number of lobes, allowing mixing and transport between the interiors and exteriors of the eddies through transfer of the lobes. This transfer of lobes is the central idea of the lobe dynamics transport theory. Indications of such lobe-related mixing and transport in the Kelvin-Stuart driven flow, as well as possible dependence of transport upon

perturbation frequency, have been shown in numerical studies (Ottino 1989 and Tsega, Michaelides, and Eschenazi 2001). The objective of this research is to analytically quantify transport rates of the Kelvin-Stuart Cat Eyes driven flow using the lobe dynamics transport theory through application of the Topological Approximation Method. Frequency dependence of the transport will be examined through calculation of internal and external lobe transport rates for a range of perturbation frequency values, from $\omega = 1.21971$ to $\omega = 3.27532$, spanning values of what is called the structure factor L, as L is varied from L = 2 to L = 10, with the perturbation strength parameter ε constant. Calculation of transport rates for the spatially periodic Kelvin-Stuart system may lead to a general model for transport in periodic systems. Mixing processes might be optimized through choice of the perturbation frequency that maximizes transport, if transport rates can be demonstrated to depend upon the perturbation frequency.

Chapter 2 introduces the streamfunction and the Kelvin-Stuart Cat Eyes dynamical system. Introduction of the streamfunction and its use as a Hamiltonian in section 2.1 is followed by a brief review of the treatment of the Kelvin-Stuart system by Lamb (Lamb 1932) and Stuart (Stuart 1967 and Stuart 1971) in section 2.2. The numerical studies of Ottino (1989) and Tsega, Michaelides, and Eschenazi (2001) are briefly presented in section 2.3. A dynamical systems analytical treatment of the Kelvin-Stuart system follows in section 2.4. Details of the calculations and derivations of the results given in section 2.4 may be found in Appendix A.

Chapter 3 begins with a presentation of the analytical Melnikov function for the Kelvin-Stuart driven flow in section 3.1. The Whisker map for the driven flow is given in section 3.2, followed by a basic introduction to the lobe dynamics transport theory of Rom-Kedar and Wiggins in section 3.3.

Chapter 4 provides a detailed examination of the Topological Approximation Method, beginning with six conditions necessary for application of the method in section 4.1 and an outline of the ten steps of the method in section 4.2. Bifurcation curves are discussed in section 4.3, followed in section 4.4 by a brief development of lobe symbolic dynamics, including general forms for the lobe dynamics transfer matrix and the initial vector. Formulas for calculating lobe intersection areas, and thus weighting values and transport rates, are given in section 4.5.

Chapter 5 applies the steps of the Topological Approximation Method to the Kelvin-Stuart Cat Eyes driven flow. Section 5.1 gives the bifurcation curves for the internal and external lobes for the Kelvin-Stuart driven flow. Section 5.2 presents graphs and discussion

related to the lobe intersection points. Calculated examples of a lobe dynamics transfer matrix and an initial vector constitute section 5.3.

The results of applying the TAM to the Kelvin-Stuart system are given in Chapter 6. Graphs of transport at each iteration, and for the cumulative transport at each iteration are given for the cases of L = 2 through L = 10 in section 6.1. A theoretical flux function, proposed by Rom-Kedar and Poje (1999), is introduced, and the results of the application of the TAM to the Kelvin-Stuart driven flow are compared to the theoretical flux function in section 6.2. Several conclusions are discussed in section 6.3, followed by suggestions for further research in section 6.4, and a final thought is given in section 6.5.

Appendix A provides details of the analytical calculations for the Kelvin-Stuart Cat Eyes dynamical system. Section A.1 presents the fixed point analysis leading to the system eigenvalues and eigenvectors. Section A.2 outlines the procedure for finding the parametric form for the heteroclinic orbit (separatrix). Details of the calculation of the analytical form of the Melnikov integral, including contour integration, are given in section A.3. Finally, the internal and external periodic functions, including approximations and approximate inverse functions for each, are derived in section A.4

Appendix B gives two complete Mathematica® programs applying the Topological Approximation Method. Section B.1 provides a self-contained module for generating bifurcation curves, both approximate and exact. The individual programming cell of Section B.2 generates the lobe intersection points, calculates the respective areas, calculates the weighting factors used in the lobe dynamics transfer matrix, creates the lobe dynamics transfer matrix and the initial lobe transport vector, calculates the area transported per iteration, and the cumulative area transferred per iteration, and generates plots for both transport areas. The program in Section B.2 is not a module, but is a self-contained individual cell that performs all of the above listed operations in one execution. Initial values need to be changed for each L value that is used. Both the module in Section B.1 and the program cell in Section B.2 can be used for other dynamical systems by redefining the Melnikov and period functions listed at the beginning each.

Chapter 2
The Kelvin-Stuart Cat Eyes Dynamical System

2.1 Streamfunctions

Streamfunctions are a fundamental concept in the study of fluid dynamics (Lamb 1932, Stuart 1967, 1971, Aref 1984, Ottino 1989, Hilborn 1994). The two-dimensional streamfunction $\Psi(x, y, t)$ is defined as the flux of a two-dimensional fluid as it crosses a boundary, and has units of (area)/(time), e.g., m^2/s (Lamb 1932). A flow must be incompressible and inviscid for a streamfunction to be integrable (Aref 1984, Ottino 1989). If the streamfunction is autonomous, or time-independent, then fluid particles move along curves of constant $\Psi(x, y)$ called streamlines. Velocity components can be defined with a number of sign conventions, but the one used here will be $\dot{x} = \frac{\partial \Psi}{\partial y}$, and $\dot{y} = -\frac{\partial \Psi}{\partial x}$, the same mathematical form as the equations of motion in Hamiltonian dynamics, thus allowing the streamfunction to be used as the Hamiltonian function in dynamical systems analysis (Aref 1984, Ottino 1989, Hilborn 1994, Tsega, Michaelides, and Eschenazi 2001). The autonomous, unperturbed streamfunction will henceforth be denoted by $H(x, y)$ rather than $\Psi(x, y)$, and will be called the Hamiltonian rather than the streamfunction.

2.2 The Kelvin-Stuart Cat Eyes Dynamical System

The Hamiltonian for the nondimensional, unperturbed Kelvin-Stuart cat eyes dynamical system is (Tsega, Michaelides, and Eschenazi 2001)

$$H(x, y) = \ln(\cosh y + A\cos x) - \ln(1 + A) \qquad \text{Eq.2.2.1}$$

For the system to remain real for all values of x and y, $|A| < 1$. The significance of the subtracted $\ln(1+A)$ term is that this now allows values of $H < 0$ for the interior of the cat eyes, which represent vortices, and values of $H > 0$ for the exterior of the cat eyes, with $H = 0$ on the curve separating the interiors and exteriors (this curve is known as the separatrix, or heteroclinic orbit). The entire system demonstrates a 2π spatial periodicity in x. The streamlines for the unperturbed Kelvin-Stuart cat eyes dynamical system are shown on the next page in Figure 2.2.1.

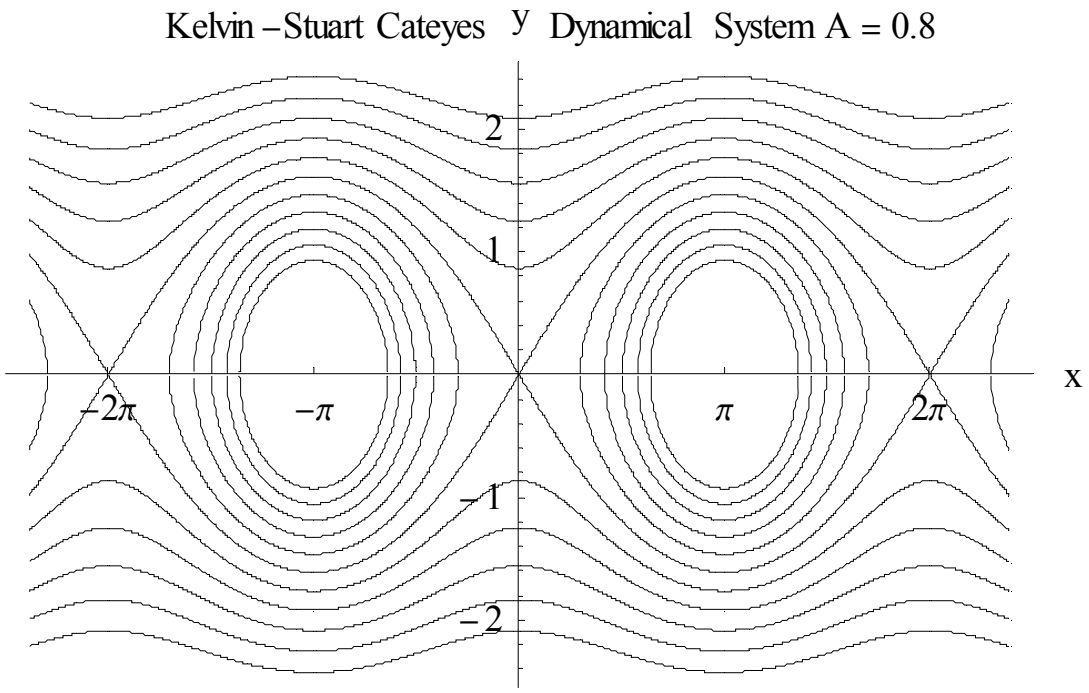

Figure 2.2.1 Streamlines for the Kelvin-Stuart cat eyes dynamical system

Rotation cells, or vortices, are observed to be centered about the points ((2n+1) π, 0). The flow external to these vortices is to the right above the vortices, and to the left below the vortices, making the rotation of the vortices clockwise. The vortices are confined within the curves, called separatrices or heteroclinic orbits, which extend between the points (2n π, 0). The equations of motion for the system, i.e., the velocity components, are given by:

$$\dot{x} = \frac{\partial H}{\partial y} = \frac{\sinh y}{\cosh y + A\cos x} \qquad \text{Eq.2.2.2.a}$$

$$\dot{y} = -\frac{\partial H}{\partial x} = \frac{A\sin x}{\cosh y + A\cos x} \qquad \text{Eq.2.2.2.b}$$

In his classic text on hydrodynamics Lamb (1932) describes on p.224-5 a system "consisting of an infinite row of equidistant vortices, each of strength *K*, whose coordinates are (0, 0), (± *a,* 0), (± 2*a,* 0), …". This is essentially the same system as shown above, but translated by ± π in the *x*-direction, and scaled so that the vortices are centered at the coordinates given by Lamb.

The velocities calculated by Lamb are:

$$u(=\dot{x}) = -\frac{\mathrm{K}}{2a}\frac{\sinh(2\pi y/a)}{\cosh(2\pi y/a)-\cos(2\pi x/a)} \qquad \text{Eq.2.2.3a}$$

$$v(=\dot{y}) = \frac{\mathrm{K}}{2a}\frac{\sin(2\pi x/a)}{\cosh(2\pi y/a)-\cos(2\pi x/a)} \qquad \text{Eq.2.2.3b}$$

These velocities are essentially the same as those given in Eq.2.2.2a, b, except for different signs and scaling.

Stuart (1967 and 1971) generalized the results of Lamb, and rescaled and translated the system to arrive at a streamfunction of $\Psi(x, y) = \ln(C\cosh y + A\cos x)$. Stuart's velocities are:

$$\dot{x} = \frac{\partial\Psi(x,y)}{\partial y} = \frac{C\sinh y}{C\cosh y + A\cos x} \qquad \text{Eq.2.2.4.a}$$

$$\dot{y} = -\frac{\partial\Psi(x,y)}{\partial x} = \frac{A\sin x}{C\cosh y + A\cos x} \qquad \text{Eq.2.2.4.b}$$

The constant C can be absorbed into the A constant by dividing both the numerator and denominator in each velocity by C, resulting in the velocities found in Eq.2.2.2a, b.

2.3 Numerical Studies of the Kelvin-Stuart Cat Eyes Dynamical System

Ottino (1989) included a numerical study of the Kelvin-Stuart cat eyes dynamical system as his Example 8.5 on p.254-260. The streamfunction used is the same as that in Eq.2.2.1, except for omission of the subtracted ln(1+A) term. The constant A, which can be considered to be a vorticity strength parameter, was set to the value of 0.8. A perturbation driving term of $\varepsilon\sin(\omega t)$ is added to the x equation of motion. Ottino describes features of the mixing and transport using a Melnikov method, which utilizes the Melnikov integral function (the Melnikov method and integral will be described in detail in section 3.1). No analytic form is given for the function $F(\omega)$ mentioned in Ottino's text on p.255, but a graph of $F(\omega)$ is given on p.257. The $F(\omega)$ function represents the amplitude dependence of the Melnikov integral function on the perturbation frequency ω. Ottino states that chaos should be maximized at the extremum of the $F(\omega)$ function, which is given as $\omega \approx 0.3$, and gives evidence to support this in his figure 8.5.3 on p.258. A graph calculated using the analytic form of the Melnikov integral as derived in Appendix A.3 of this dissertation is shown on the next page in Figure 2.3.1.

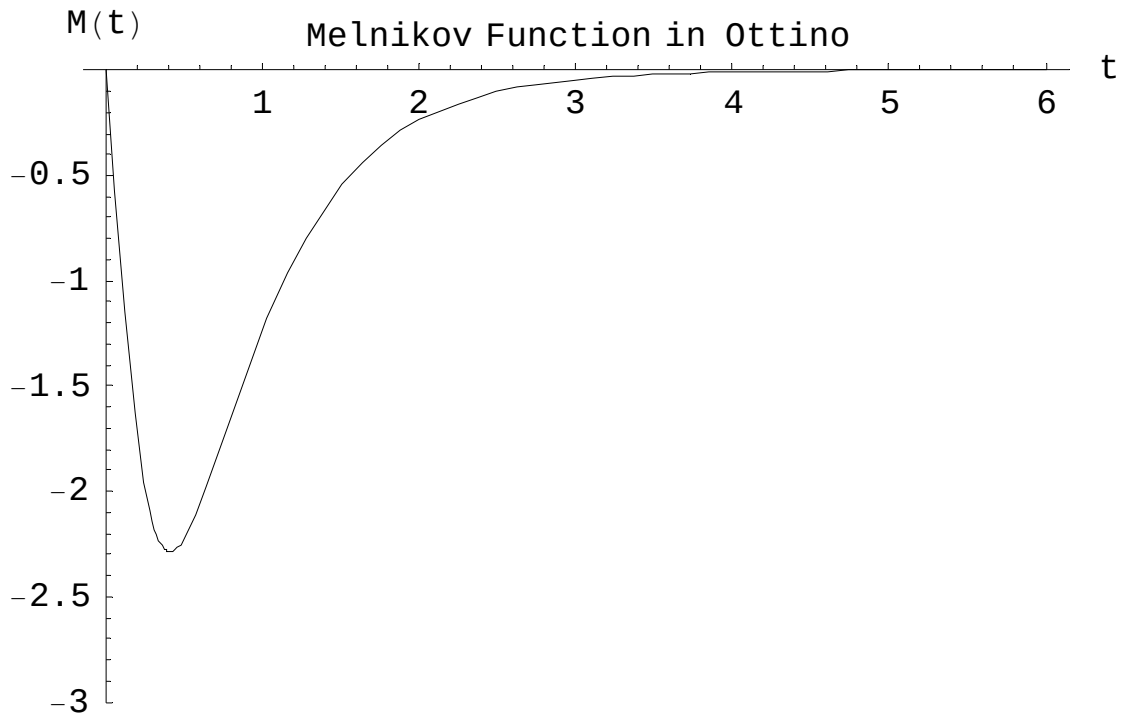

Figure 2.3.1 Calculated Melnikov amplitude function F(ω) for the Kelvin-Stuart system corresponding to the graph found on p.257 in Ottino (1989)

This graph is calculated using the function $F_2(\omega) = \dfrac{\sinh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}$ defined in eq.A.3.15 of Appendix A, section A.3, multiplied by a factor of $\dfrac{-2\pi}{\sqrt{A}}$ to give the correct Melnikov integral as defined by Ottino on p.255. The extremum of the $F_2(\omega)$ function can be calculated explicitly, and occurs at $\omega = 0.410768$. Figure 2.3.1, when compared to the graph in Ottino's figure 8.5.2 on p.257, provides an external check of the analytical derivations of the Melnikov integral found in the present work.

Tsega, Michaelides, and Eschenazi (2001) did a numerical Hamiltonian analysis using Poincaré sections of the Kelvin-Stuart cat eyes driven flow. As in Ottino (1989), A was set to 0.8, but this study differed from that of Ottino in that the perturbation term $\varepsilon \sin(\omega t)$ was added to both the x and y equations of motion. Chaos seemed to be maximized around $\omega \approx 0.3$, as was found by Ottino. One aim of this investigation was to observe differences in the Poincaré sections when the perturbation frequency ω was increased from a value of 0.5 to 2.4 while the perturbation strength ε was kept fixed at a value of 0.1. The results indicated a dependency of

mixing and transport upon the perturbation frequency ω. A desire to find an analytical expression for the transport and to study its dependence upon ω is a major motivation in performing the research in this dissertation.

Both numerical studies amply demonstrate what Aref (1984) termed chaotic advection, "a situation in which an innocuous, fully deterministic velocity field … produces an essentially stochastic response in the Lagrangian advection characteristic of a passive tracer", as well as Ottino's definition of mixing "… that from a kinematical viewpoint fluid mixing is the efficient stretching and folding of material lines and surfaces" (Ottino 1989). Lobe dynamics transport theory, which will be presented in chapter 3, is one attempt to mathematically describe this stretching and folding of material lines and surfaces.

2.4 Dynamical Systems Analysis

A detailed dynamical systems analysis for the Kelvin-Stuart dynamical system can be found in Appendix A - only significant results are reported in this section. The Hamiltonian for the perturbed, or driven flow, of the Kelvin-Stuart cat eyes dynamical system is

$$H(x,y) = \ln(\cosh y + A\cos x) - \ln(1+A) + \varepsilon(x+y)\sin(\omega t) \qquad \text{Eq.2.4.1}$$

The equations of motion for the driven system are:

$$\dot{x} = \frac{\partial H}{\partial y} = \frac{\sinh y}{\cosh y + A\cos x} + \varepsilon\sin(\omega t) \qquad \text{Eq.2.4.2a}$$

$$\dot{y} = -\frac{\partial H}{\partial x} = \frac{A\sin x}{\cosh y + A\cos x} + \varepsilon\sin(\omega t) \qquad \text{Eq.2.4.2b}$$

To find the fixed points of the system, set both equations of motion for the unperturbed system (Eq.2.2.2a, b) equal to 0 and solve for x and y. This results in fixed points at ($n\pi$, 0).

Eigenvalues for the points ($2n\pi$, 0) are $\lambda_{\pm} = \pm\frac{\sqrt{A}}{1+A}$, so ($2n\pi$, 0) are hyperbolic fixed points (saddles). The corresponding normalized eigenvectors are $\frac{1}{\sqrt{1+A}}\begin{pmatrix}1\\ \sqrt{A}\end{pmatrix}$ for eigenvalue λ_+, which gives the direction of the unstable manifold, and $\frac{1}{\sqrt{1+A}}\begin{pmatrix}1\\ -\sqrt{A}\end{pmatrix}$ for eigenvalue λ_-, which gives the direction for the stable manifold. The unstable manifold is the set of points whose

trajectories or orbits depart from a saddle point as $t \rightarrow -\infty$, and the stable manifold is the set of points whose trajectories or orbits approach a saddle point as $t \rightarrow \infty$ (Hilborn 1994). Eigenvalues for the points $((2n+1)\pi, 0)$ are $\mu_{\pm} = \pm i\frac{\sqrt{A}}{1-A}$, so $((2n+1)\pi, 0)$ are centers of rotation. These results verify the observations made about the flow of the Kelvin-Stuart cat eyes streamlines as discussed after Figure 2.2.1. For $A < 0$, the locations of the centers of rotation and the saddle points are translated by $\pm\pi$ in x, as in Lamb's analysis. The value of the eigenvalue

$$\lambda = \frac{\sqrt{A}}{1+A} \qquad \text{Eq.2.4.3}$$

will be used without the ± sign for the remainder of the dissertation.

The stable and unstable manifolds of neighboring saddle points of the unperturbed dynamical system intersect tangentially at all points and thus are coincident as one curve, the heteroclinic orbit or separatrix map. The heteroclinic orbit acts as a barrier to transport between the interior trajectories of the vortices and the exterior trajectories. No transport is possible across the separatrix for the unperturbed system. When the system is perturbed, the unstable and stable manifolds separate and are no longer coincident. If the stable and unstable manifolds of the perturbed system intersect transversely, then it can be shown (e.g. Guckenheimer and Holmes 1983) that an infinite number of transverse intersections exist, creating a structure known as a heteroclinic tangle. The heteroclinic tangle is composed of an infinite number of lobes. Transport in the perturbed system occurs through transfer of lobes between the interior and exterior of a vortex. The heteroclinic tangle and lobes will be studied in chapter 3.

Streamlines for the unperturbed Kelvin-Stuart dynamical system can be calculated by choosing a value for H in Eq.2.2.1 and then solving for y as a function of H and x. The heteroclinic orbit can be found by choosing $H = 0$. A form of the unperturbed heteroclinic orbit as parameterized by time may be derived from the unperturbed equations of motion (Eqs.2.2.2a, and b) (details of this calculation are found in Appendix A.2). The parameterized form of the heteroclinic orbit is important for deriving the Melnikov integral function. The parameterized form of the heteroclinic orbit for the unperturbed Kelvin-Stuart dynamical system is

$$\vec{\Gamma}_0(t) = \left(Arc\cos\left(\frac{\sinh^2(\lambda t) - \lambda/\sqrt{A}}{\cosh^2(\lambda t) - \lambda\sqrt{A}} \right), Arc\cosh\left(\frac{\cosh^2(\lambda t) + \lambda\sqrt{A}}{\cosh^2(\lambda t) - \lambda\sqrt{A}} \right) \right) \qquad \text{Eq.2.4.8}$$

The parameterized form of the heteroclinic orbit is graphed in Figure 2.4.1 on the next page.

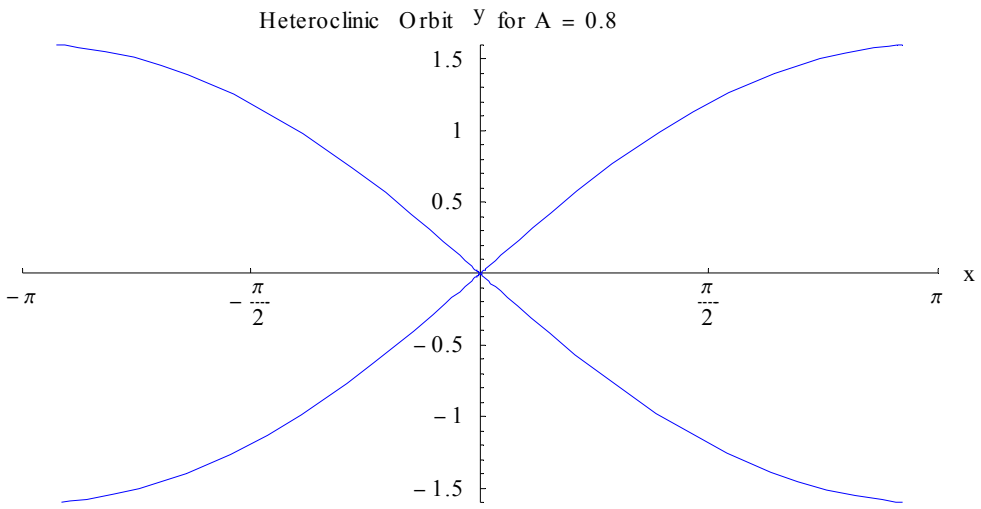

Figure 2.4.1 Heteroclinic Orbit – Separatrix Map

Other branches of the heteroclinic orbit can be found by inclusion of appropriate positive and negative signs and addition and subtraction of multiples of 2π.

Figure 2.4.2 shows a fundamental vortex cell, which is repeated with the 2π spatial periodicity of the Kelvin-Stuart Cat Eyes dynamical system. All analysis for the Kelvin-Stuart dynamical system will be for one fundamental cell, and thus applicable to all cells due to the spatial periodicity.

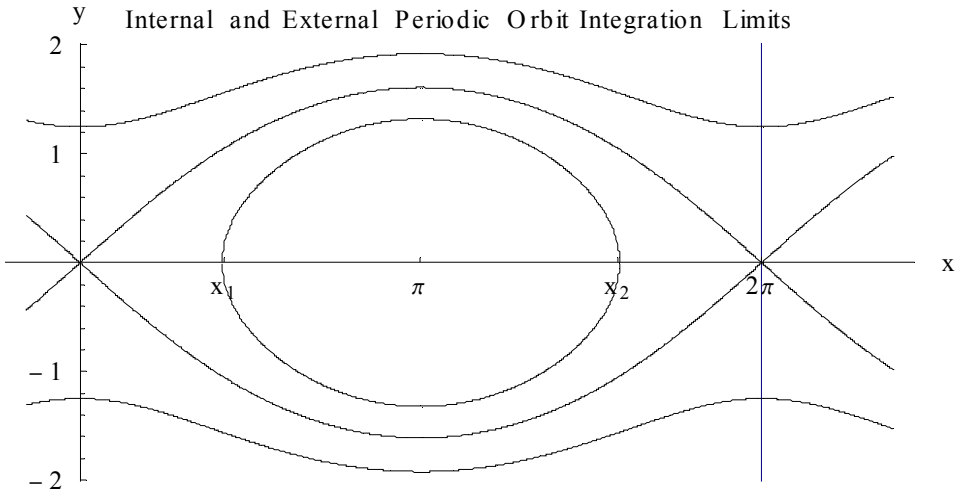

Figure 2.4.2 The heteroclinic orbit, and representative internal and external orbits with the associated limits of integration for each case.

As can be seen in Figure 2.4.2 on the previous page, interior orbits are closed curves, while exterior orbits exhibit the spatial periodicity of the system. Interior orbits have time periods defined as the time to complete one orbit of the closed curve. Exterior orbits have time periods defined as the time to complete one spatial period. Interior orbits are uniquely characterized by values of $H < 0$, while exterior orbits are uniquely characterized by values of $H > 0$. The period for any orbit can thus be expressed as a function of the value of H, for both internal and external orbits. These functions will be called the internal and external period functions, respectively. The period as a function of H can be calculated for both the interior and exterior orbits by integrating $\int dt = \int \frac{dx}{\dot{x}}$ over the limits of integration shown in figure 2.4.2 on the previous page. Performing the integrals results in period functions given by:

$$P_{\text{int}}(H) = \frac{4}{\lambda}\exp(H)K\left(\frac{1}{4\lambda^2}\exp(2H) - \left(\frac{1-A}{1+A}\right)^2\right), \quad H < 0 \qquad \text{Eq.2.4.4a}$$

$$P_{ext}(H) = \frac{4\exp(H)}{\sqrt{\exp(2H) - \left(\frac{1-A}{1+A}\right)^2}} K\left(\frac{2\lambda}{\sqrt{\exp(2H) - \left(\frac{1-A}{1+A}\right)^2}}\right), \quad H > 0 \qquad \text{Eq.2.4.4b}$$

K is the complete elliptic integral of the first kind, with the parameter equal to the square of the modulus, as defined in Arfken (1985). Details of the derivations of the internal and external period functions can be found in Appendix A.4. The period for both the internal and external period functions approaches ∞ as H approaches 0, since $H = 0$ represents the heteroclinic orbit, which has an infinite period. The minimum value of H for interior orbits approaches $\ln\left(\frac{1-A}{1+A}\right)$, which is the value of H at a vortex center. For $A = 0.8$, $\ln\left(\frac{1-A}{1+A}\right) = -2.1972246$. Graphs of the internal and external period functions for $A = 0.8$ are given in Figure 2.4.3 on the next page, with H ranging from the minimum value of -2.1972246 to a maximum of 2.1972246.

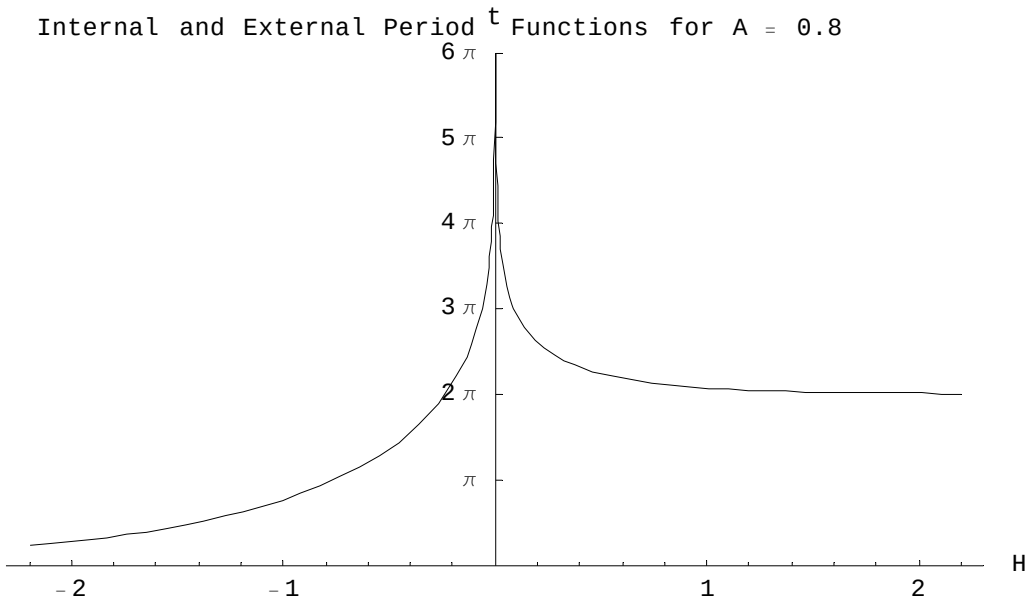

Figure 2.4.3 Internal and external period functions, with the internal function for negative H, and the external function for positive H

It is important to note that the graph shows ½ times the internal period function, rather than the actual internal period function, since the ½ factor appears everywhere that the internal period function is used (the factor is introduced in the Whisker map, which may be found in section 3.2). This factor of ½ is also included in the definition of the internal period function used in the programs found in Appendix B.

Approximate forms of the internal and external period functions are necessary for finding approximate inverse functions (used for finding bifurcation values as discussed in section 4.1) for the two period functions. Details of finding the approximations and inverses are found in Appendix A.4. The approximations for the period functions and the inverses for the period function approximations are found to be:

$$P_{\text{int}}(H) \approx \frac{2}{\lambda}\left[\ln(32\lambda^2) - \ln(-H)\right], \quad H < 0 \qquad \text{Eq.2.4.5a}$$

$$P_{ext}(H) \approx \frac{1}{\lambda}\left[\ln(32\lambda^2) - \ln(H)\right], \quad H > 0 \qquad \text{Eq.2.4.5b}$$

$$P_{\text{int}}^{-1}(t) \approx -32\lambda^2 \exp\left(\frac{-t}{2\lambda}\right), \quad H < 0 \qquad \text{Eq.2.4.6a}$$

$$P_{ext}^{-1}(t) \approx -32\lambda^2 \exp\left(\frac{-t}{\lambda}\right), \quad H < 0 \qquad \text{Eq.2.4.6b}$$

A graph depicting the relative error for the approximate period functions is given in Figure 2.4.4:

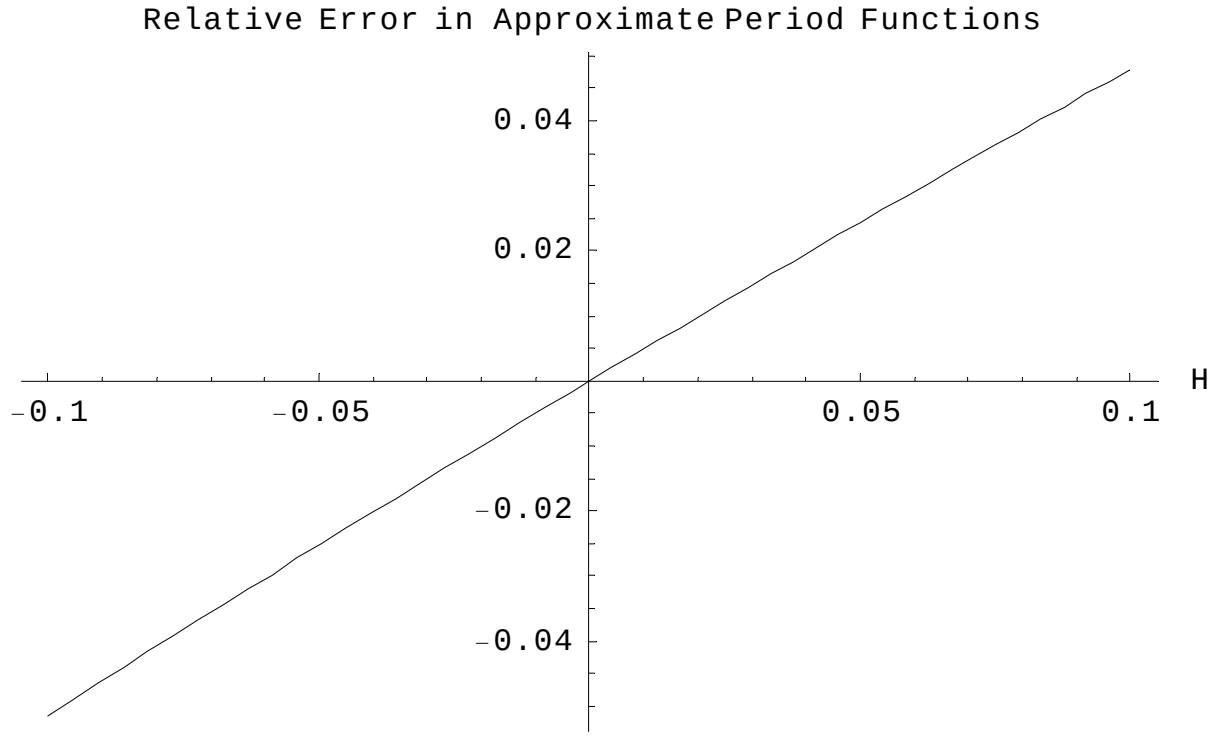

Figure 2.4.4 Relative error in the approximate period functions

All of the calculations performed in the research were within a range of H values between -0.01 and 0.01, which on the graph indicates relative errors on the order of 0.005. The relative error was calculated as (exact – approximation)/(exact).

Derivatives of the period function approximations are needed to find approximate values for times when bifurcations occur. The derivatives of the period function approximations are found to be:

$$P'_{\text{int}}(H) \approx -\frac{2}{\lambda H}, \quad H < 0 \qquad \text{Eq.2.4.7a}$$

$$P'_{ext}(H) \approx -\frac{1}{\lambda H}, \quad H > 0 \qquad \text{Eq.2.4.7b}$$

A graph depicting the relative error for the derivatives of the approximate period functions is given in Figure 2.4.5:

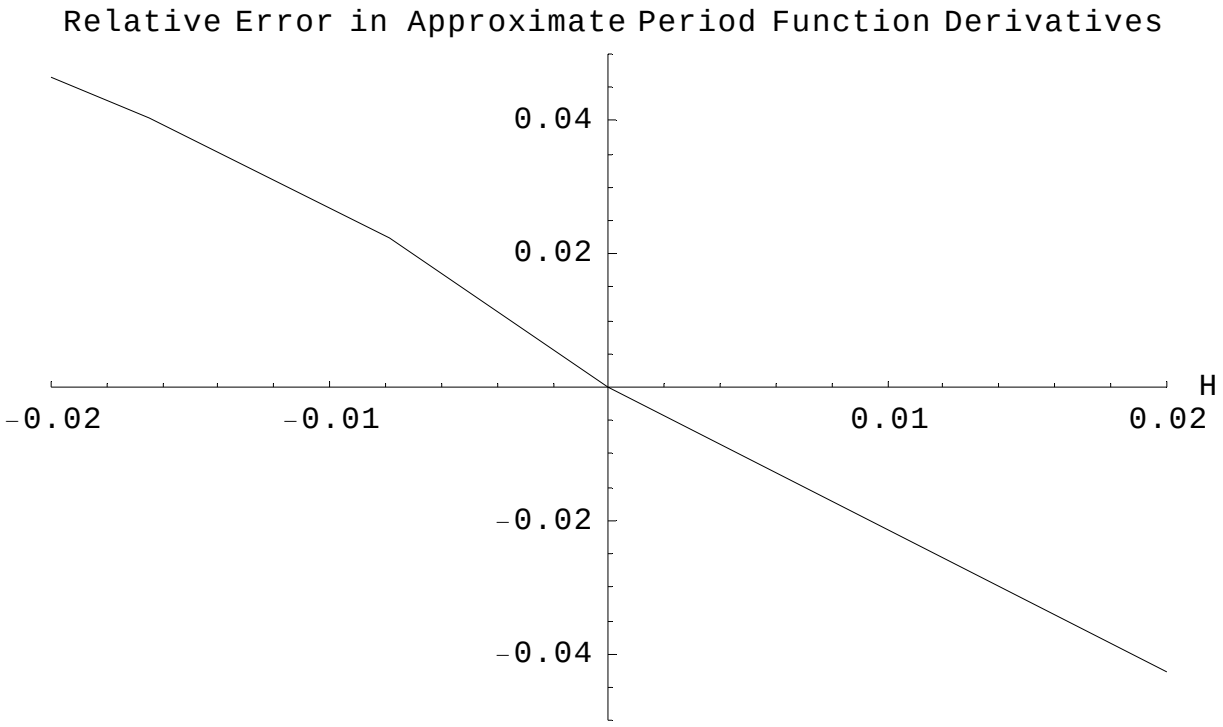

Figure 2.4.5 Relative error in the derivatives of the approximate period functions

Chapter 3

Melnikov Function, Whisker Map, and Lobe Dynamics Transport Theory

3.1 Melnikov Function

Lobe dynamics transport theory, as implied by its name, focuses on what are called lobes, which are formed by the transverse intersections of the stable and unstable manifolds in the heteroclinic tangle of the perturbed dynamical system. Crucial to being able to work with the lobes is to find a function that is proportional to the distance between the stable and unstable manifolds at any time, so that the intersection points may be found where the distance is zero. One such function is the Melnikov function, which has been used extensively in dynamical systems work and can be found in many dynamical systems texts (Arrowsmith and Place 1990, Guckenheimer and Holmes 1983, Hilborn 1994, Nayfeh and Balachandram 1995, Wiggins 1990, and Wiggins 1992). Guckenheimer and Holmes (1983) define the distance between the manifolds as

$$d(t_0) = \frac{\varepsilon M(t_0)}{\left|\vec{f}(q_0(0)\right|}, \quad \vec{f} = (\dot{x}, \dot{y}) \qquad \text{Eq.3.1.1}$$

The $M(t_0)$ function in the numerator of the distance function is the Melnikov function. The distance is along the perpendicular of the $\vec{f}$ vector, as shown in Figure 3.1.1.

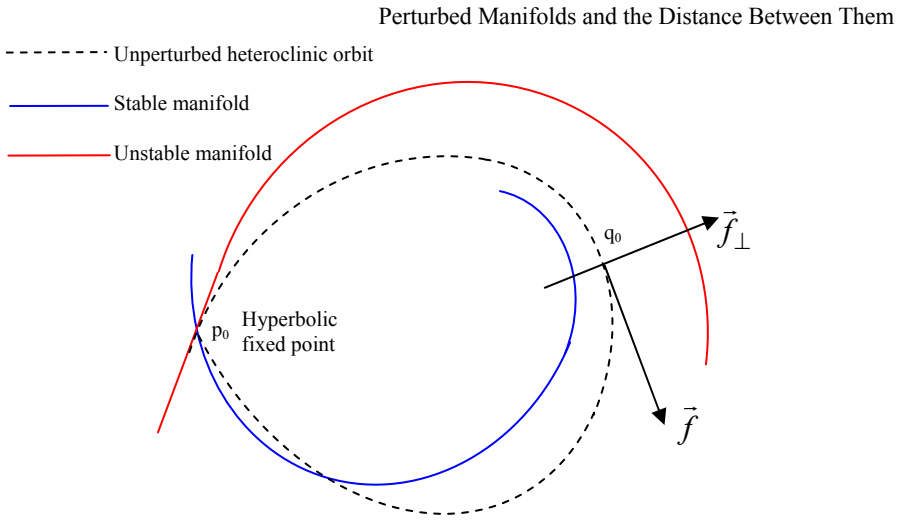

Figure 3.1.1 Perturbed Manifolds and the Distance Between Them

The vectors $\vec{f}$ (unperturbed equations of motion) and $\vec{g}$ (perturbation terms in x and y) for the driven Kelvin-Stuart flow are defined as:

$$\vec{f} = \begin{pmatrix} \dfrac{\sinh y}{\cosh y + A\cos x} \\ \dfrac{A\sin x}{\cosh y + A\cos x} \end{pmatrix}, \quad \vec{g} = \begin{pmatrix} \sin(\omega t) \\ \sin(\omega t) \end{pmatrix} \qquad \text{Eq.3.1.2}$$

The Melnikov function is defined as:

$$M(t_0) = \int_{-\infty}^{\infty} dt \exp(-\int \vec{\nabla} \cdot \vec{f}(\vec{\Gamma}_0(s))\, ds)\, \vec{f}(\vec{\Gamma}_0(t)) \wedge \vec{g}(\vec{\Gamma}_0(t), t + t_0) \qquad \text{Eq.3.1.3}$$

The exponential term drops out of Eq.3.1.3 since $\vec{\nabla} \cdot \vec{f} = 0$ for the Kelvin-Stuart Cat Eyes dynamical system. The wedge product is defined as $\vec{f} \wedge \vec{g} = f_x g_y - f_y g_x$. The details of the derivation of the analytical result for the Melnikov integral for the Kelvin-Stuart driven flow can be found in Appendix A.3. The resulting function is

$$M(t_0) = \pm 2\pi \sqrt{(F_1(\omega))^2 + (F_2(\omega))^2}\, \sin(\omega t + \varphi)\,,$$

$$F_1(\omega) = \frac{\cosh(\omega\delta / 2\lambda)}{\cosh(\omega\pi / 2\lambda)}, \quad F_2(\omega) = \frac{\sinh(\omega\delta / 2\lambda)}{\cosh(\omega\pi / 2\lambda)}, \quad \varphi = Arc\tan\left(\frac{F_2(\omega)}{F_1(\omega)}\right), \qquad \text{Eq.3.1.4}$$

$$\delta = Arc\cos\left(\frac{1 - A}{1 + A}\right), \quad \lambda = \frac{\sqrt{A}}{1 + A}$$

Choose $\varphi = 0$ in Eq.3.1.4 so that the Melnikov function has exactly two zeroes, at $t = 2n\dfrac{\pi}{\omega}$ and $t = (2n+1)\dfrac{\pi}{\omega}$, for each fundamental period of the perturbation frequency ω. These zeroes specify the lobes for the Kelvin-Stuart Cat Eyes driven flow. The graphs of the Melnikov function for $A = 0.8$ and for several values of the perturbation frequency ω are found in Figure 3.1.2 on the next page.

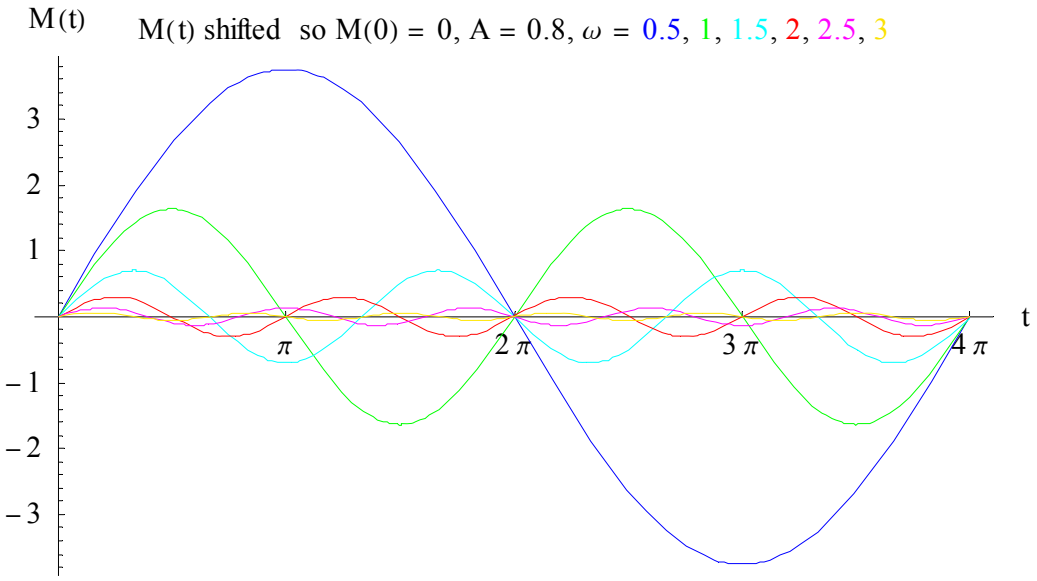

Figure 3.1.2 Shifted Melnikov function, with zero at t = 0

The amplitude of the Melnikov function is determined by a function of the perturbation frequency ω, given by

$$C(\omega) = 2\pi\sqrt{(F_1(\omega))^2 + (F_2(\omega))^2} \qquad \text{Eq.3.1.5}$$

which is graphed in Figure 3.1.3.

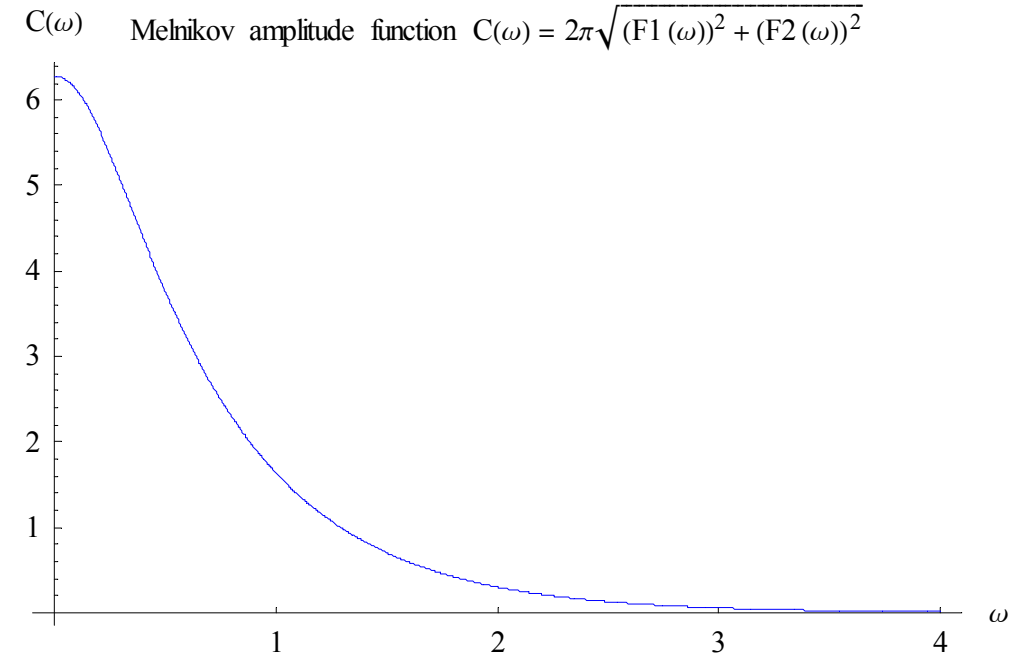

Figure 3.1.3 Melnikov amplitude function C (ω)

Notice that the amplitude falls sharply with increasing frequency. It will be shown that lobe size depends on the amplitude of the Melnikov function, so lobe size will also decrease rapidly with increasing frequency ω. The area of a lobe is the area under the Melnikov function for one half of a fundamental period.

3.2 Whisker Map

The unperturbed Kelvin-Stuart Hamiltonian is autonomous, i.e., time-independent. The perturbation function introduces time dependence into the system. Observations of the system at regular intervals of the period of the driving frequency can be considered as Poincaré return maps (Rom-Kedar and Wiggins 1990, Wiggins 1992). Lobes are found from zeroes in the Melnikov function, which correspond to zeroes of the driving perturbation. These lobes undergo stretching and folding, which leads to intersecting with previous lobes. Intersection areas must be subtracted from a lobe when calculating transport, so finding these intersection points is crucial. Chirikov (1979) introduced a map he called the Whisker map to follow the perturbed manifolds near the unperturbed separatrix map (Hilborn 1994 calls this the Chirikov Standard Map, and defines it in terms of angular coordinates). The Whisker map is used to find the intersection points where a lobe intersects another lobe. The Whisker map is defined by

$$h_{n+1} = h_n + M(t_n), \quad h_0 = 0 \qquad \text{Eq.3.2.1a}$$

$$t_{n+1} = \begin{cases} t_n + \frac{1}{2} P_{\text{int}}(\varepsilon\, h_{n+1}), & h_{n+1} < 0 \\ t_n + P_{ext}(\varepsilon\, h_{n+1}), & h_{n+1} > 0 \end{cases} \qquad \text{Eq.3.2.1b}$$

The h iterates calculate perturbed Hamiltonian values, and depend upon the Melnikov function. The t iterates calculate the periods of orbits using the internal and external period functions, and are not directly related to the period of the perturbation frequency. Since the lobes are defined by zeroes of the Melnikov function, which do depend on the perturbation period, lobe intersection times, which are found using the Whisker map, will be bracketed between perturbation periods. The h and t Poincaré cross-sections used in the Whisker map are shown in Figure 3.2.1.

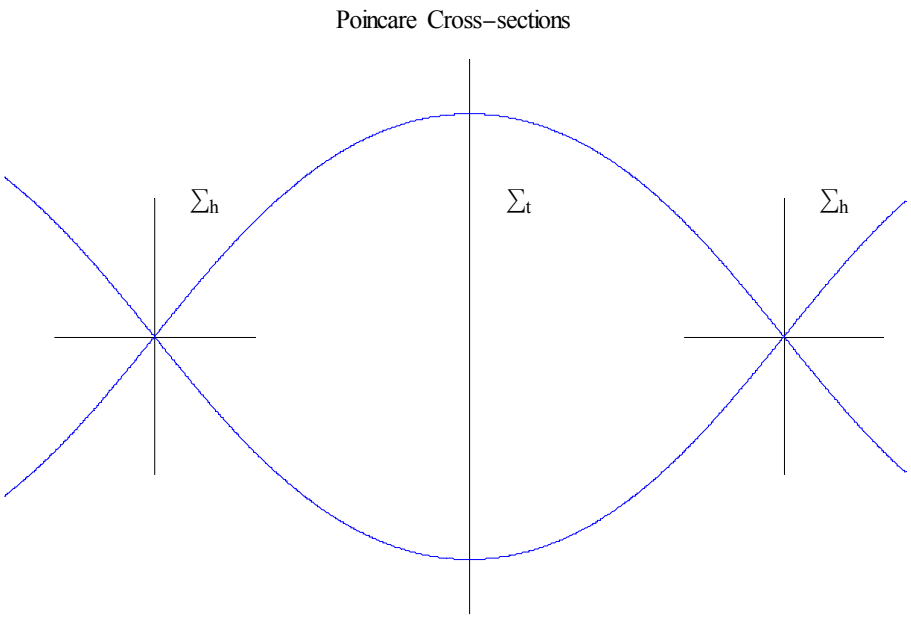

Figure 3.2.1 Heteroclinic Orbit and h and t Poincaré Cross-sections for the Whisker Map

The *t* cross-section occurs at $x = \pi$ and the *h* cross-sections occur at $x = 0$ and $x = 2\pi$. Note that an internal orbit will travel from one *h* cross-section to the other in one half of its period, hence the factor of ½ for the internal period function in Eq.3.2.1b. This can be seen in Figure 3.2.2.

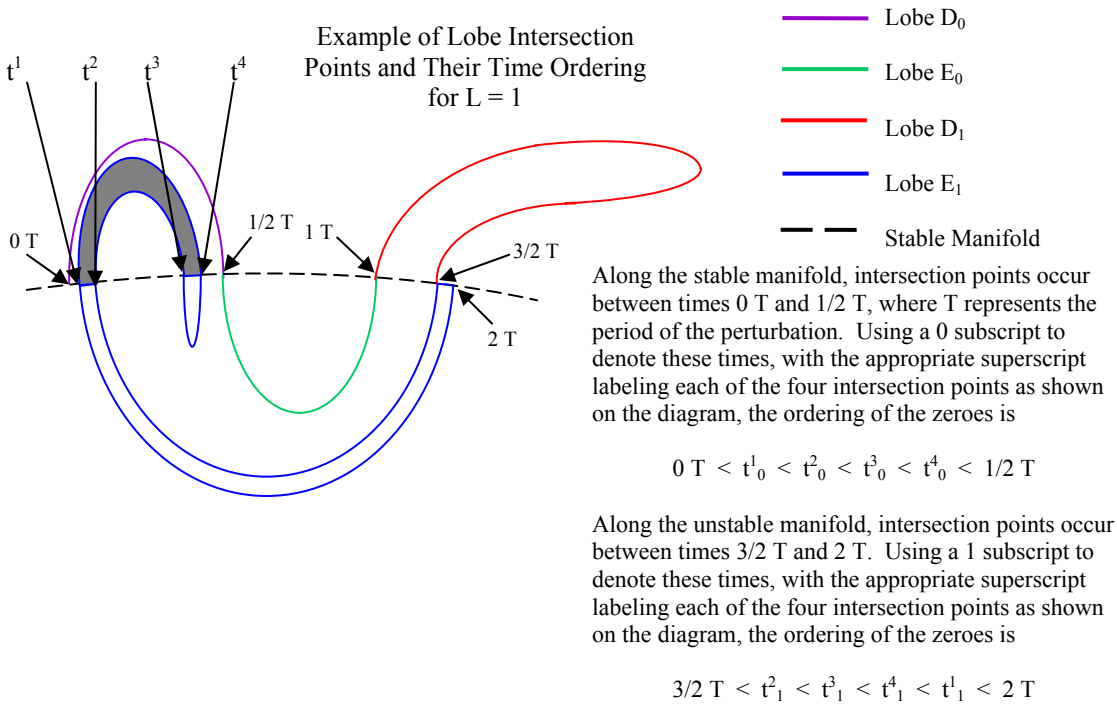

Figure 3.2.2 Example of lobe intersection points and their time ordering

The four intersection points in the upper left of Figure 3.2.2, t^1, t^2, t^3, and t^4, have different subscripts depending upon whether time is calculated along the stable manifold (subscript 0) or along the unstable manifold (subscript 1). The lobe notation (D, E) used in Figure 3.2.2 will be explained in section 3.3.

3.3 Lobe Dynamics Transport Theory

Rom-Kedar and Wiggins developed a transport theory based on the evolution of lobes as the system is iterated by the perturbation function (Rom-Kedar, Leonard and Wiggins 1990, Rom-Kedar and Wiggins 1990, Rom-Kedar and Wiggins 1991, and Wiggins 1992). They proved basic rules and properties of transport related to lobe dynamics which can be used with many general dynamical systems. The Kelvin-Stuart system is one of three systems studied as an example (the other two are the cubic potential and the Duffing oscillator).

Several basic terms used in lobe dynamics transport theory can be defined as follows, with defined terms in bold face. (The definitions here modify those found in Rom-Kedar and Wiggins (1990) and Wiggins (1992), using notation in Rom-Kedar (1990, 1994) and Litvak-Hinenzon (1996).). Illustrative examples of these definitions are shown in Figure 3.3.1 on the following page.

1) Points p_i, q_i in phase space are called **heteroclinic points** if they belong to both a stable manifold $W^s(x^*_m)$ and an unstable manifold $W^u(x^*_n)$, where $x^*_{m,n}$ denote hyperbolic fixed points of the system.

2) Consider a heteroclinic point p_0 and let $S[x^*_m, p_0]$ denote the segment of $W^s(x^*_m)$ from x^*_m to p_0 and $U[x^*_m, p_0]$ denote the segment of $W^u(x^*_n)$ from x^*_n to p_0. Then p_0 is called a **primary intersection point (pip)** if $S[x^*_m, p_0]$ and $U[x^*_n, p_0]$ intersect only in q_0. Pips are located at zeroes of the Melnikov function. All p_i, q_i in Figure 3.3.1 are pips.

3) Let pip q_i be in between and adjacent to pips p_i and p_{i+1}. A **D lobe** is the area bounded by the segments $S[p_i, q_i]$ and $U[p_i, q_i]$, and an **E lobe** is the area bounded by the segments $S[p_{i+1}, q_i]$ and $U[p_{i+1}, q_i]$. A combination of a D lobe and its neighboring E lobe, with the same subscripts, is termed a **turnstile lobe**.

4) A **region** is a simply connected domain of phase space with boundaries consisting of the phase space and segments of stable and unstable manifolds starting at hyperbolic fixed points and ending at either pips or at the boundary of the phase space.

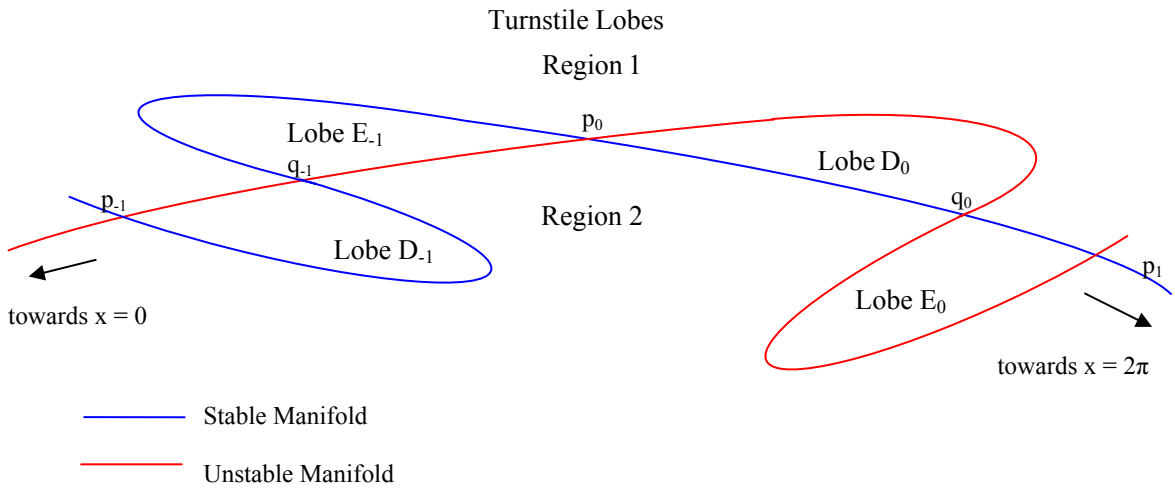

Figure 3.3.1 Illustrative examples of pips, lobes, turnstile lobes, and regions

The two basic rules of lobe dynamics transport theory as developed by Rom-Kedar and Wiggins (1990) are:

1) The transport between regions is determined by the initial distribution of the relevant lobes in the regions (the relevant lobes are the lobes that move from one region to the other under iteration of the system), i.e., the turnstile lobes. Only the lobes are transported for any iteration – any area not enclosed in a lobe will not be transported.

2) The initial distribution of all the relevant lobes can be found by following the evolution of a finite collection of lobes that are images or preimages of all the relevant lobes. Images and preimages refer to forward and backward iterates of the turnstile lobes. Forward iterates are denoted by positive subscripts, while back iterates use negative subscripts.

The set of all forward and backward iterates (images and preimages) forms the heteroclinic tangle for the system. The evolution of the lobes and the heteroclinic tangle are shown in Figure 3.3.2.

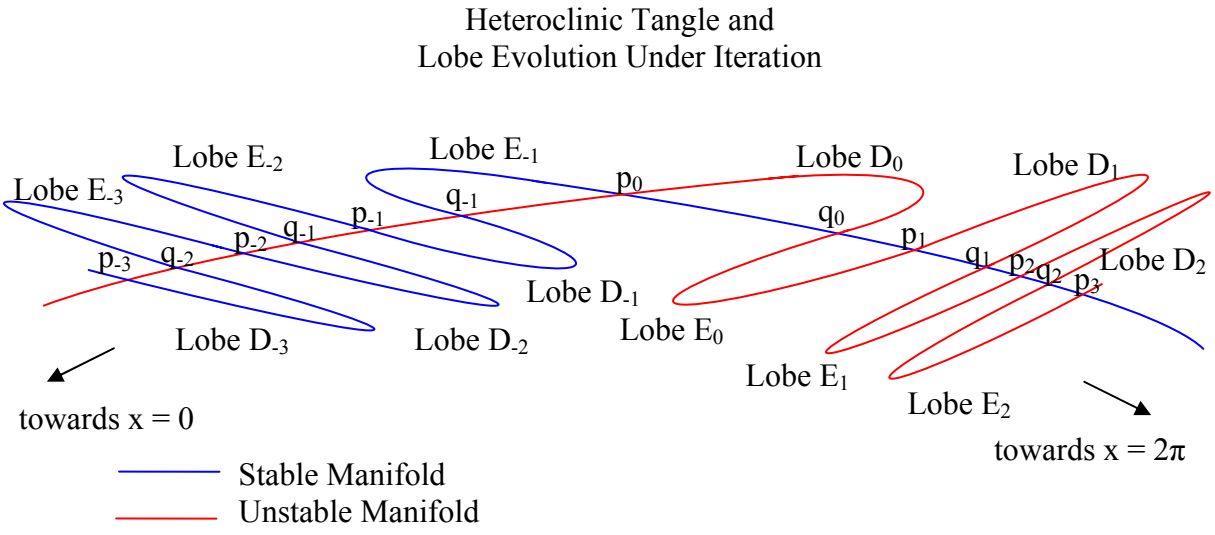

Figure 3.3.2 The heteroclinic tangle and examples of lobe evolution under iteration of the driving perturbation

Five general properties of lobe transport are:

1) Only one turnstile lobe is involved per iteration in transport between neighboring regions. (Complicated collections of lobes can be grouped so as to be considered one turnstile.)
2) Fluxes (transport) are determined by lobe contents.
3) The lobe contents are determined by lobe intersections.
4) The lobe dynamics give a tool to compute the lobe intersections.
5) Conservation of lobe area gives global results about the relations between the total transport rates between different regions. Conservation of area only occurs for non-dissipative systems, such as incompressible, inviscid fluids.

Lobe evolution can get quite complex, with lobes intersecting previous lobes. This is essential for properties 2, 3, and 4. A method of calculating lobe intersections is necessary to apply lobe dynamics transport theory. The Topological Approximation Method developed by Rom-Kedar is one such method, which will be outlined in chapter 4. The complexity of lobe evolution is illustrated on the next page in Figure 3.3.3. Representative internal and external lobes are also shown.

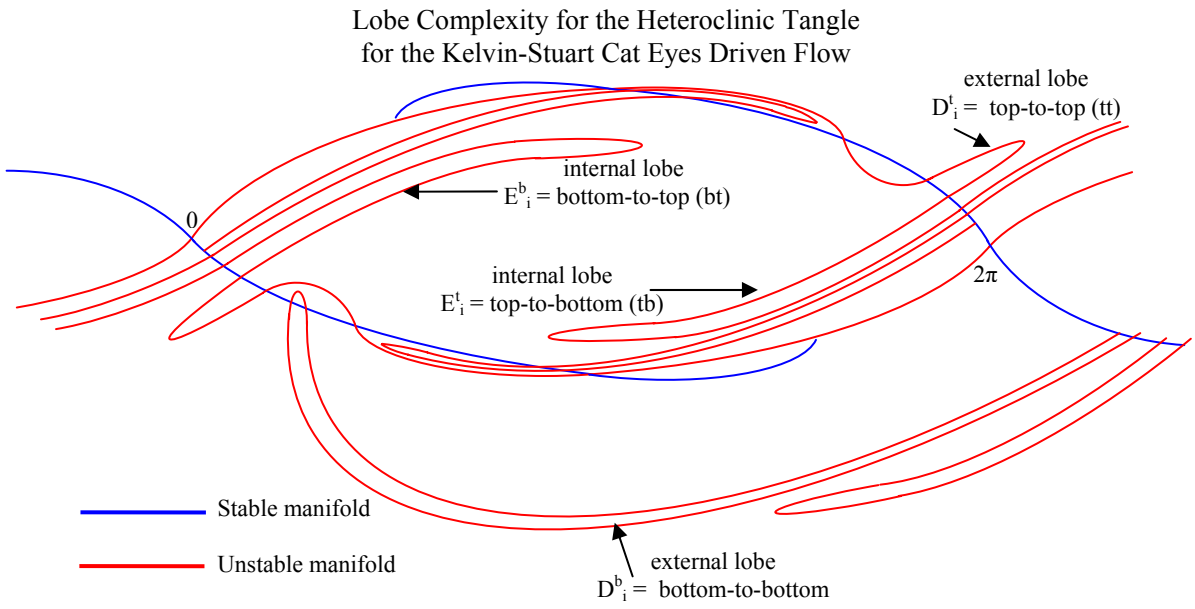

Figure 3.3.3 Lobe complexity for the Kelvin-Stuart cat eyes driven flow, with representative internal and external lobes

From Figure 3.3.3 different lobes and their primary intersections can be identified (red for internal lobes and blue for external lobes will be used throughout the remainder of the dissertation):

Internal Lobes

tb (top-to-bottom): $E^t_j \cap D^b_0$

bt (bottom-to-top): $E^b_j \cap D^t_0$

External Lobes

tt (top-to-top): $D^t_{k+1} \cap E^t_0$

bb (bottom-to-bottom): $D^b_{k+1} \cap E^b_0$

All other intersections are second order, which means that a lobe will be inside another lobe when that lobe intersects a third lobe, or even higher order. For a first order approximation, only the primary intersections will be used in this research. The smallest integer j for internal lobes (k for external lobes), such that the intersection is not empty is called the **structure factor or structure index** (Easton 1986), and is labeled with an L.

Lobes for the Kelvin-Stuart Cat Eyes System for L = 1, perturbation strength $\varepsilon = 0.1$, and perturbation frequency $\omega = 1.15$ are graphed in Figure 3.3.4.

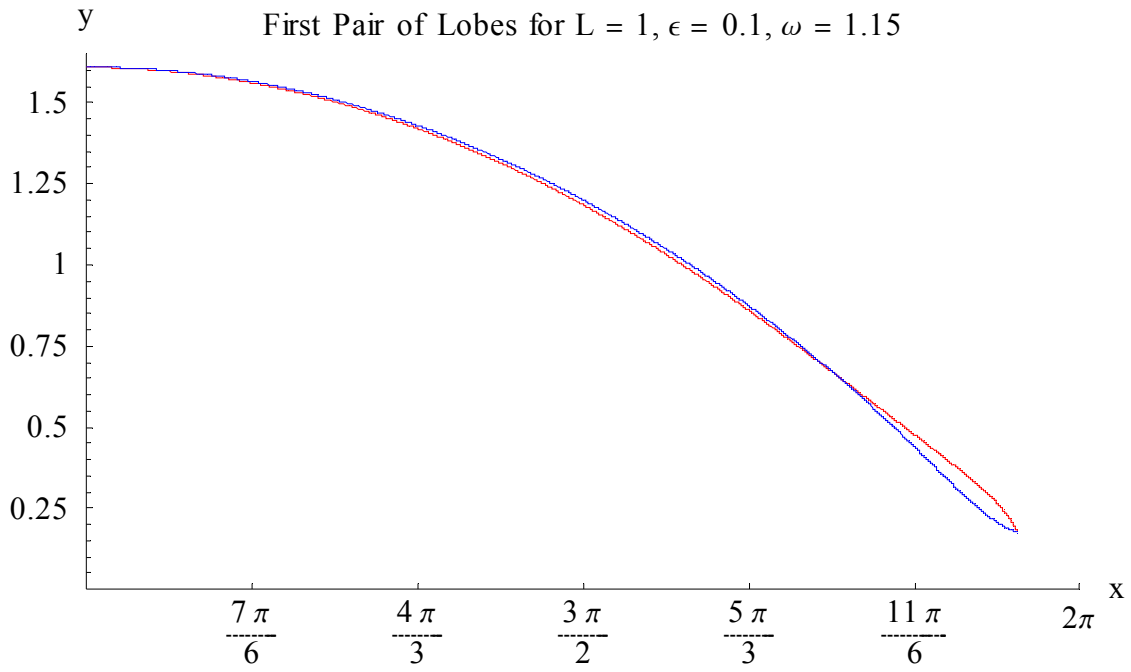

Figure 3.3.4 The first pair of lobes for L = 1, $\varepsilon = 0.1$, and $\omega = 1.15$

Chapter 4
Topological Approximation Method (TAM)

4.1 Six Conditions for Applying the Topological Approximation Method

Rom-Kedar developed what she termed the Topological Approximation Method in a series of papers (Rom-Kedar 1990, 1993, 1994, 1995). She specified six necessary conditions in order to apply the Topological Approximation Method:

1) H_0 and H_1 are C^r, $r \geq 4$, and H_1 is periodic in t with period $T = 2\pi/\omega$. H_0 is the unperturbed Hamiltonian, and H_1 is the perturbation term.
2) The origin is a hyperbolic fixed point of the unperturbed dynamical system.
3) For the unperturbed system, the stable and unstable manifolds coincide, forming a heteroclinic orbit $\vec{\Gamma}_0$.
4) For the unperturbed system, the interior of $\vec{\Gamma}_0$ is foliated by periodic orbits with period $P_{int}(H_0)$, labeled by their values of the unperturbed Hamiltonian H_0 (i.e., $H_0 < 0$). For $H_0 > 0$, orbits are spatially periodic with spatial period 2π.
5) H_1 is analytic in (x, y) for all t.
6) The Melnikov function M (t) has at least two simple zeroes in within a fundamental period interval. Moreover, M (t) = c has a finite number of solutions within a fundamental period interval.

The Kelvin-Stuart driven flow will be shown to satisfy all six conditions.

1) For the Kelvin-Stuart driven flow, the unperturbed Hamiltonian is

$$H_0(x,y) = \ln(\cosh y + A\cos x) - \ln(1+A) \qquad \text{Eq.4.1.1a}$$

while the perturbation term is

$$H_1(x,y,t) = \varepsilon(x+y)\sin(\omega t) \qquad \text{Eq.4.1.1b}$$

Both functions can be seen to be C^∞ if $|A| < 1$, so condition 1 is satisfied.

2) All of the points $(2n\pi, 0)$ are hyperbolic fixed points of the unperturbed dynamical system (as shown in Appendix A.1), so condition 2 is satisfied.

3) Condition 3 is satisfied, with the heteroclinic orbit given by Eq.2.4.8 and shown in Figure 2.4.3.

4) The internal and external period functions satisfying condition 4 are calculated in Appendix A.4 and shown in Figure 2.4.2.

5) The spatial dependence of the perturbation term is $(x + y)$, which is analytic, thus satisfying condition 5.

6) The Melnikov function can be written as an amplitude function $C(\omega)$ multiplied by the perturbation term $\sin(\omega t)$. Thus, the zeroes of the Melnikov function are precisely the zeroes of the perturbation term, which satisfy condition 6.

4.2 Outline of the Topological Approximation Method

The Topological Approximation Method involves the following ten steps:

1) Write the Hamiltonians for the unperturbed and perturbed dynamical systems.
2) Find expressions for the periods of the periodic orbits as functions of the values of the unperturbed Hamiltonian, for both internal and external orbits.
3) Classify the lobes of the dynamical system using characteristic labels using the structure factor L following the trellis system developed by Easton.
4) Calculate the Melnikov integral using the expression for the heteroclinic orbit of the unperturbed system.
5) Use the Melnikov integral and the internal and external period functions to find the Whisker map, which is used as the Poincaré return map.
6) Find the bifurcation curves from the Whisker map, the Melnikov integral and the period functions.
7) Describe the lobe dynamics of the system, which can be written in the form of a matrix, the lobe dynamics transfer matrix, including the number and location of various weighting values.
8) Find the zeroes of the Whisker map and the Melnikov integral to calculate area integrals which provide estimates of the initial escape rates.
9) Use the estimated initial escape rates to calculate weighting values.
10) Calculate the transport for any iteration by multiplying powers of the lobe dynamics transfer matrix times the initial escape rates vector.

Steps one through five have already been carried out in previous sections. The details of steps six through ten will be given in the remaining sections of this chapter.

4.3 Bifurcation Curves

Bifurcation curves represent values of the perturbation strength ε as a function of perturbation frequency ω, where lobe intersections undergo bifurcations. Bifurcation curves also depend on the structure factor L. Minimum bifurcation curves indicate the tip of a lobe just entering another lobe, i.e., going from 0 through 1 to 2 intersection points (1 intersection point at the actual bifurcation curve). Maximum bifurcation curves indicate the tip of an intersecting lobe that is just leaving another lobe, i.e., going from 2 through 3 to 4 intersection points (3 intersection points at the actual bifurcation curve). For the Kelvin-Stuart driven flow bifurcation curves are found to be different for internal and external lobes.

Bifurcation curves occur when both

$$h_2 = M(t_0) + M(t_1) = 0 \text{ and } {dh_2}/{dt_0} = 0 \qquad \text{Eq.4.3.1a}$$

with t_1 defined as

$$t_1 = t_0 + P(\varepsilon M(t_0)) \qquad \text{Eq.4.3.1b}$$

where P (H) is either the internal or external period function, whichever applies, with the ½ factor incorporated into the internal period function. Further, t_1 has to meet the following condition:

$LT < t_1 < (L+1)\,T$ for internal lobes, and $(L+1)\,T < t_1 < (L+2)\,T$ for external lobes

Eq.4.3.1c

Given that $M(t_0) = C(\omega)\sin(\omega t)$, Eq.4.3.1a,b,c imply that

$$t_1 = (L+1)T - t_0 = (L+1)\frac{2\pi}{\omega} \qquad \text{Eq.4.3.2}$$

$$\frac{dh_2}{dt_0} = 0 \Rightarrow \frac{dM(t_0)}{dt_0} + \frac{dM(t_1)}{dt_1}\frac{dt_1}{dt_0} = 0 \qquad \text{Eq.4.3.3}$$

Using Eq.4.3.1b to find dt_1/dt_0 leads to

$$M'(t_0) + M'(t_1)\left[1 + \frac{dP}{dH}\frac{d(\omega M(t_0))}{dt_0}\right] = 0 \qquad \text{Eq.4.3.4a}$$

The derivatives of the approximate period functions, found in Eq.2.4.7a, b (making sure to include the ½ factor for the internal period function), allow Eq.4.3.4a to be rewritten as

$$M'(t_0) + M'(t_1)\left[1 - \frac{\omega M'(t_0)}{\lambda M(t_0)}\right] = 0 \qquad \text{Eq.4.3.4b}$$

Substituting $M(t_0) = C(\omega)\sin(\omega t)$ and $M'(t_0) = \omega C(\omega)\cos(\omega t)$ into Eq.4.3.4b and using Eq.4.3.1b to substitute for t_1 leads to

$$C(\omega)\cos(\omega t_0)\left[2 - \tfrac{\omega}{\lambda}\cot(\omega t_0)\right] = 0 \qquad \text{Eq.4.3.4c}$$

Solving Eq.4.3.4c for t_0 gives the following results

$$\begin{cases} t_0 = \dfrac{\pi}{2\omega}, \quad t_0 = \dfrac{1}{\omega} Arc\tan\left(\dfrac{\omega}{2\lambda}\right) \\ t_0 = \dfrac{3\pi}{2\omega}, \quad t_0 = \pi + \dfrac{1}{\omega} Arc\tan\left(\dfrac{\omega}{2\lambda}\right) \end{cases} \qquad \text{Eq.4.3.5}$$

These t_0 values represent the time of the intersection point when the tip of a lobe first intersects another lobe, as shown in Figure 3.2.2, or the time when an intersecting lobe tip begins to leave the lobe it is intersecting. The first set of values for t_0 occurs when an E_L (internal) lobe intersects a D_0 external lobe. The second set of t_0 values occurs when a D_{L+1} (external) lobe intersects an E_0 internal lobe. The value of ε as a function of ω can be found by using the inverses of the approximate period functions (remembering to include the ½ factor for the internal period function), found in Eq.2.4.6a, b, with Eq.4.3.1b to solve for ε.

$$\varepsilon = \frac{P^{-1}(t_1 - t_0)}{M(t_0)} \qquad \text{Eq.4.3.6}$$

Substituting the values of t_0 from Eq.4.3.5 into Eq.4.3.6 will produce minimum and maximum values for ε, specifically, the second value in each set will yield the minimum, and the first value will give the maximum. Minimum bifurcation curves indicate the tip of a lobe just entering another lobe, i.e., going from 0 through 1 to 2 intersection points (1 intersection point for the actual bifurcation curve). Maximum bifurcation curves indicate the tip of an intersecting lobe that is just leaving another lobe, i.e., going from 2 through 3 to 4 intersection points (3 intersection points for the actual bifurcation curve).

The bifurcation values found using Eq.4.3.5 and Eq.4.3.6 are only approximate values. The actual values can be found using a root finding algorithm, using the approximate values as initial estimates, to solve Eq.4.3.1a, b, c directly. Calculated minimum and maximum values for

the internal and external lobes differ, even though the approximate values are equal for both cases. A Mathematica® module (developed with the assistance of Ruskeepää 2004 and Wolfram 1999) to calculate the approximate and exact values for the minimum and maximum bifurcation curves for the Kelvin-Stuart cat eyes driven flow can be found in Appendix B.1.

4.4 Symbolic Dynamics, the Lobe Dynamics Transfer Matrix, and the Initial Vector

Easton (1986) developed a lobe classification scheme using the structure factor (or structure index) L, which was the minimum number of iterations a lobe needed to intersect another lobe. Rom-Kedar (1990, 1993, and 1994) refined this classification scheme and devised a symbolic dynamics for the lobes. The lobe symbolic dynamics describe how strips - parts of lobes - evolve under iteration of the Whisker map. Strips go to other strips (either the same type of strip or a different strip) under iteration. Weighting factors define what fraction of a strip goes to another strip upon the next iteration. All of the possible strips included in the symbolic dynamics can be written as a vector, the initial escape rates vector. The actual dynamics of strip evolution under iteration can be written as a matrix, the lobe dynamics transfer matrix. The strips, and thus the lobes, at any iteration can be calculated by multiplying an initial escape rates vector by the appropriate power of the lobe dynamics transfer matrix.

Strips are defined by the lobe boundaries as follows:

1) f_1 strips for E_n internal lobes are bounded by either their own stable manifold boundary or the stable manifold boundary of a D_j external lobe, $j > L$, and the stable manifold boundary of D_0, while f_1 strips for D_n external lobes are bounded by either their own stable manifold boundary or the stable manifold boundary of an E_j internal lobe, $j > L$, and the stable manifold boundary of E_0.

2) f_{L+1} strips for E_n internal lobes are bounded by the stable manifold boundary of the D_1 external lobe and the stable manifold boundary of D_0, while f_{L+1} strips for D_n external lobes are bounded by the stable manifold boundary of the E_1 internal lobe and the stable manifold boundary of E_0.

3) f_k strips for E_n internal lobes are bounded by the stable manifold boundaries of the D_{L+1-k} and $D_{L+1-(k-1)}$ external lobes, while f_k strips for D_n external lobes are bounded by the stable manifold boundaries of the E_{L+1-k} and $E_{L+1-(k-1)}$ internal lobes.

4) g strips are in the interior of the D_0 (E_0) lobes. The g strips represent the area of intersection between the lobes, which must be calculated and subtracted from the overall area of a lobe to accurately determine the amount of transport at any iteration. The first order approximation of the lobe intersections assumes that once a g strip is created, the area of the g strip is sufficient for calculating the lobe transport and need not be tracked further. Second and higher order approximations would include tracking a g strip as it is stretched and folded within the lobe it intersects as that lobe is stretched and folded by further iterations.

Figure 4.4.1 presents an example of lobe symbolic dynamics using the forced cubic potential from Rom-Kedar's papers (1990, 1993, and 1994). The g strips defined in 4) above are represented by the g_1 strips in the diagram. The g_0 strips shown in the diagram are not needed, since for the cubic potential, once a strip becomes a g strip, it always remains a g strip, i.e., once a strip intersects the D_0 lobe, it stays inside forward iterates of D_0.

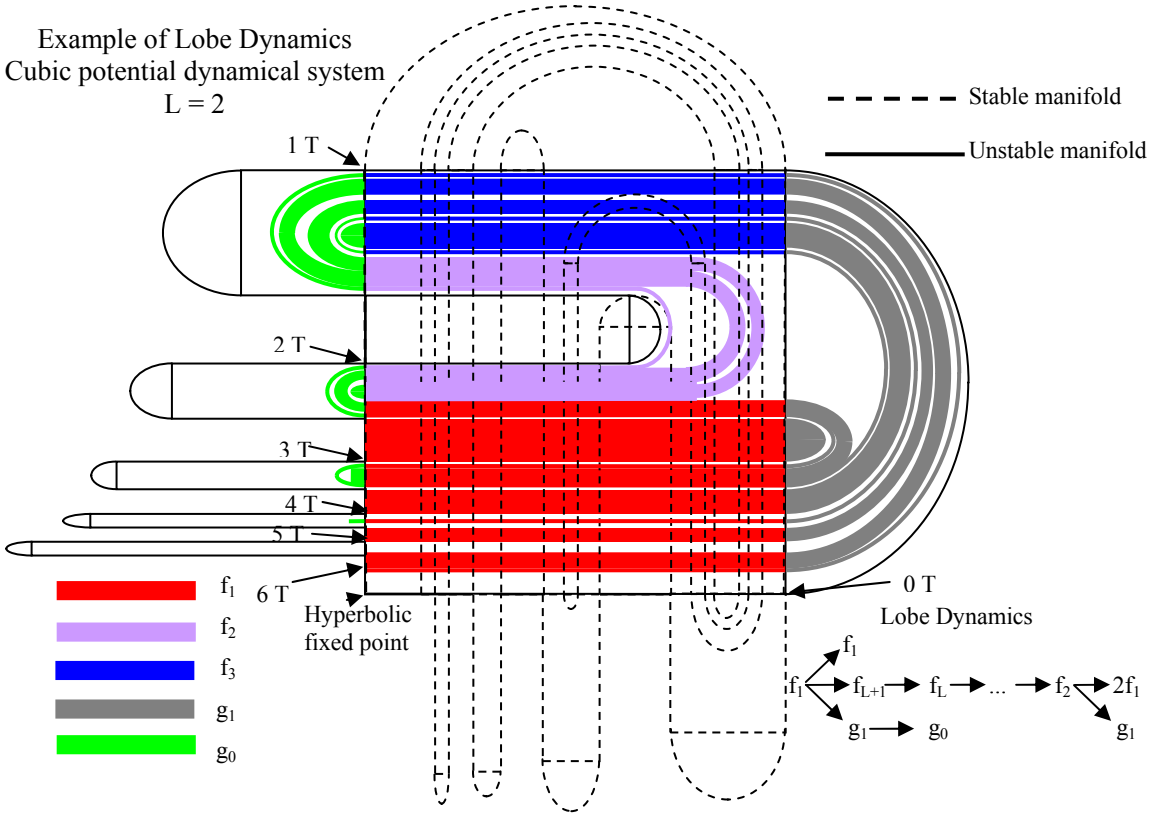

Figure 4.4.1 Example of lobe dynamics, using L = 2 for the cubic potential system

Referring to Figure 4.4.1, the colored strips are labeled according to the definitions given in 1) through 4), and the arrows in the lower right corner of the diagram give the symbolic dynamics, i.e., what happens to a strip upon iteration. For this example, an f_1 strip (red) stretches and folds to create three different strips: a new f_1 (red) strip, an f_{L+1} (in this case, $f_{L+1} = f_3$) strip (blue), and a g ($g = g_1$) strip (grey). An f_{L+1} strip (blue) becomes an f_L (in this case, $f_L = f_2$) strip (purple). An f_2 strip (purple) stretches and folds into two f_1 strips (red) and one g strip (grey). Any intermediate f strips will progressively move under iteration from f_{L+1} to f_2. Note that the symbolic dynamics hold for any value of L. Observe from the diagram that lobe E_2 ends at time 3T, T being the fundamental period of the perturbation frequency, lobe E_3 ends at time 4T, lobe E_4 ends at 5T, and lobe E_5 ends at 6T. Lobe E_2, i.e., lobe E_L for this case, is the first E lobe to intersect lobe D_0 (the lobe on the right that includes all of the grey g strips), and is composed of one f_1 (red) strip and one g (grey) strip. Lobe E_3 (E_{L+1}) begins as a red f_1 strip, has a grey g strip intersecting lobe D_0, then has a blue f_3 strip, and ends with its tip intersecting lobe D_1, the top lobe on the left. The tip of lobe E_4 (E_{L+2}) intersects lobe D_2, after starting as a red f_1 strip, then having a grey g strip intersecting lobe D_0, a blue f_3 strip, wrapping around the tip of lobe E_3 in lobe D_1, and has an f_2 (purple) strip going from lobe D_1 to lobe D_2, as according to strip definition 3). Lobe E_5 (E_{2L+1}) starts as a red f_1 strip, has an outer grey g strip in lobe D_0, a blue f_3 strip, wraps around in lobe D_1, has a purple f_2 strip running from D_1 to D_2, wraps around in lobe D_2, and then has a red f_1 strip running to lobe D_0, a second, inner grey strip in D_0, another red f_1 strip running from lobe D_0 to lobe D_3, and ends in lobe D_3. This sequence encapsulates the lobe dynamics for L = 2, and is generalized to all L. Notice that lobe E_5 (E_{2L+1}) is the first lobe to have two different intersections areas (outer and inner) in lobe D_0. Intermediate lobes between lobe E_L and E_{2L+1} essentially act as a delay before this second intersection is achieved.

The symbolic dynamics can be made more precise by including weighting factors. These weighting factors are necessary whenever a strip is folded and stretched to make several strips, as for f_1 and f_2 strips. Each weighting factor, denoted with a subscripted s, represents what fraction of the original strip becomes a particular resultant strip. Now an f_1 strip will go to an s_1f_1 strip (s_1 representing the fraction of the original f_1 strip that becomes a new f_1 strip), an s_2f_{L+1} strip (s_2 representing the fraction of the original f_1 strip that becomes an f_{L+1} strip), and a $(1-s_1-s_2)$ g strip. An f_2 strip will now become 2 s_3f_1 strips (s_3 representing the fraction of the original f_2 strip that composes one of the two resultant f_1 strips), and a $(1-2s_3)$ g strip.

Figure 4.4.2 shows representative strips for the Kelvin-Stuart Cat Eyes driven flow in the case of L = 2.

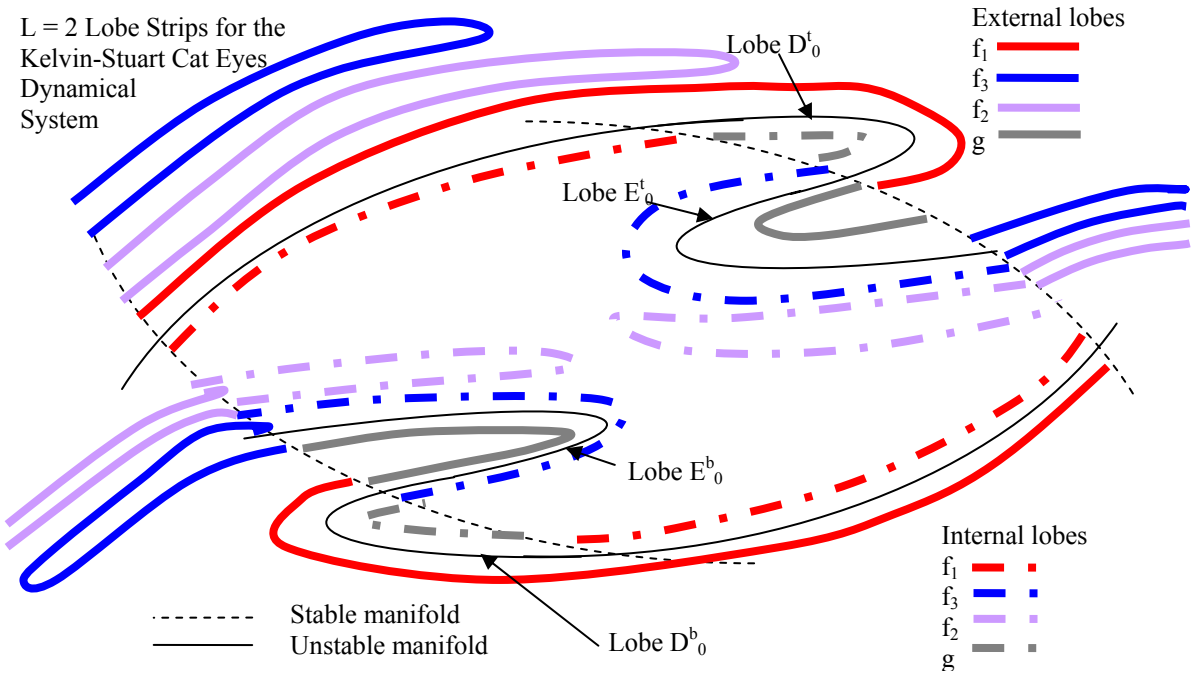

Figure 4.4.2 Representative strips for the Kelvin-Stuart driven flow for L = 2

The strips for the Kelvin-Stuart driven flow follow the dynamics listed below:

External Lobes

1) An f_1^{tt} (red) strip upon iteration becomes an $s_1^{tt}\, f_1^{tt}$ (red) strip, an $s_2^{tt}\, f^{tt}_{L+1}$ (blue) strip, and a $(1 - s_1^{tt} - s_2^{tt})\, g^{tb}$ (grey) strip (tt = external lobe from top to top).

2) An f^{tt}_{L+1} (blue) strip becomes an f^{tt}_{L} strip, which goes through a sequence of intermediate f^{tt} strips until it becomes an f_2^{tt} (purple) strip.

3) An f_2^{tt} (purple) strip generates 2 $s_3^{tt}\, f_1^{tt}$ (red) strips and a $(1 - 2\, s_3^{tt})\, g^{tb}$ (grey) strip.

Similar dynamics exist for bb lobes (bb = external lobe from bottom to bottom). If the system is symmetrical, the tt and bb dynamics are identical.

Internal Lobes

1) An f_1^{tb} (red) strip upon iteration becomes an $s_1^{tb} f_1^{tb}$ (red) strip, an $s_2^{tb} f^{bt}_{L+1}$ (blue) strip and a $(1 - s_1^{tb} - s_2^{tb})$ g^{bb} (grey) strip (tb = internal lobe from top to bottom).

2) An f^{bt}_{L+1} (blue) strip becomes an f^{bt}_{L} strip, which goes through a sequence of intermediate f^{bt} strips until it becomes an f_2^{bt} (purple) strip.

3) An f_2^{bt} (purple) strip generates 2 s_3^{bt} f_1^{bt} (red) strips and a $(1 - 2\, s_3^{bt})$ g^{bb} (grey) strip.

Similar dynamics exist for bt lobes (bt = internal lobe from bottom to top). If the system is symmetrical, the bt and tb dynamics are identical.

The symbolic dynamics can be put into a [4(L+2)] X [4(L+2)] matrix called the lobe dynamics transfer matrix, denoted by M_L. The block structure of M_L is

$$M_L = \begin{array}{c} \\ 4 \\ L+1 \\ L+1 \\ L+1 \\ L+1 \end{array} \begin{array}{c} \begin{array}{ccccc} 4 & L+1 & L+1 & L+1 & L+1 \end{array} \\ \begin{pmatrix} 0 & G^{tt} & G^{tb} & G^{bt} & G^{bb} \\ 0 & F^{tt} & 0 & 0 & 0 \\ 0 & 0 & F_1^{tb} & F_2^{bt} & 0 \\ 0 & 0 & F_2^{tb} & F_1^{bt} & 0 \\ 0 & 0 & 0 & 0 & F^{bb} \end{pmatrix} \end{array} \qquad \text{Eq.4.4.1}$$

The dimensions of each block are written external to the matrix in Eq.4.4.1, so each G block submatrix has dimensions 4 X (L+1), while each of the F block submatrices have dimensions of (L+1) X (L+1). Zero entries in Eq.4.4.1 represent zero block submatrices of the appropriate dimensions.

The four G block submatrices are:

$$G^{tt} = \begin{array}{c} \\ g^{tt} \\ g^{tb} \\ g^{bt} \\ g^{bb} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{tt} & f_2^{tt} & f_3^{tt} & \cdots & f_{L-1}^{tt} & f_L^{tt} & f_{L+1}^{tt} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1-s_1^{tt}-s_2^{tt} & 1-2s_3^{tt} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.2a}$$

$$G^{tb} = \begin{array}{c} \\ g^{tt} \\ g^{tb} \\ g^{bt} \\ g^{bb} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{tb} & f_2^{tb} & f_3^{tb} & \cdots & f_{L-1}^{tb} & f_L^{tb} & f_{L+1}^{tb} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1-s_1^{tb}-s_2^{tb} & 1-2s_3^{tb} & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.2b}$$

$$G^{bt} = \begin{array}{c} \\ g^{tt} \\ g^{tb} \\ g^{bt} \\ g^{bb} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{bt} & f_2^{bt} & f_3^{bt} & \cdots & f_{L-1}^{bt} & f_L^{bt} & f_{L+1}^{bt} \end{array} \\ \begin{pmatrix} 1-s_1^{bt}-s_2^{bt} & 1-2s_3^{bt} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.2c}$$

$$G^{bb} = \begin{array}{c} \\ g^{tt} \\ g^{tb} \\ g^{bt} \\ g^{bb} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{bb} & f_2^{bb} & f_3^{bb} & \cdots & f_{L-1}^{bb} & f_L^{bb} & f_{L+1}^{bb} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1-s_1^{bb}-s_2^{bb} & 1-2s_3^{bb} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.2d}$$

These blocks represent the fraction of strips in the previous iterate that will generate grey *g* strips in the current iterate. The superscript refers to which type of lobe will be generating grey *g* strips of intersection areas. The row labels indicate where the generated *g* strip will be located, while the column labels indicate which previous strip will produce a *g* strip in the current iteration. For example, in the G^{bb} block submatrix, only f_1^{bb} and f_2^{bb} strips (bottom external lobe strips) will generate *g* strips, which will be located in g^{bt} (bottom-to-top internal lobe E^b_0).

Dynamics involving just *f* strips are represented by the square *F* blocks. The external blocks, F^{tt} and F^{bb}, have the same form, with just relabeling of the appropriate superscripts. The form of the F^{tt} block is shown in Eq.4.4.3a. The rows and columns are labeled with the L+1 possible *f* strips, from f_1 to f_{L+1}.

$$F^{tt} = \begin{array}{c} \\ f_1^{tt} \\ f_2^{tt} \\ f_3^{tt} \\ \vdots \\ f_{L-1}^{tt} \\ f_L^{tt} \\ f_{L+1}^{tt} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{tt} & f_2^{tt} & f_3^{tt} & \cdots & f_{L-1}^{tt} & f_L^{tt} & f_{L+1}^{tt} \end{array} \\ \begin{pmatrix} s_1^{tt} & 2s_3^{tt} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ s_2^{tt} & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.3a}$$

The two types of internal lobes (tb and bt) each have two submatrix blocks. The form of each submatrix block is identical for the two internal lobes, with just relabeling of superscripts. Block F_1 has the following form:

$$F_1^{tb} = \begin{array}{c} \\ f_1^{tb} \\ f_2^{tb} \\ f_3^{tb} \\ \vdots \\ f_{L-1}^{tb} \\ f_L^{tb} \\ f_{L+1}^{tb} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{tb} & f_2^{tb} & f_3^{tb} & \cdots & f_{L-1}^{tb} & f_L^{tb} & f_{L+1}^{tb} \end{array} \\ \begin{pmatrix} s_1^{tb} & 2s_3^{tb} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.3b}$$

The F_1 block submatrices represent the sequence of *f* strips for a particular internal lobe, e.g., the sequence of tb lobe strips from f_{L+1}^{tb} to f_1^{tb}. The F_2 block connects the two internal lobe types together, since an f_1^{tb} strip upon iteration will generate an f_{L+1}^{bt} strip, in addition to other strips. The F_2 block submatrices have all zero entries, with one exception:

$$F_2^{tb} = \begin{array}{c} \\ f_1^{bt} \\ f_2^{bt} \\ f_3^{bt} \\ \vdots \\ f_{L-1}^{bt} \\ f_L^{bt} \\ f_{L+1}^{bt} \end{array} \begin{array}{c} \begin{array}{ccccccc} f_1^{tb} & f_2^{tb} & f_3^{tb} & \cdots & f_{L-1}^{tb} & f_L^{tb} & f_{L+1}^{tb} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 2s_2^{tb} & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \text{Eq.4.4.3c}$$

An example of a lobe dynamics transfer matrix, for the case of L = 2, is given in Figure 4.4.3. Block submatrices for external lobes are highlighted in blue, while block submatrices for the internal lobes are highlighted in red.

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-s_1^{bt}-s_2^{bt} & 1-2s_3^{bt} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1-s_1^{tt}-s_2^{tt} & 1-2s_3^{tt} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-s_1^{bb}-s_2^{bb} & 1-2s_3^{bb} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-s_1^{tb}-s_2^{tb} & 1-2s_3^{tb} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_1^{tt} & 2s_3^{tt} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_2^{tt} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & s_1^{tb} & 2s_3^{tb} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2^{bt} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_1^{bt} & 2s_3^{bt} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2^{tb} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_1^{bb} & 2s_3^{bb} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2^{bb} & 0 & 0
\end{pmatrix}$$

Figure 4.4.3 Lobe Dynamics Transfer Matrix for L = 2

The lobe dynamics transfer matrix represents the symbolic dynamics of the system. The matrix operates on vectors which have components that are the areas contained in each type of strip at a particular iteration. The vectors have dimension [4(*L*+2)] X 1 and the following structure:

$$v = \begin{matrix} 4 \\ L+1 \\ L+1 \\ L+1 \\ L+1 \end{matrix} \begin{pmatrix} v^{G} \\ v^{tt} \\ v^{tb} \\ v^{bt} \\ v^{bb} \end{pmatrix} \qquad \text{Eq.4.4.4a}$$

The first four components of v give the lobe intersection areas:

$$v^G = \begin{pmatrix} g^{tt} \\ g^{tb} \\ g^{bt} \\ g^{bb} \end{pmatrix} \qquad \text{Eq.4.4.4b}$$

Each group of (L+1) X 1 components are the areas of the f strips for a particular lobe type, for example, v^{tt} gives the areas of the tt f strips:

$$v^{tt} = \begin{pmatrix} f_1^{tt} \\ f_2^{tt} \\ \vdots \\ f_L^{tt} \\ f_{L+1}^{tt} \end{pmatrix} \qquad \text{Eq.4.4.4c}$$

Vectors will be labeled with a subscript to denote which iteration they represent. The initial vector is labeled with subscript L, since iteration L is the first in which lobes intersect. As an example, the case of L = 2 for the initial vector v_L is shown in Figure 4.4.4.

$$\begin{pmatrix} \textbf{intersection area of internal lobe L} \\ \textbf{intersection area of external lobe L} \\ \textbf{intersection area of external lobe L} \\ \textbf{intersection area of internal lobe L} \\ \textbf{lobe area} - \textbf{intersection area of external lobe L} \\ \textbf{0} \\ \textbf{0} \\ \textbf{lobe area} - \textbf{intersection area of internal lobe L} \\ \textbf{0} \\ \textbf{0} \\ \textbf{lobe area} - \textbf{intersection area of internal lobe L} \\ \textbf{0} \\ \textbf{0} \\ \textbf{lobe area} - \textbf{intersection area of external lobe L} \\ \textbf{0} \\ \textbf{0} \end{pmatrix}$$

Figure 4.4.4 Initial Vector for L = 2

4.5 Lobe Intersection Areas and Weighting Factors

Introduce the following notation for the initial vector, the lobe dynamics transfer matrix, and the n^{th} iteration escape rates vector:

v_L = the initial vector – first four components give the values of areas of lobe intersections for the L^{th} iterate

$v_L(1)$ = intersection area of the L^{th} iterate of the bt internal lobe

$v_L(2)$ = intersection area of the L^{th} iterate of the tt external lobe

$v_L(3)$ = intersection area of the L^{th} iterate of the bb external lobe

$v_L(4)$ = intersection area of the L^{th} iterate of the tb internal lobe

M_L = the lobe dynamics transfer matrix for structure factor L

v_n = the n^{th} iteration escape rates vector – first four components give the values of areas of lobe intersections for the n^{th} iterate

$v_n = (M_L)^{n-L} v_L$

$v_n(1)$ = intersection area of the n^{th} iterate of the bt internal lobe

$v_n(2)$ = intersection area of the n^{th} iterate of the tt external lobe

$v_n(3)$ = intersection area of the n^{th} iterate of the bb external lobe

$v_n(4)$ = intersection area of the n^{th} iterate of the tb internal lobe

The entries of the lobe dynamics transfer matrix depend on the values of the weighting factors, which themselves depend upon the lobe intersection areas. The relationship between the weighting values and the lobe intersection area can be determined by several successive multiplications of the lobe dynamics transfer matrix M_L and the initial vector v_L, as indicated in the definition of the n^{th} iteration escape rates vector $v_n = (M_L)^{n-L} v_L$. For n = L+1, $v_{L+1} = M_L v_L$. Performing the multiplication results in the following values for $v_{L+1}(i)$, i = 1, 2, 3, or 4.

$$v_{L+1}(1) = (1 - s_1^{bt} - s_2^{bt}) v_L(5 + 2(L+1)), \quad v_{L+1}(2) = (1 - s_1^{tt} - s_2^{tt}) v_L(5),$$
$$v_{L+1}(3) = (1 - s_1^{bb} - s_2^{bb}) v_L(5 + 3(L+1)), \quad v_{L+1}(4) = (1 - s_1^{tb} - s_2^{tb}) v_L(5 + (L+1)) \qquad \text{Eq.4.5.1}$$

For n = L+2, $v_{L+2} = M_L^2 v_L$. Performing the multiplication results in the following values for $v_{L+2}(i)$, i = 1, 2, 3, or 4.

$$v_{L+2}(1) = (1 - s_1^{bt} - s_2^{bt}) s_1^{bt} v_L(5 + 2(L+1)), \quad v_{L+2}(2) = (1 - s_1^{tt} - s_2^{tt}) s_1^{tt} v_L(5),$$
$$v_{L+2}(3) = (1 - s_1^{bb} - s_2^{bb}) s_1^{bb} v_L(5 + 3(L+1)), \quad v_{L+2}(4) = (1 - s_1^{tb} - s_2^{tb}) s_1^{tb} v_L(5 + (L+1)) \qquad \text{Eq.4.5.2}$$

For n = 2L+1, $v_{2L+1} = M_L^{L+1} v_L$. Performing the multiplication results in the following values for $v_{2L+1}(i)$, i = 1, 2, 3, or 4.

$$\begin{aligned}
v_{2L+1}(1) &= (1-s_1^{bt}-s_2^{bt})(s_1^{bt})^L v_L(5+2(L+1)) + (1-2s_3^{bt})\, s_2^{tb} v_L(5+(L+1)),\\
v_{2L+1}(2) &= (1-s_1^{tt}-s_2^{tt})(s_1^{tt})^L v_L(5) + (1-2s_3^{tt})\, s_2^{tt} v_L(5),\\
v_{2L+1}(3) &= (1-s_1^{bb}-s_2^{bb})(s_1^{bb})^L v_L(5+3(L+1)) + (1-2s_3^{bb})\, s_2^{bb} v_L(5+3(L+1)),\\
v_{2L+1}(4) &= (1-s_1^{tb}-s_2^{tb})(s_1^{tb})^L v_L(5+(L+1)) + (1-2s_3^{tb})\, s_2^{bt} v_L(5+2(L+1))
\end{aligned} \qquad \text{Eq.4.5.3}$$

The switching of the superscipts and The different integer multiplying (*L*+1) in the second terms of the first and fourth expressions, and the switching of the superscipts, reflect the nature of the internal lobes as they move from the bottom to the top and back to the bottom again (or vice versa). Examining Eqs.4.5.1, 2 leads to the following expressions for the s_1 and s_2 terms:

$$\begin{aligned}
s_1^{bt} &= \frac{v_{L+2}(1)}{v_{L+1}(1)}, \quad s_1^{tt} = \frac{v_{L+2}(2)}{v_{L+1}(2)}, \quad s_1^{bb} = \frac{v_{L+2}(3)}{v_{L+1}(3)}, \quad s_1^{tb} = \frac{v_{L+2}(4)}{v_{L+1}(4)}\\
s_2^{bt} &= 1 - s_1^{bt} - \frac{v_{L+1}(1)}{v_L(5+2(L+1))}, \quad s_2^{tt} = 1 - s_1^{tt} - \frac{v_{L+1}(2)}{v_L(5)},\\
s_2^{bb} &= 1 - s_1^{bb} - \frac{v_{L+1}(3)}{v_L(5+3(L+1))}, \quad s_2^{tb} = 1 - s_1^{tb} - \frac{v_{L+1}(4)}{v_L(5+(L+1))}
\end{aligned} \qquad \text{Eq.4.5.4}$$

Eqs.4.5.3, 4 allow the following expressions for the s_3 terms to be derived:

$$\begin{aligned}
s_3^{bt} &= \frac{1}{2} + \frac{(s_1^{bt})^L v_{L+1}(1) - v_{2L+1}(1)}{2 s_2^{tb} v_L(5+(L+1))}, \quad s_3^{tt} = \frac{1}{2} + \frac{(s_1^{tt})^L v_{L+1}(2) - v_{2L+1}(2)}{2 s_2^{tt} v_L(5)},\\
s_3^{bb} &= \frac{1}{2} + \frac{(s_1^{bb})^L v_{L+1}(3) - v_{2L+1}(3)}{2 s_2^{bb} v_L(5+3(L+1))}, \quad s_3^{tb} = \frac{1}{2} + \frac{(s_1^{tb})^L v_{L+1}(4) - v_{2L+1}(4)}{2 s_2^{bt} v_L(5+2(L+1))}
\end{aligned} \qquad \text{Eq.4.5.5}$$

Formulas for all of the weighting factors have been found in terms of lobe intersection areas. Note that these formulas are valid for L ≥ 2. For L = 0 or L = 1, the method for finding the expressions in Eqs.4.5.4, 5 will not work, because the equations can only be solved for linear combinations of weighting factors, not the individual weighting factors. Some other means of determining the weighting factors must be found for L = 0 and L = 1.

The total area transported at iteration n for a particular lobe type is found by summing the v_i vectors from L to n, i.e., using the tt external lobe as an example,

$$c^{tt}_n = \text{tt lobe area} - \sum_{i=L}^{n} v_i(2) \qquad \text{Eq.4.5.6}$$

Similar results hold for other lobes. The cumulative area transported by iteration n is

$$\sum_{i=0}^{n} c^{tt}_n \qquad \text{Eq.4.5.7}$$

Key to calculating the weighting factors and the transport rates is finding the areas of lobe intersections. Zeroes of Whisker map iterates are used to find the areas of lobe intersections. Essentially, an iterate of the Whisker map is integrated between its zero locations. The following formulas give various lobe and lobe intersections areas:

$$\text{Area of a lobe} = \left| \int_0^{\pi/\omega} M(t)\, dt \right| \qquad \text{Eq.4.5.8}$$

$$\text{Area of intersection for lobe L} = \left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt \right| \qquad \text{Eq.4.5.9}$$

$$\text{Area of intersection for lobe L + 1} = \left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt + \int_{t_0^3}^{t_0^4} \varepsilon\, h_2(t)\, dt \right| \qquad \text{Eq.4.5.10}$$

$$\text{Area of intersection for lobe L + 2} = \left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt + \int_{t_0^3}^{t_0^4} \varepsilon\, h_2(t)\, dt \right| \qquad \text{Eq.4.5.11}$$

Area of intersection for lobe 2L + 1 =

$$\left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt + \int_{t_0^3}^{t_0^4} \varepsilon\, h_2(t)\, dt \right| + \left| \int_{\tau_0^1}^{\tau_0^2} \varepsilon\, h_3(t)\, dt + \int_{\tau_0^3}^{\tau_0^4} \varepsilon\, h_3(t)\, dt \right| \qquad \text{Eq.4.5.12}$$

To find zeroes for lobes L, L + 1, L + 2, and the outer strip of lobe 2L + 1, solve the following for t_0:

$$\begin{aligned} & h_2 = M(t_0) + M(t_1) = 0 \\ & t_1 = t_0 + \begin{cases} \frac{1}{2} P_{\text{int}}(\varepsilon M(t_0)), & M(t_0) < 0 \\ P_{ext}(\varepsilon M(t_0)), & M(t_0) > 0 \end{cases} \\ & jT < t_1 < (j+1)T, \quad T = \tfrac{2\pi}{\omega} \\ & j = L, L+1, L+2, 2L+1 \end{aligned} \qquad \text{Eq.4.5.13}$$

To find zeroes for the inner strip of lobe 2L + 1 solve for t_0 satisfying the following conditions:

$$\begin{aligned} & h_3 = M(t_0) + M(t_1) + M(t_2) = 0 \\ & t_1 = t_0 + \begin{cases} \frac{1}{2} P_{\text{int}}(\varepsilon M(t_0)), & M(t_0) < 0 \\ P_{ext}(\varepsilon M(t_0)), & M(t_0) > 0 \end{cases} \\ & LT < t_1 < (L+1)T, \quad T = \tfrac{2\pi}{\omega} \\ & t_2 = t_1 + \begin{cases} \frac{1}{2} P_{\text{int}}(\varepsilon h_2), & h_2 < 0 \\ P_{ext}(\varepsilon h_2, & h_2 > 0 \end{cases} \\ & (2L+1)T < t_2 < (2L+2)T, \quad T = \tfrac{2\pi}{\omega} \end{aligned} \qquad \text{Eq.4.5.14}$$

Chapter 5
Application of the Topological Approximation Method to the Kelvin-Stuart Cat Eyes Driven Flow

5.1 Kelvin-Stuart Cat Eyes Driven Flow Bifurcation Curves

The method for calculating bifurcation curves detailed in section 4.3 was used to develop a Mathematica® module (with invaluable assistance from Ruskeepää 2004 and Wolfram 1999) to calculate the approximate and exact values for the minimum and maximum bifurcation curves for the Kelvin-Stuart cat eyes driven flow. The module (found in Appendix B.1) calculates six different bifurcation curves: the approximate minimum (shown in green) and approximate maximum (magenta), the exact minimum (red) and maximum (orange) bifurcation curves for internal lobes, and the exact minimum (dark blue) and maximum (light blue) bifurcation curves for external lobes. Sets of these six bifurcation curves can be calculated for any value of L. The results for L = 0 to L = 10, from left to right, are shown in Figure 5.1.1

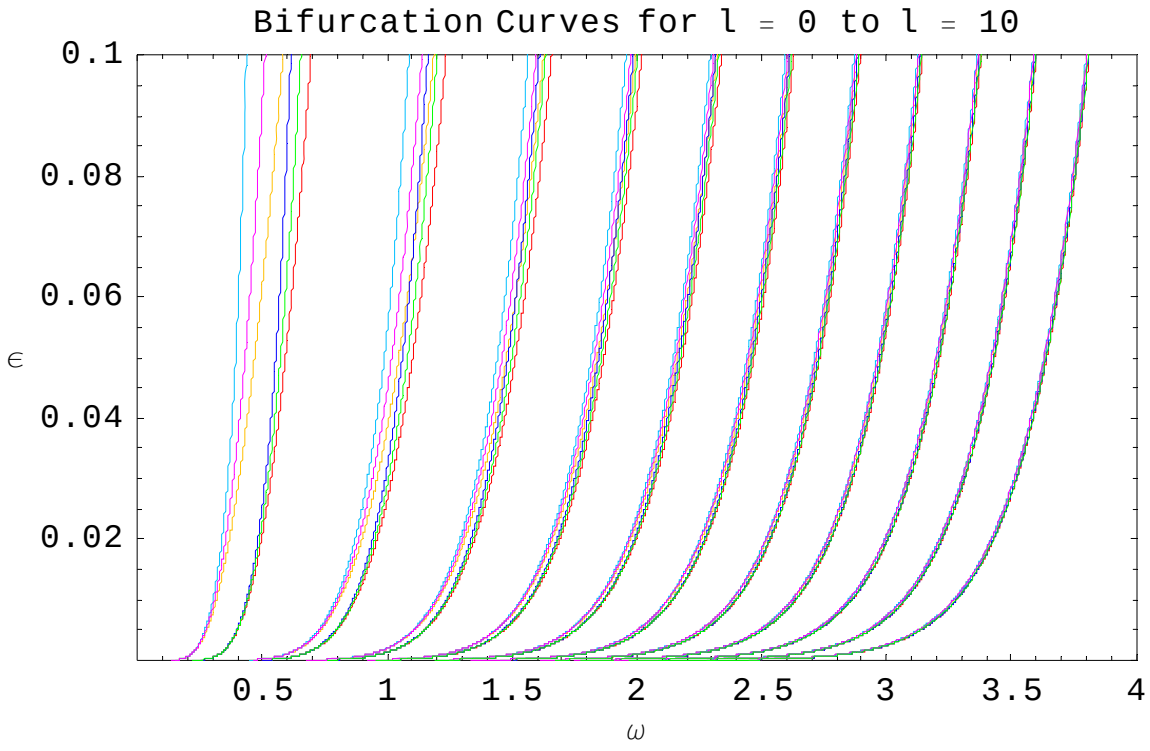

Figure 5.1.1 Kelvin-Stuart Cat Eyes Bifurcation Curves for L = 0 to L = 10

Given the differences in bifurcation curves for internal and external lobes, different lobe intersection areas should be expected for the internal and external lobes, with this difference more pronounced for smaller frequencies and decreasing with increasing frequency.

The bifurcation curves are given as ε in terms of functions of ω, but more useful for investigating the frequency dependence of the transport rates is to look at bifurcations as ω varies with constant ε. As ε and L increase, the internal and external bifurcation curves separate so that no overlap in frequency occurs, indicating that one set of lobes (internal or external) will go through a complete set of bifurcations (0 to 1 to 2 to 3 to 4 intersections points) while the other set of lobes is still at 0 intersection points. A common frequency, with constant ε, for which both the internal and external lobes undergo bifurcations, is desired. To achieve this, a value of ε = 0.01 was chosen so that a relatively large frequency overlap range existed for each L value. Even with this choice of ε large gaps exist between L value bifurcation curve sets. The Topological Approximation Method can only be applied between the minimum and maximum bifurcation curves for any value of L (Rom-Kedar 1994), so the TAM cannot be applied to these gap regions of the ε-ω space.

Figure 5.1.2 zooms in on the L = 2 bifurcation curves shown in Figure 5.1.1, focusing on the ω interval from 1.0 to 1.5 and the ε interval from 0.005 to 0.015.

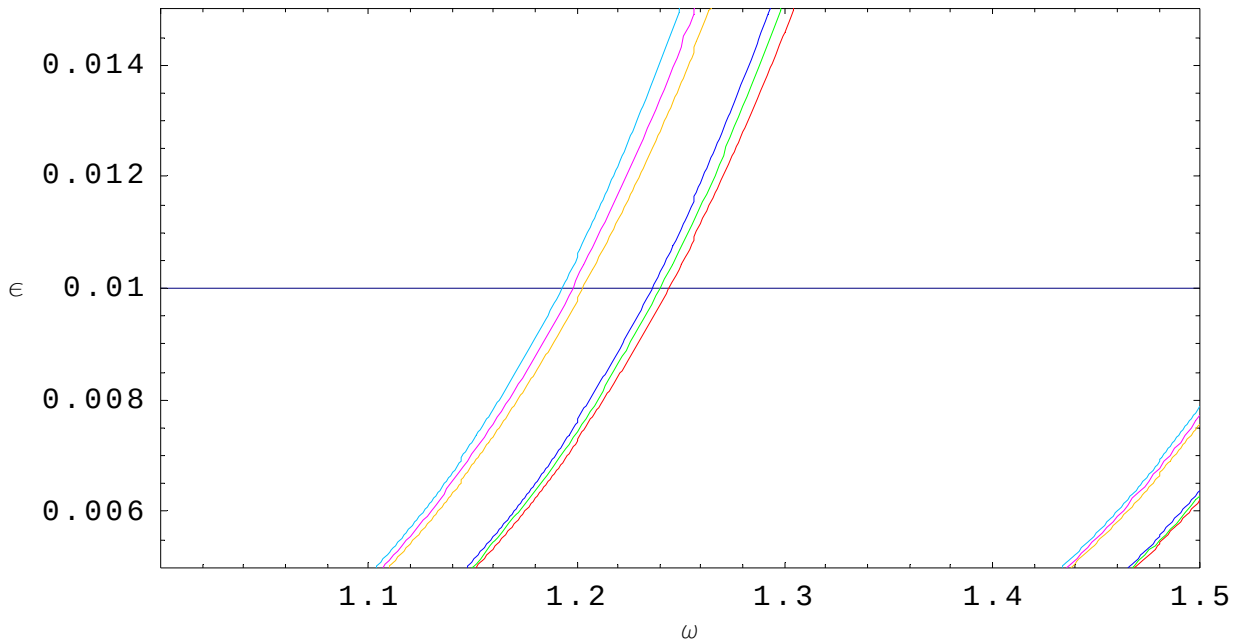

Figure 5.1.2 Zoom-in on L = 2 Bifurcation Curve

Figure 5.1.3 shows what happens as ω varies while ε is held constant at 0.01, the horizontal line in the middle of the Figure 5.1.2. The ω values start at 1.23619, which lies on the dark blue curve (the minimum for the external lobes) of Figure 5.1.2, decreases to 1.21971, which is located halfway between the dark blue and orange curve (the maximum for the internal lobes), decreases further to 1.20323 on the orange curve, and then decreases to 1.18987, to the left of the orange curve and to the right of the light blue curve (maximum for external lobes). In Figure 5.1.3, external lobes lie above the 0 axis, and internal lobes lie below. The green curve represents the Melnikov function itself, while the blue curves are the first iterates of the Whisker map. Look carefully at the intersections of the lobes in each case: ω = 1.23619 - one intersection for the external and two for the internal, ω = 1.21971 - two intersections for both the external and the internal, ω = 1.20323 - two intersections for the external and three for the internal, ω = 1.18987 - two intersections for the external and four for the internal.

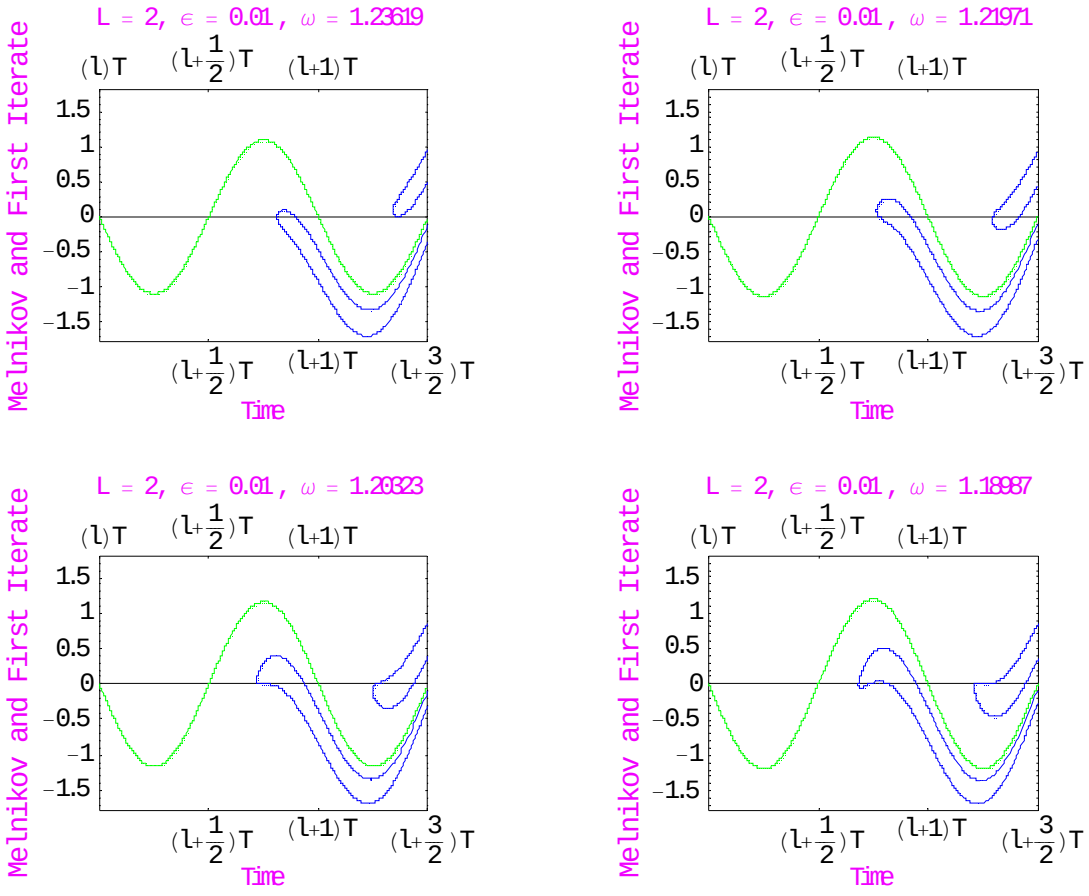

Figure 5.1.3 Variation of ω, with L = 2 and ε = 0.01

5.2 Lobe Intersection Points

A Mathematica® program (found in Appendix B.2), actually one large individual programming cell, was written to calculate all of the values listed in sections 4.4 and 4.5, starting with finding the lobe intersection points, using Eqs.4.5.13, 14:

To find zeroes for lobes L, L + 1, L + 2, and the outer strip of lobe 2L + 1, solve the following for t_0:

$$h_2 = M(t_0) + M(t_1) = 0$$
$$t_1 = t_0 + \begin{cases} \frac{1}{2} P_{\text{int}}(\varepsilon M(t_0), & M(t_0) < 0 \\ P_{ext}(\varepsilon M(t_0), & M(t_0) > 0 \end{cases} \qquad \text{Eq.4.5.13}$$
$$jT < t_1 < (j+1)T, \quad T = \frac{2\pi}{\omega}$$
$$j = L, L+1, L+2, 2L+1$$

To find zeroes for the inner strip of lobe 2L + 1 solve for t_0 satisfying the following conditions:

$$h_3 = M(t_0) + M(t_1) + M(t_2) = 0$$
$$t_1 = t_0 + \begin{cases} \frac{1}{2} P_{\text{int}}(\varepsilon M(t_0), & M(t_0) < 0 \\ P_{ext}(\varepsilon M(t_0), & M(t_0) > 0 \end{cases}$$
$$LT < t_1 < (L+1)T, \quad T = \frac{2\pi}{\omega} \qquad \text{Eq.4.5.14}$$
$$t_2 = t_1 + \begin{cases} \frac{1}{2} P_{\text{int}}(\varepsilon h_2), & h_2 < 0 \\ P_{ext}(\varepsilon h_2, & h_2 > 0 \end{cases}$$
$$(2L+1)T < t_2 < (2L+2)T, \quad T = \frac{2\pi}{\omega}$$

For each value of L, a frequency ω is selected between the minimum bifurcation curve for the external lobes (dark blue curve in Figure 5.1.1) and the maximum internal lobe bifurcation curve (orange curve in Figure 5.1.1). The module allows any choice between these two frequency extremes, but for the present research the frequency at the middle of the range is chosen. Once the frequency is determined, the Mathematica® root finding command, FindRoot, is used to solve for the various zeroes listed in Eqs.4.5.13, 14. Initial values for t_0 were generally selected quite broadly, since precision in choosing the initial t_1 values seemed to be more important for ensuring finding accurate zero values. The conditions on t_1 found in Eqs.4.5.13, 14 were sufficient to provide good estimates for initial values for t_1. In Figures 5.2.1, 5.2.2, and 5.2.3 on the following pages, vertical lines are shown at zeroes calculated by the module. The red curve is the Melnikov function (Whisker map iterate h_1), and the blue curve is the Whisker map iterate

h_2. Results are shown for internal lobes only in Figures 5.2.1 and 5.2.2. Lobe intersection areas are the sums of the areas above the central axis for each lobe. Zeroes are ordered from left to right as t_0^1 through t_0^4, as in Figure 3.2.2.

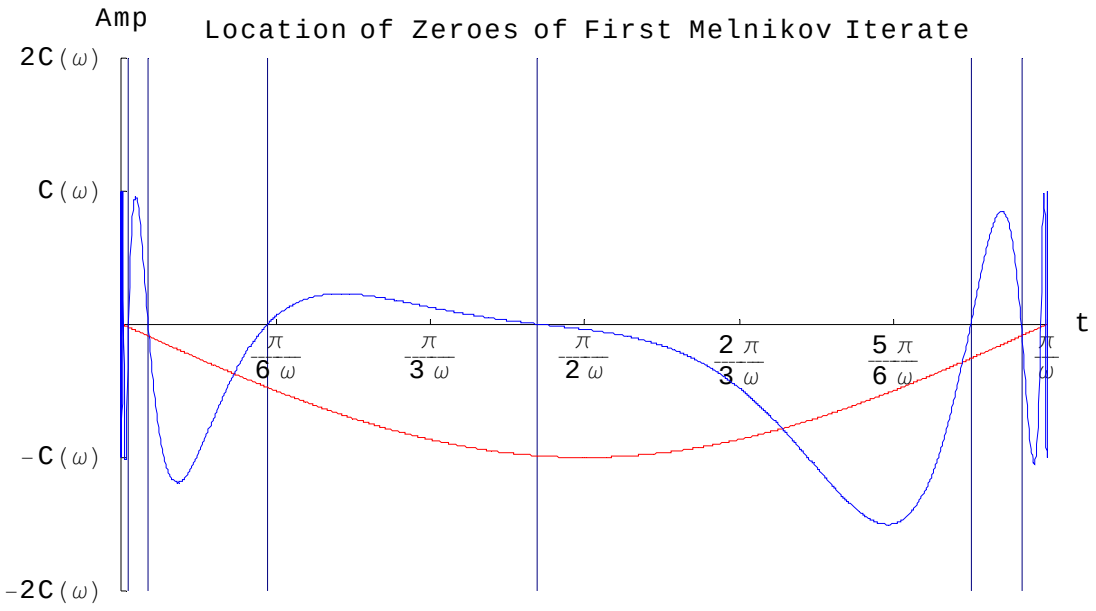

Figure 5.2.1 Intersection Points for Lobes L and L + 1 for L = 2, ε = 0.01, ω = 1.21971

Figure 5.2.2 zooms in on the L+1 lobe intersections (the L+2 intersections can also be seen).

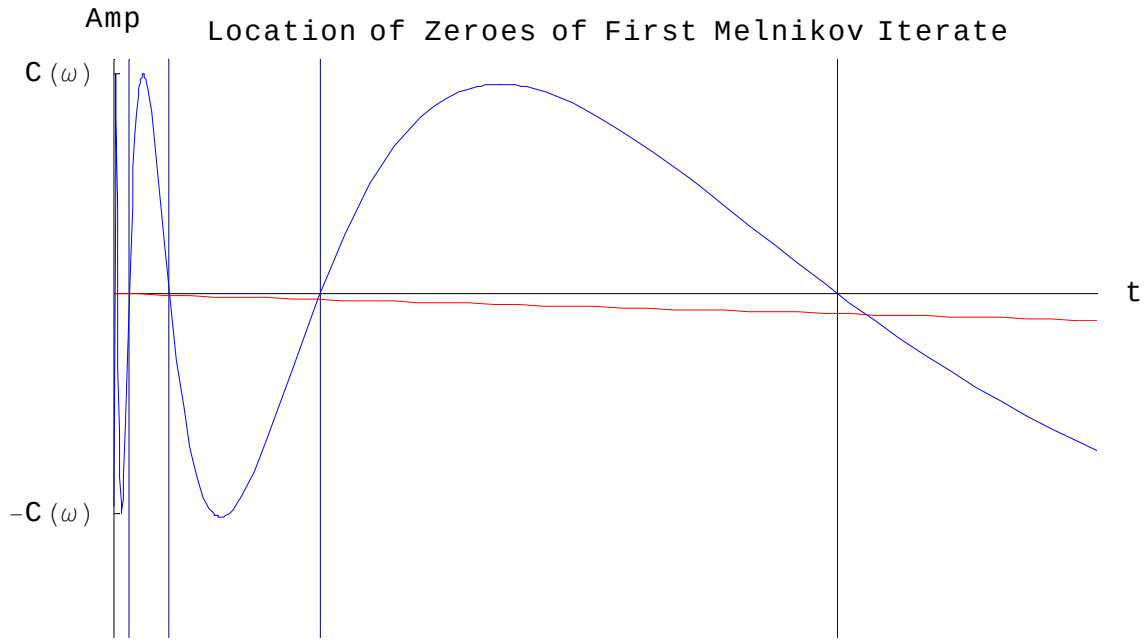

Figure 5.2.2 Zoom-in on Lobe L + 1 intersections for L = 2, ε = 0.01, ω = 1.21971

Figures 5.2.1 and 5.2.2 indicate how accurately the module pinpoints the intersection points. The module also finds the lobe 2L+1 intersections as accurately, even for L = 10.

Figure 5.2.3 graphs a parametric plot of the lobes, using the Whisker map iterates and the values of t_1 rather than those of t_0, which are used in Figures 5.2.1 and 5.2.2. The vertical lines represent the locations of the t_1 calculated values. The green central curve represents the original lobes, and the blue curves represent the iterated lobes as they stretch back to intersect the original lobes. The external portions of lobes are above the 0 central axis, and the internal portions of lobes are below. The first set of four vertical lines is for the internal lobe L+1 below the green curve (observe that this portion is above the central axis and thus is now external), and the second set of four vertical lines is for the external lobe above the green curve (this portion is below the central axis, and is thus now internal). Areas of lobe intersections are the areas inside the vertical lines, between the blue curves and the central axis. The blue curve closest to the green curve uses the outermost vertical lines, while the blue curve further from the green uses the innermost vertical lines. The area of the second is subtracted from the first. The time ordering is that of the t_1 times found in Figure 3.2.2, with t_1^2 first, followed by t_1^3, then t_1^4, and last, t_1^1. The area intersections found here using t_1 times equal those found by using the t_0 times.

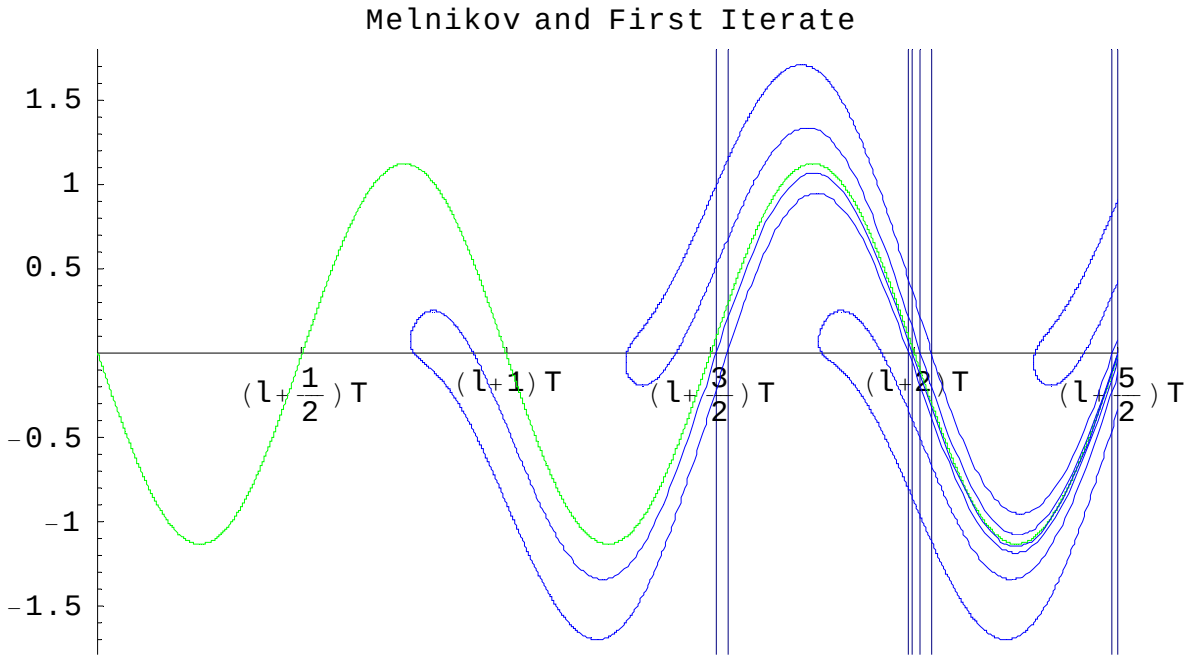

Figure 5.2.3 Parametric Plot of Lobes L and L + 1 for L = 2, ε = 0.01, ω = 1.21971

The inner strip of the 2L+1 lobe presents special difficulties – there are many zeroes of the h_3 Whisker map iterate between consecutive zeroes of the h_2 Whisker map iterate. How does one choose the correct set of zeroes for any value of L? Figure 5.2.4 shows the Melnikov integral (h_1 Whisker map iterate) in red and the h_2 (blue) and h_3 (black) Whisker map iterates. The vertical lines indicate the zeroes of the h_3 Whisker map iterate calculated by the module. As can be seen, these values accurately match one of the possible h_3 Whisker map iterate lobes, but how is it known that this pair of possible lobes (left and right) are the correct pair for the intersection points of the inner strip of lobe 2L+1?

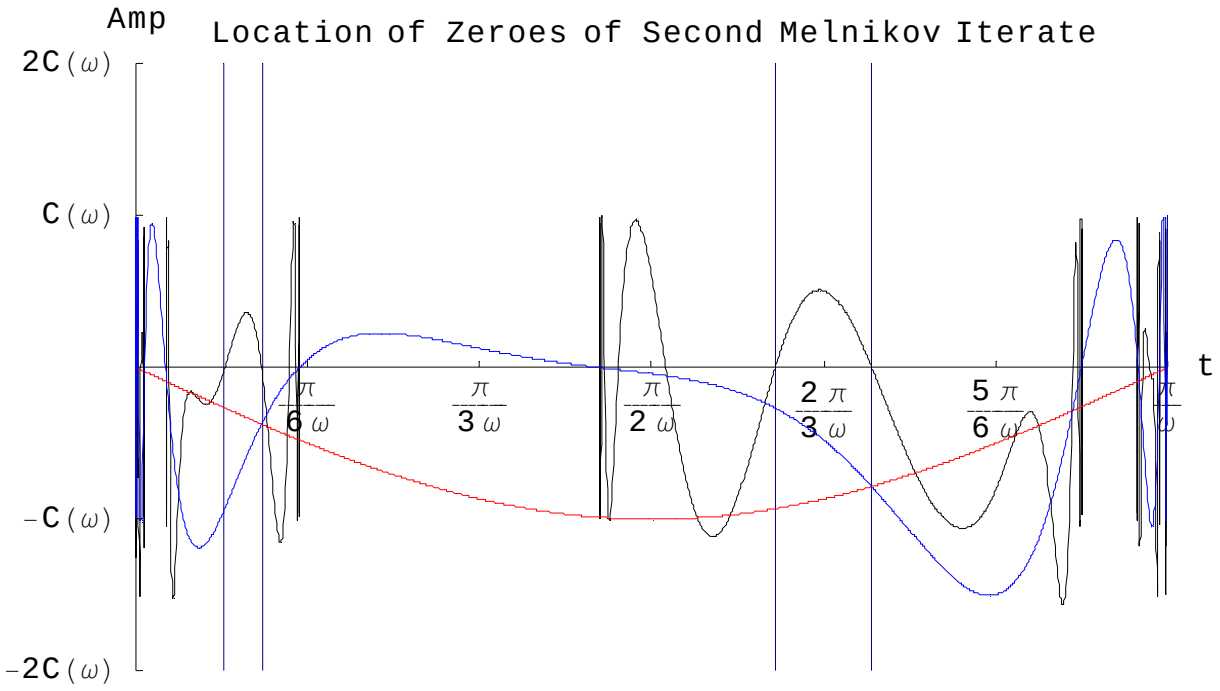

Figure 5.2.4 Intersection Points for the inner part of lobe 2L + 1,
for L = 2, ε = 0.01, ω = 1.21971

Infinitely many possible lobes and intersection points for the h_3 Whisker map iterate exist between consecutive zeroes of the h_2 Whisker map iterate, so being able to accurately target the correct intersection points in the initial estimate is vital. The condition on t_1 found in Eq.4.5.14 is the most important criterion, followed by the condition on t_2. This is demonstrated in Figure 5.2.5, which zooms in on the pair of intersection points found on the right in Figure 5.2.4. Extra curves which indicate the conditions on t_1 and t_2 have been included to demonstrate how these conditions can be used to identify the correct pair of t_2 intersection points.

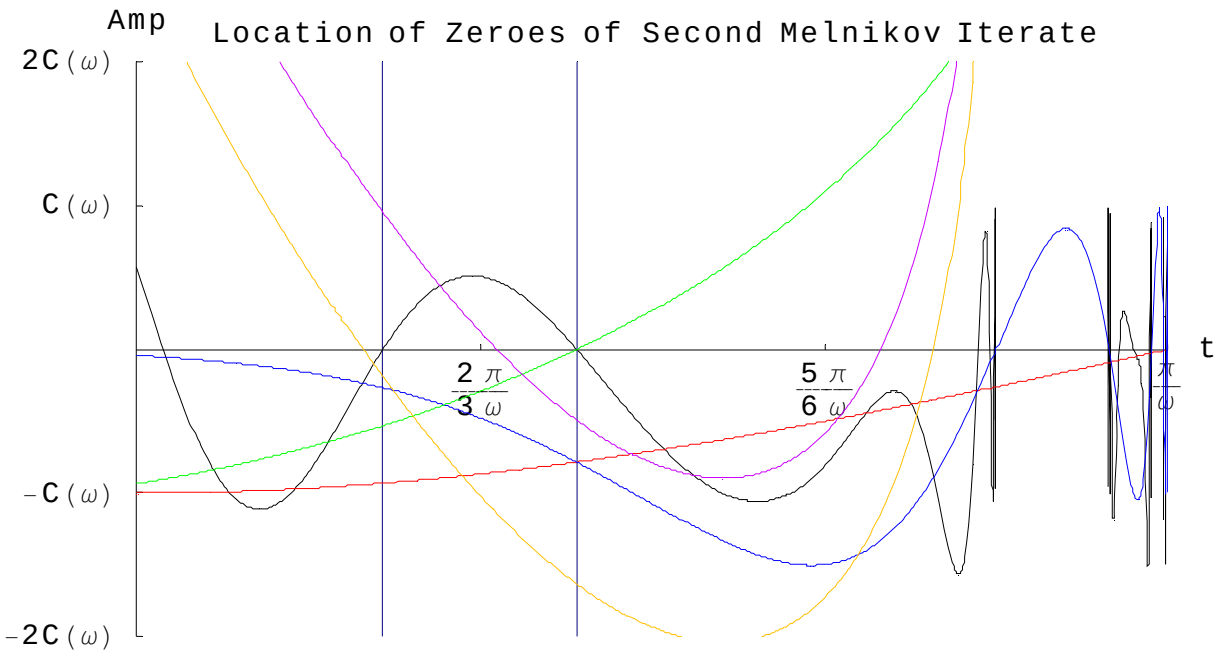

Figure 5.2.5 Zoom-in the inner part of Lobe 2L + 1 intersections
for L = 2, ε = 0.01, ω = 1.21971

The green curve in Figure 5.2.5 indicates points where t_1 = (L+1)T, with smaller values for t_1 located above the green curve and larger values below. Only one pair of intersection points for the h_3 Whisker map iterate has an intersection point with t_1 = (L+1)T, so this condition on t_1 accurately selects the correct choice of intersection points. The green curve also goes through the left point in the left pair of correct intersection points for the h_3 Whisker map iterate shown on Figure 5.2.4, so this condition is sufficient for finding all of the correct intersection points for the inner strip of lobe 2L+1. The purple and orange curves indicate values of t_2 = 2L+7/4 and 2L+2, respectively.

Figures 5.2.4 and 5.2.5 refer to internal lobes only. The lobe intersection area for the inner part of lobe 2L+1 is the sum of the two correct areas above the central axis. Zeroes are ordered from left to right as t_0^1 through t_0^4, as in Figure 3.2.2.

Figure 5.2.6 graphs a parametric plot of the lobes, using the Whisker map iterates and the values of t_2 rather than those of t_0 or t_1. The vertical lines represent the locations of the t_2 calculated values. The green central curve represents the original lobes, the blue curves represent the iterated lobes (L, L+1, L+2, and the outer strip of 2L+1) as they stretch back to

intersect the original lobes. The red curves represent the inner portion of lobe 2L+1 as it winds around the tip of lobe L. The external portions of lobes are above the 0 central axis, and the internal portions of lobes are below. The first set of four vertical lines is for the inner part (red) of internal lobe 2L+1 below the green curve (observe that this portion is above the central axis and thus is now external), and the second set of four vertical lines is for the inner part of the external lobe 2L+1 above the green curve (this portion is below the central axis, and is thus now internal). Areas of lobe intersections are the areas inside the vertical lines, between the red curves and the central axis. The red curve closest to the green curve uses the outermost vertical lines, while the red curve further from the green uses the innermost vertical lines. The area of the second is subtracted from the first. The time ordering is t_2^4 first, followed by t_2^1, then t_2^2, and last, t_2^3. The area intersections found here using t_2 times equal those found by using the t_0 times.

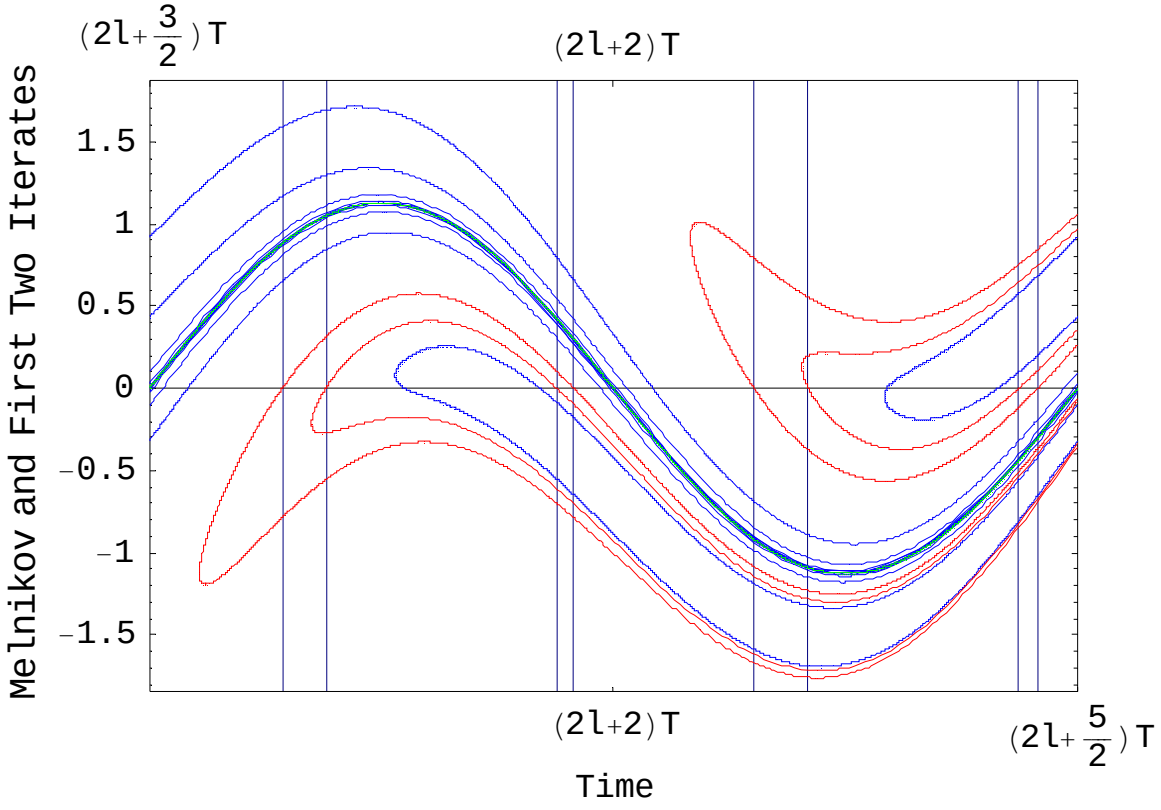

Figure 5.2.6 Parametric Plot of Lobes L, L + 1, L + 2, and 2L + 1,
for L = 2, ε = 0.01, ω = 1.21971

5.3 Lobe Dynamics Transfer Matrix and Initial Vector for L = 2

Finding all of the relevant lobe intersection points allows Eqs.4.5.8-12 to be used to calculate the areas needed to determine the initial vector and the lobe dynamics transfer matrix.

$$\text{Area of a lobe} = \left| \int_0^{\pi/\omega} M(t)\, dt \right| \qquad \text{Eq.4.5.8}$$

$$\text{Area of intersection for lobe L} = \left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt \right| \qquad \text{Eq.4.5.9}$$

$$\text{Area of intersection for lobe L + 1} = \left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt + \int_{t_0^3}^{t_0^4} \varepsilon\, h_2(t)\, dt \right| \qquad \text{Eq.4.5.10}$$

$$\text{Area of intersection for lobe L + 2} = \left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt + \int_{t_0^3}^{t_0^4} \varepsilon\, h_2(t)\, dt \right| \qquad \text{Eq.4.5.11}$$

Area of intersection for lobe 2L + 1 =

$$\left| \int_{t_0^1}^{t_0^2} \varepsilon\, h_2(t)\, dt + \int_{t_0^3}^{t_0^4} \varepsilon\, h_2(t)\, dt \right| + \left| \int_{\tau_0^1}^{\tau_0^2} \varepsilon\, h_3(t)\, dt + \int_{\tau_0^3}^{\tau_0^4} \varepsilon\, h_3(t)\, dt \right| \qquad \text{Eq.4.5.12}$$

Only the values of the areas in Eqs.4.5.8, and 9 are needed for the initial vector v_L, since the first four components of v_L are just the values of the intersection areas for lobe L (found using Eq.4.5.9) for each of four different lobe types, and the only other nonzero values of the initial vector are the area of a lobe (Eq.4.5.8), which is equal for all four lobe types, minus the L lobe intersection areas. For L = 2, ε = 0.01, ω = 1.21971 the initial vector is calculated to be

$$\begin{pmatrix} 0.00108168 \\ 0.00070892 \\ 0.00070892 \\ 0.00108168 \\ 0.0177769 \\ 0 \\ 0 \\ 0.0174042 \\ 0 \\ 0 \\ 0.0174042 \\ 0 \\ 0 \\ 0.0177769 \\ 0 \\ 0 \end{pmatrix}$$

Figure 5.3.1 Calculated Initial Vector for L = 2, ε = 0.01, ω = 1.21971

The area of a lobe for L = 2, ε = 0.01, ω = 1.21971 is 0.01848588, so the intersection areas for internal lobes (the first and fourth components in the initial vector), 0.00108168, only constitutes 5.85% of the area of an internal lobe, while for external lobes (the second and third components of the initial vector), the intersection area of 0.00070892 is only 3.835% of the area of an external lobe. These results provide some evidence justifying the choice to calculate only first order intersections and ignore second and higher order intersections, since higher order intersections would most likely be powers of the above percentages, and thus much smaller.

The additional areas calculated by Eqs.4.5.10-12 allow the weighting factors, and thus the lobe dynamics transfer matrix, to be determined. The lobe dynamics transfer matrix for L = 2, ε = 0.01, ω = 1.21971 is

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0692 & 0.0795 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.069 & 0.0913 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.069 & 0.0913 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0692 & 0.0795 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.105 & 0.909 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.826 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.106 & 0.921 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.824 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.106 & 0.921 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.824 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.105 & 0.909 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.826 & 0 & 0
\end{pmatrix}$$

Figure 5.3.2 Calculated Lobe Dynamics Transfer Matrix for L = 2, ε = 0.01, ω = 1.21971

The entries in the first four rows represent the fraction of strips that end up in intersection areas. Note that all of these values are less than 0.1, so second and higher order intersections would most likely be less than 0.01. The weighting factors are found at the other non-zero, non-unity values in the matrix. Looking at the lower right corner, for external lobes the s_1 weighting factor is 0.105, the s_2 weighting factor is 0.826, and the s_3 weighting factor is ½ (0.909) = 0.4545. The weighting factors for the internal lobes are found at the center of the matrix, and are $s_1 = 0.106$, $s_2 = 0.824$, and $s_3 = ½ (0.921) = 0.4605$.

The lobe dynamics transfer matrix is observed to be quite sparse, and becomes even sparser as L increases, with only four new nonzero entries added with each increase in L. Powers of the lobe dynamics transfer matrix thus can be calculated quite quickly, even when L = 10 and the matrix enlarges to a size of 48 X 48.

Once the initial vector v_L and the lobe dynamics transfer matrix M_L have been established, the escape rate vectors v_n can be calculated from $v_n = (M_L)^{n-L} v_L$. The total area transported at iteration n for a particular lobe type is found by summing the v_i vectors from L to n, i.e., using the tt external lobe as an example,

$$c^{tt}_n = \text{tt lobe area} - \sum_{i=L}^{n} v_i(2) \qquad \text{Eq.5.3.1}$$

Similar results hold for other lobes. The cumulative area transported by iteration n is

$$\sum_{i=0}^{n} c^{tt}_n \qquad \text{Eq.5.3.2}$$

Chapter 6
Results, Conclusions, and Further Research

6.1 Transport Rates for L = 2 Through L = 10

The transport rates for each iteration, for 100 iterations, were calculated using the program in Appendix B.2, for L values from L = 2 through L = 10. For each L value, ε was held constant at $\varepsilon = 0.01$. Frequency values were chosen at the middle value of the frequency interval bracketed by the frequency of the minimum internal bifurcation curve and the frequency of the maximum external bifurcation curve. The frequency values are summarized in Figure 6.1.1.

L	ω
2	1.21971
3	1.55572
4	1.85766
5	2.13393
6	2.39005
7	2.62982
8	2.85599
9	3.07063
10	3.27532

Figure 6.1.1 Table of ω values for L = 2 to L = 10

Graphs of the area transferred per iteration, and the cumulative area transferred by iteration for L = 2 through L = 10 are given on the next nine pages.

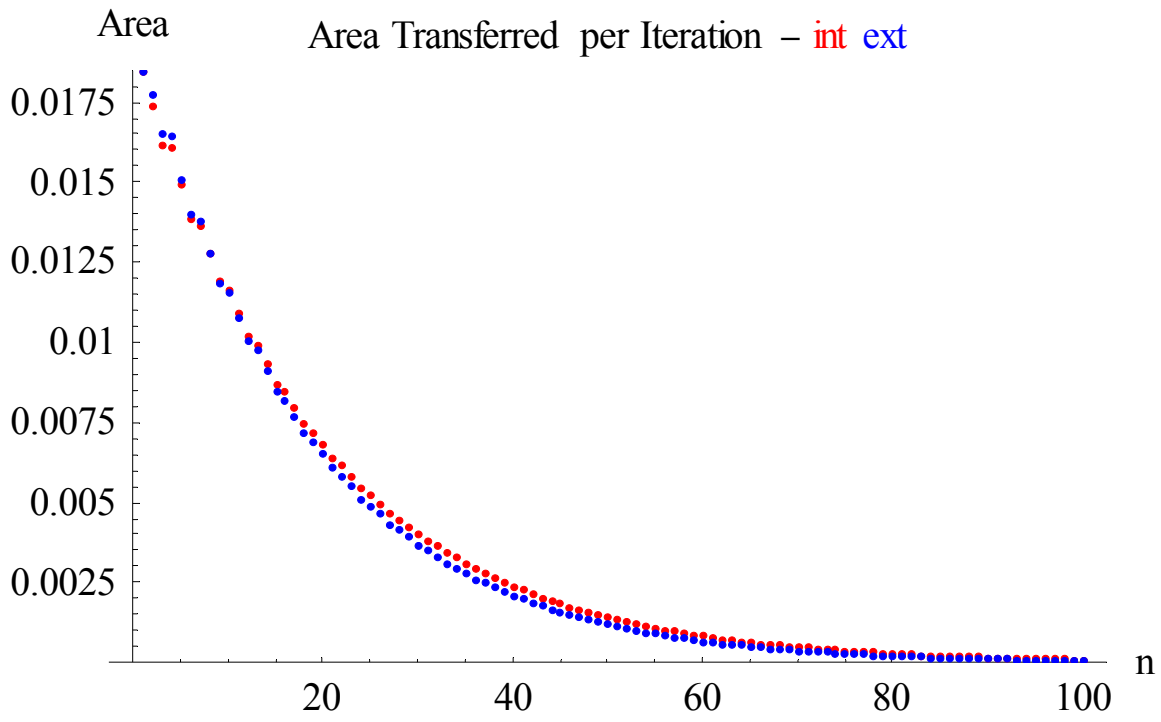

Figure 6.1.2 L = 2, ε = 0.01, ω = 1.21971 Area transferred per iteration

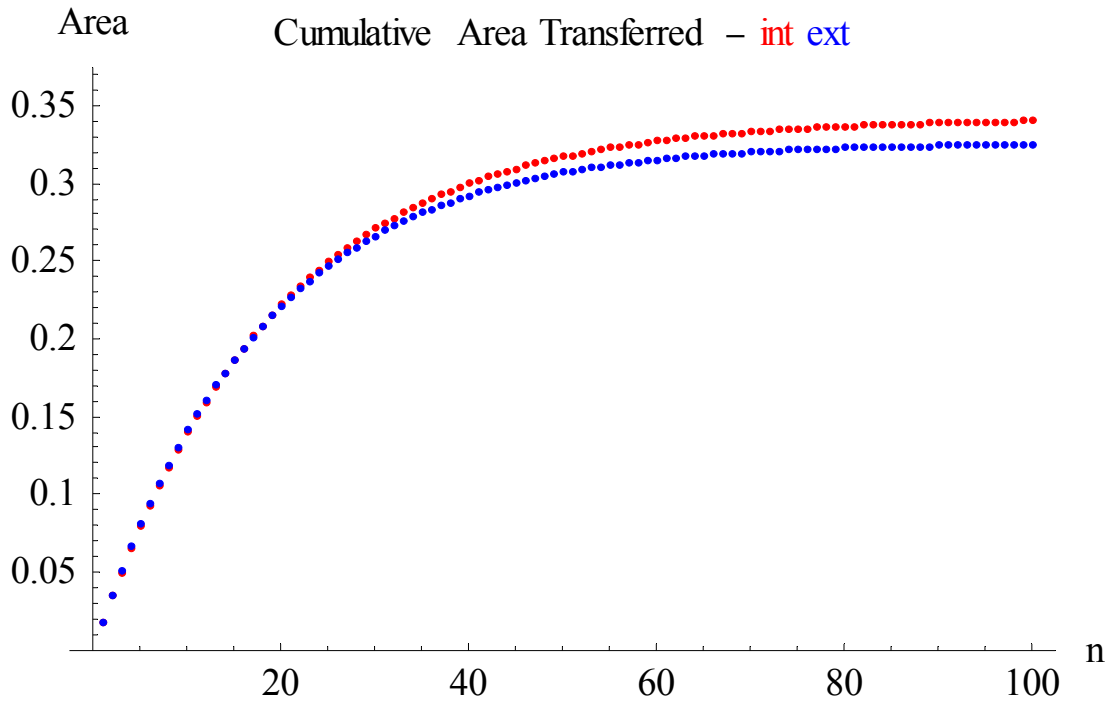

Figure 6.1.3 L = 2, ε = 0.01, ω = 1.21971 Cumulative area transferred

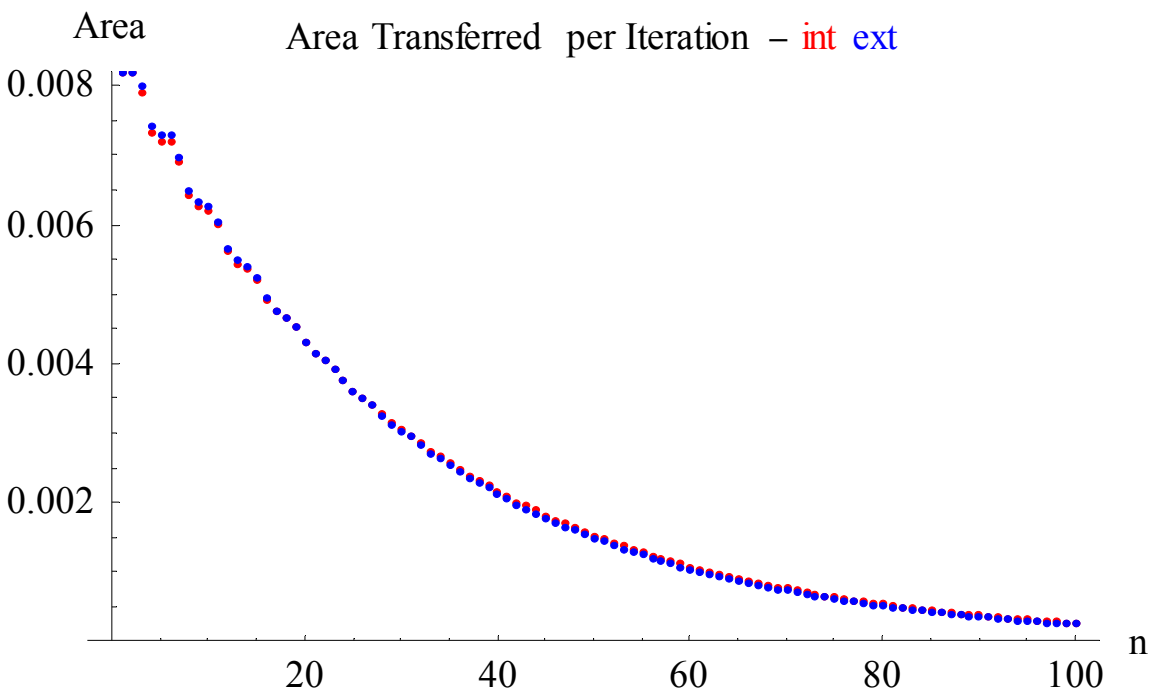

Figure 6.1.4 L = 3, ε = 0.01, ω = 1.55572 Area transferred per iteration

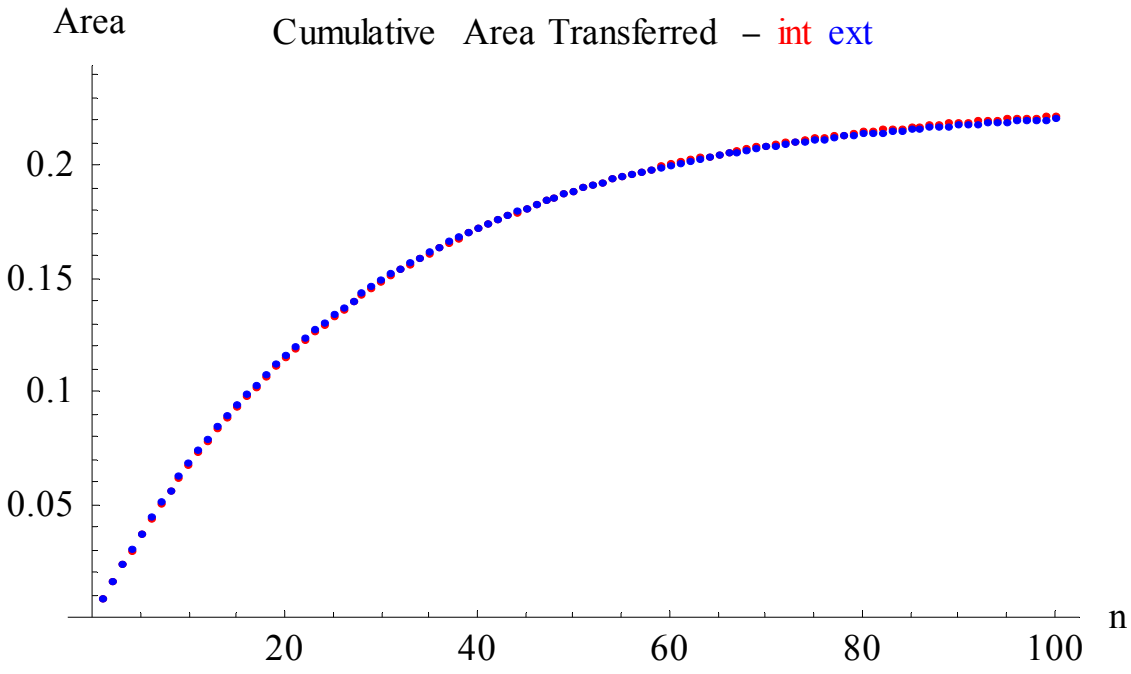

Figure 6.1.5 L = 3, ε = 0.01, ω = 1.55572 Cumulative area transferred

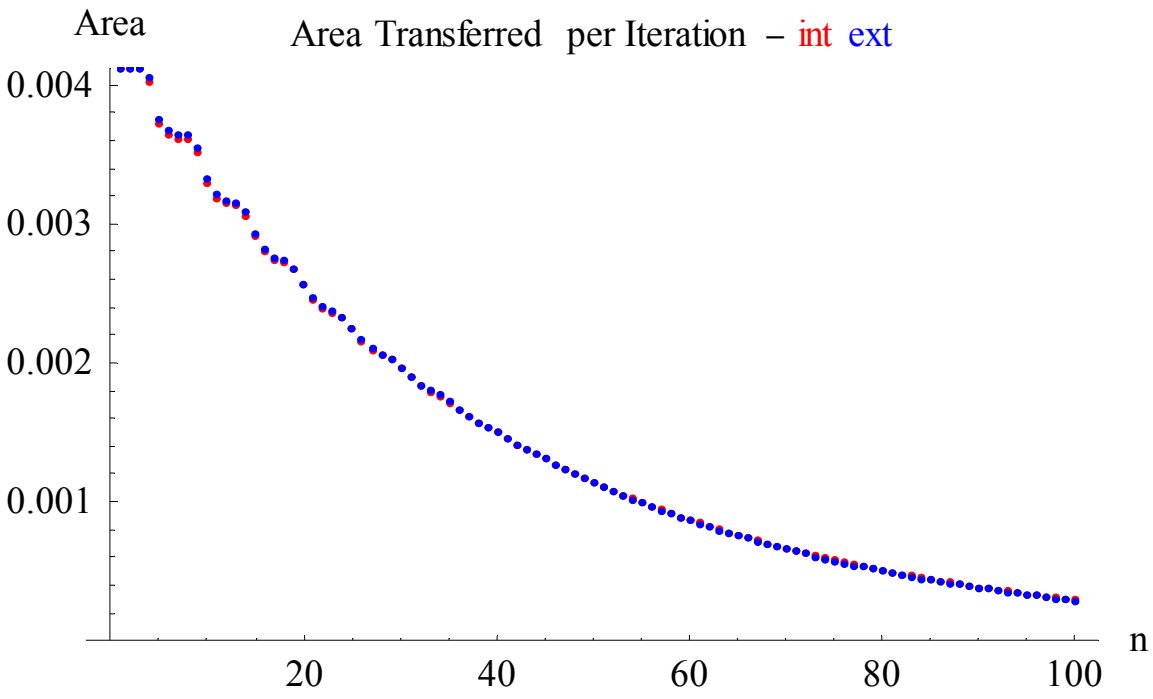

Figure 6.1.6 L = 4, ε = 0.01, ω = 1.85766 Area transferred per iteration

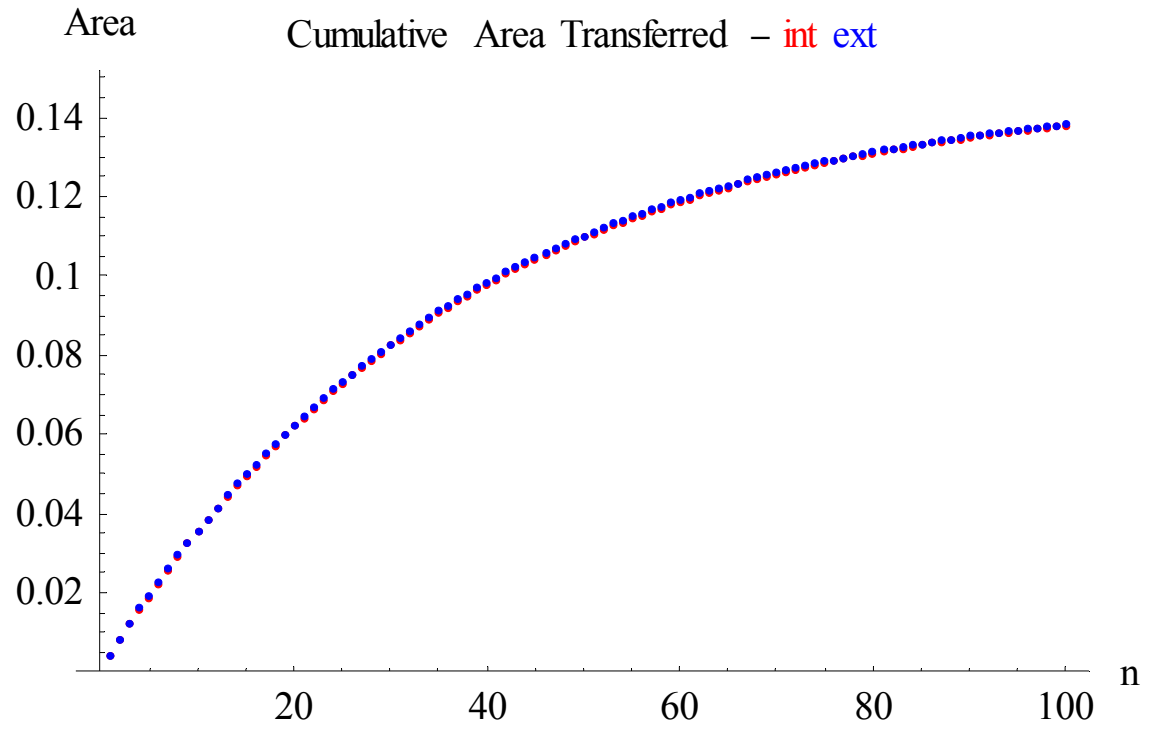

Figure 6.1.7 L = 4, ε = 0.01, ω = 1.85766 Cumulative area transferred

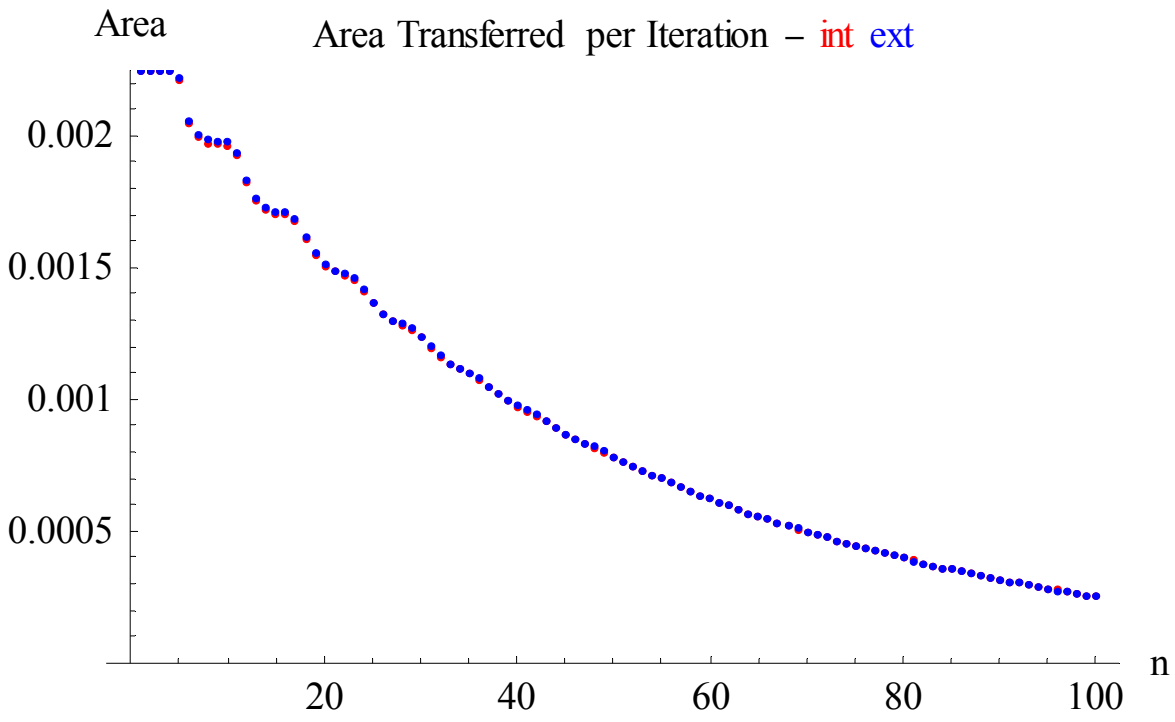

Figure 6.1.8 L = 5, ε = 0.01, ω = 2.13393 Area transferred per iteration

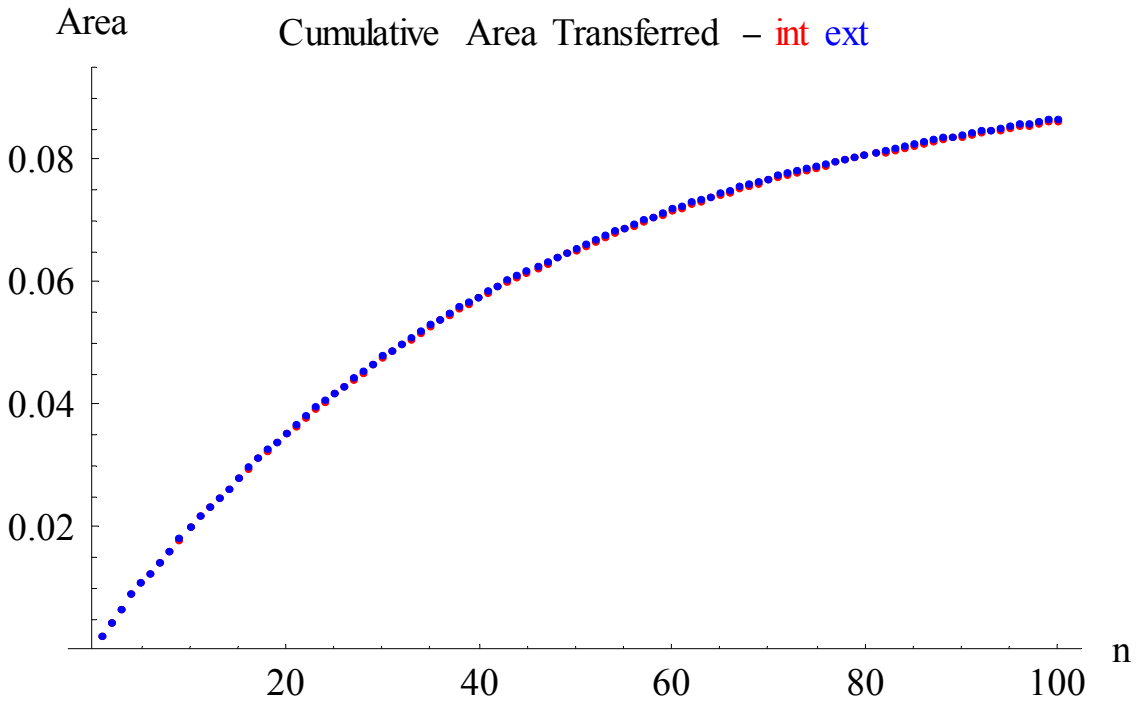

Figure 6.1.9 L = 5, ε = 0.01, ω = 2.13393 Cumulative area transferred

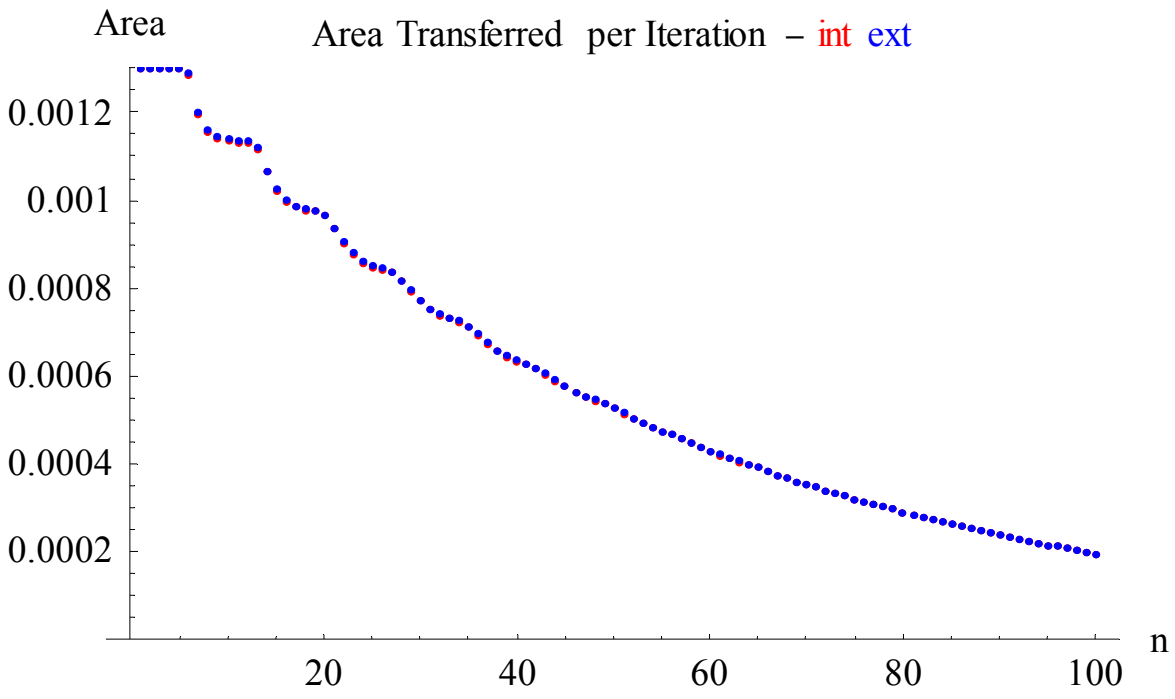

Figure 6.1.10 L = 6, ε = 0.01, ω = 2.39005 Area transferred per iteration

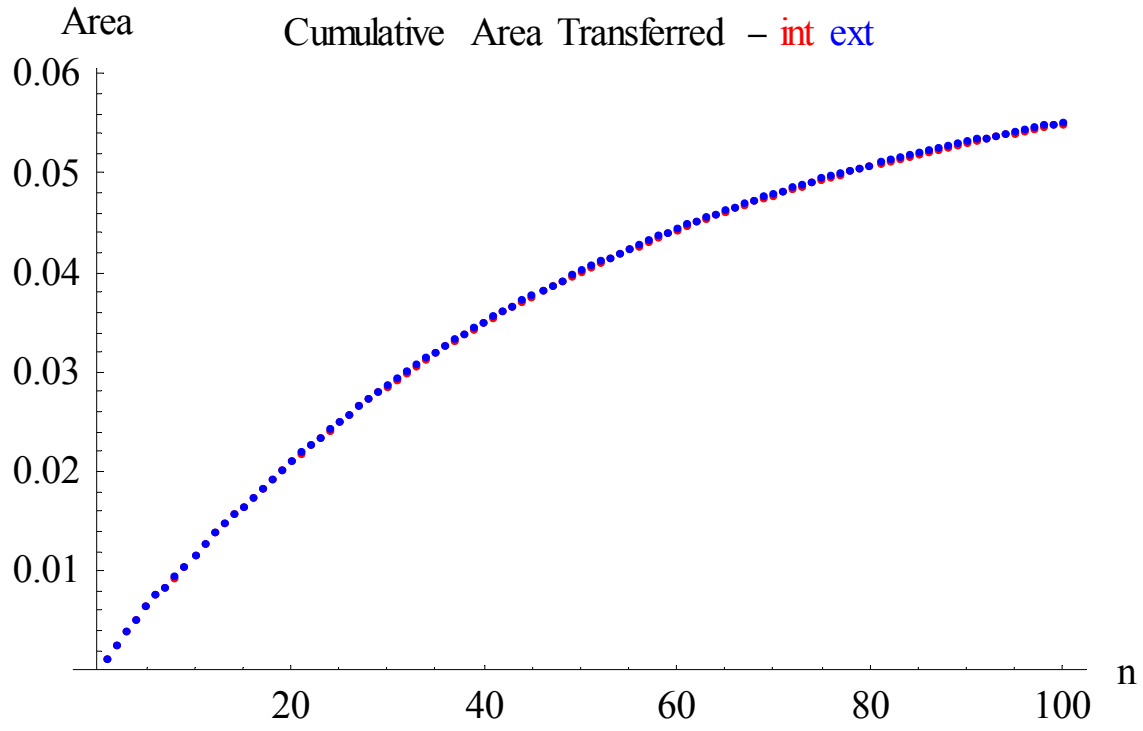

Figure 6.1.11 L = 6, ε = 0.01, ω = 2.39005 Cumulative area transferred

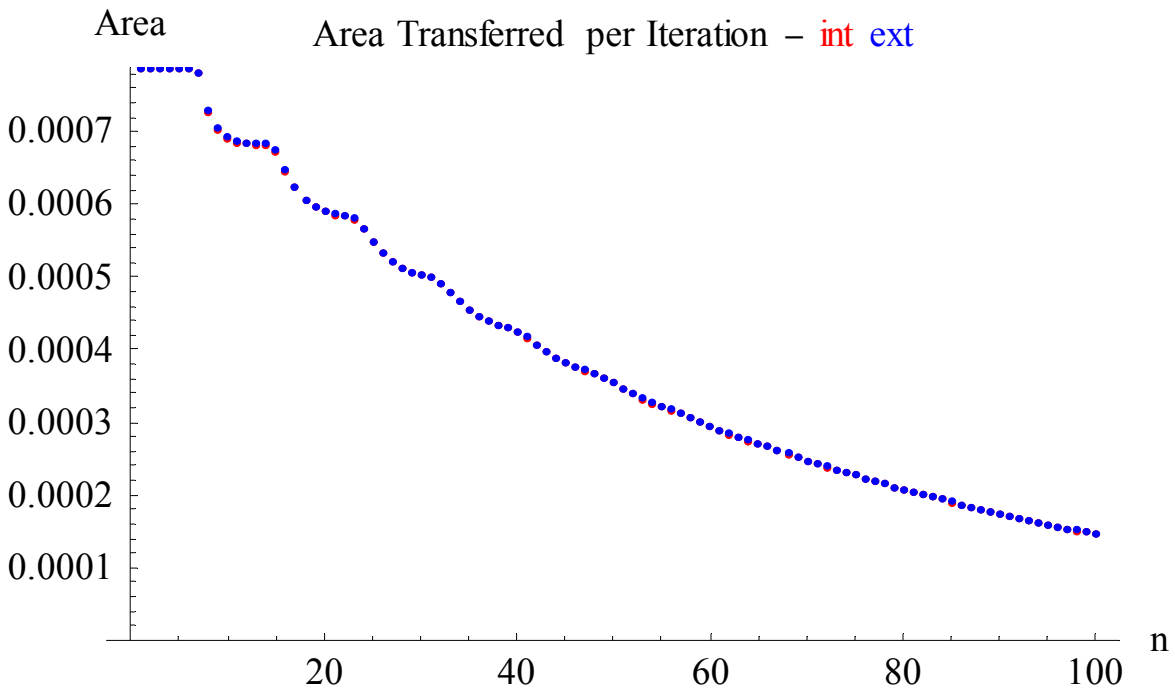

Figure 6.1.12 L = 7, ε = 0.01, ω = 2.62982 Area transferred per iteration

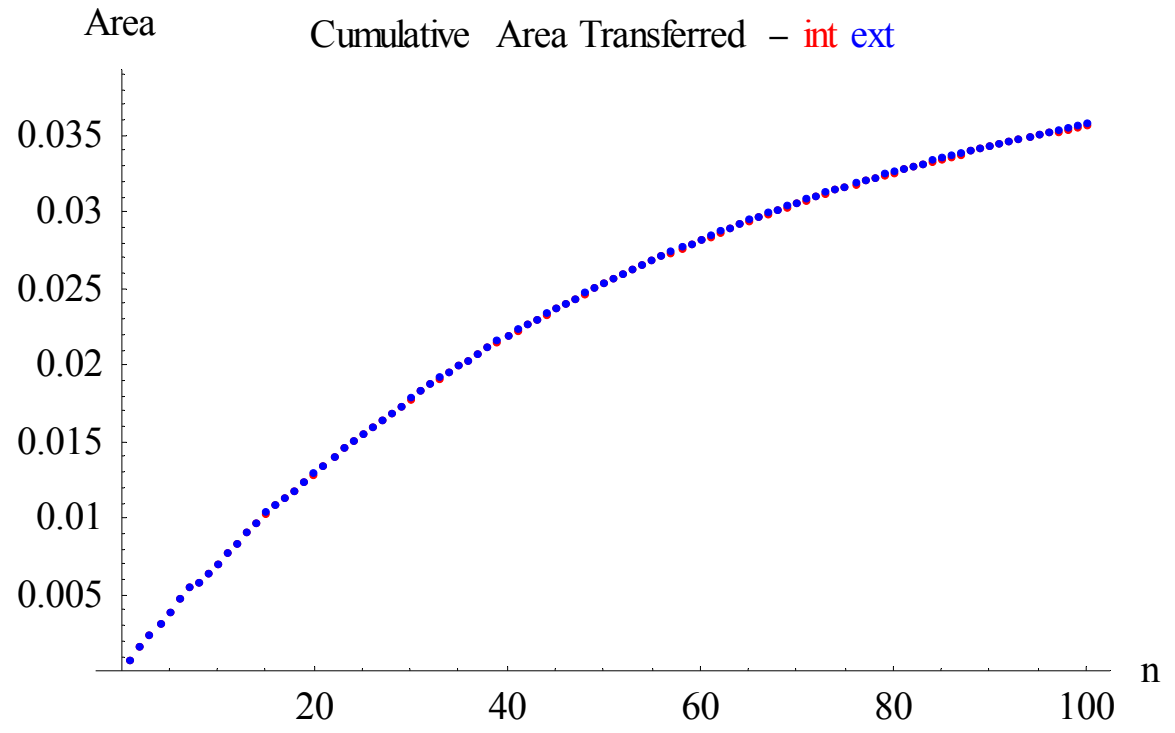

Figure 6.1.13 L = 7, ε = 0.01, ω = 2.62982 Cumulative area transferred

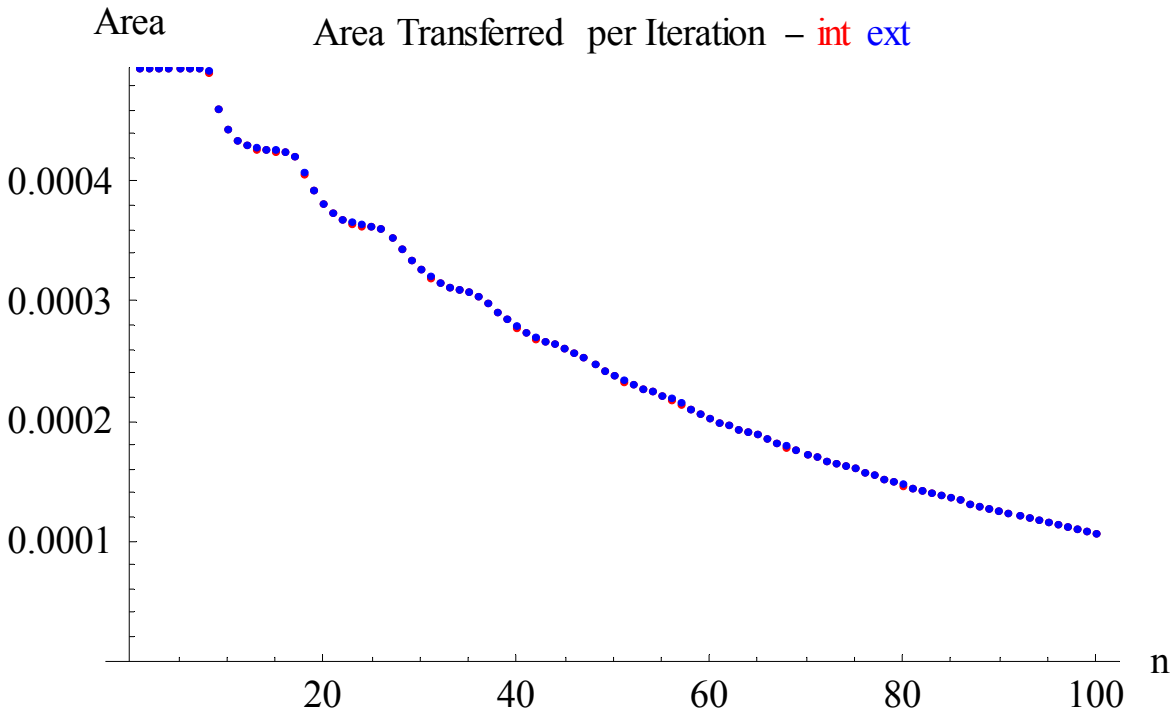

Figure 6.1.14 L = 8, ε = 0.01, ω = 2.85599 Area transferred per iteration

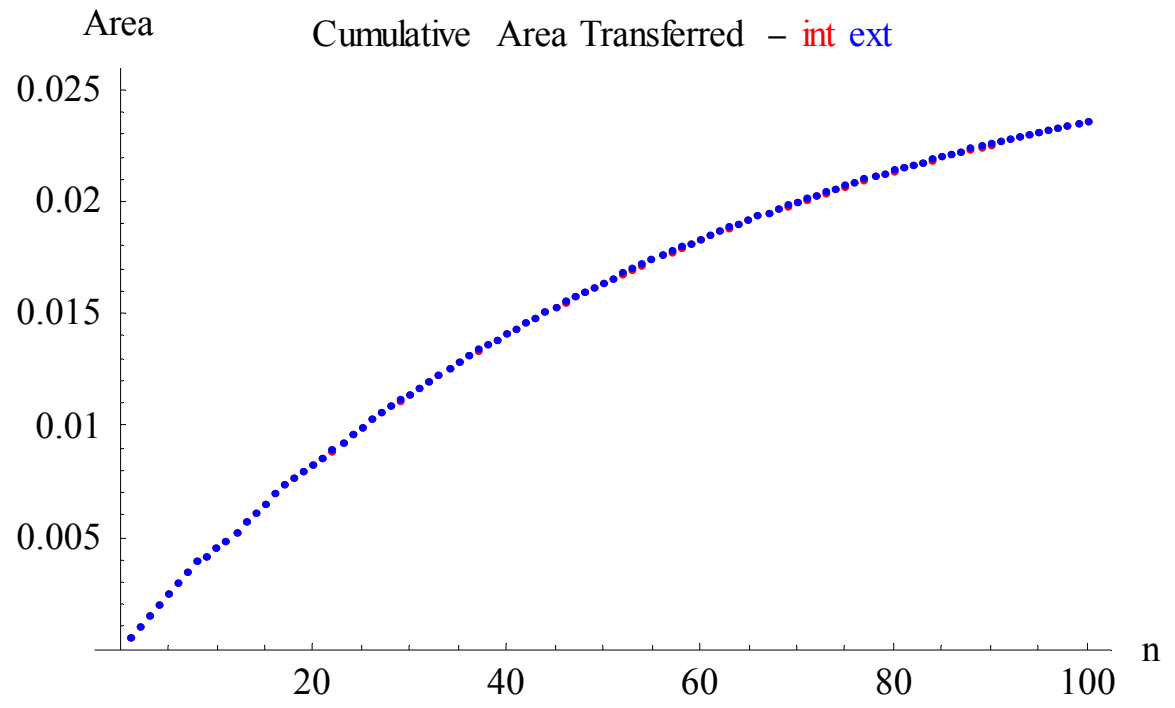

Figure 6.1.15 L = 8, ε = 0.01, ω = 2.85599 Cumulative area transferred

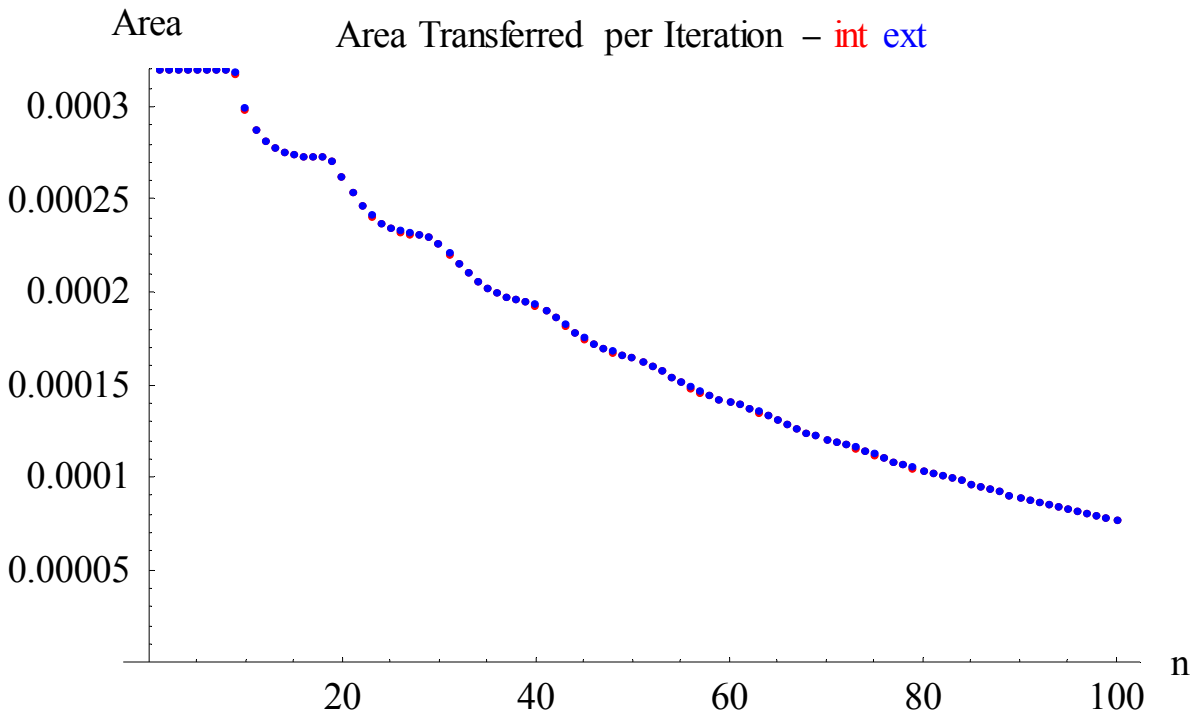

Figure 6.1.16 L = 9, ε = 0.01, ω = 3.07063 Area transferred per iteration

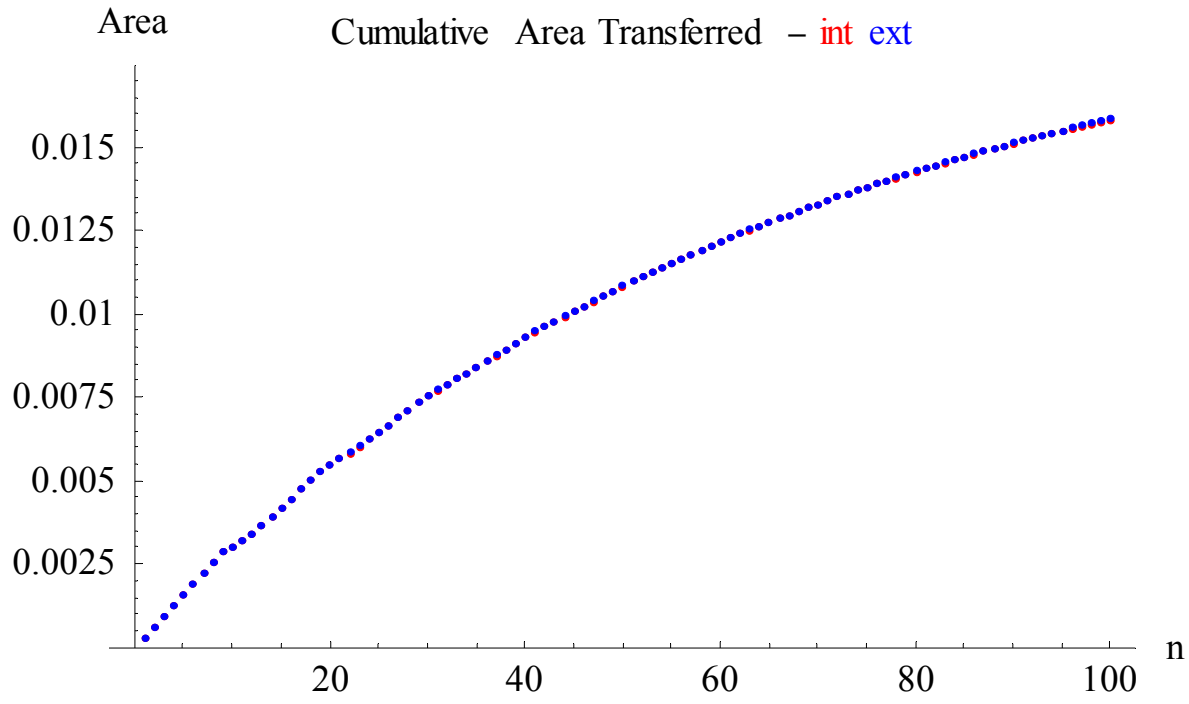

Figure 6.1.17 L = 9, ε = 0.01, ω = 3.07063 Cumulative area transferred

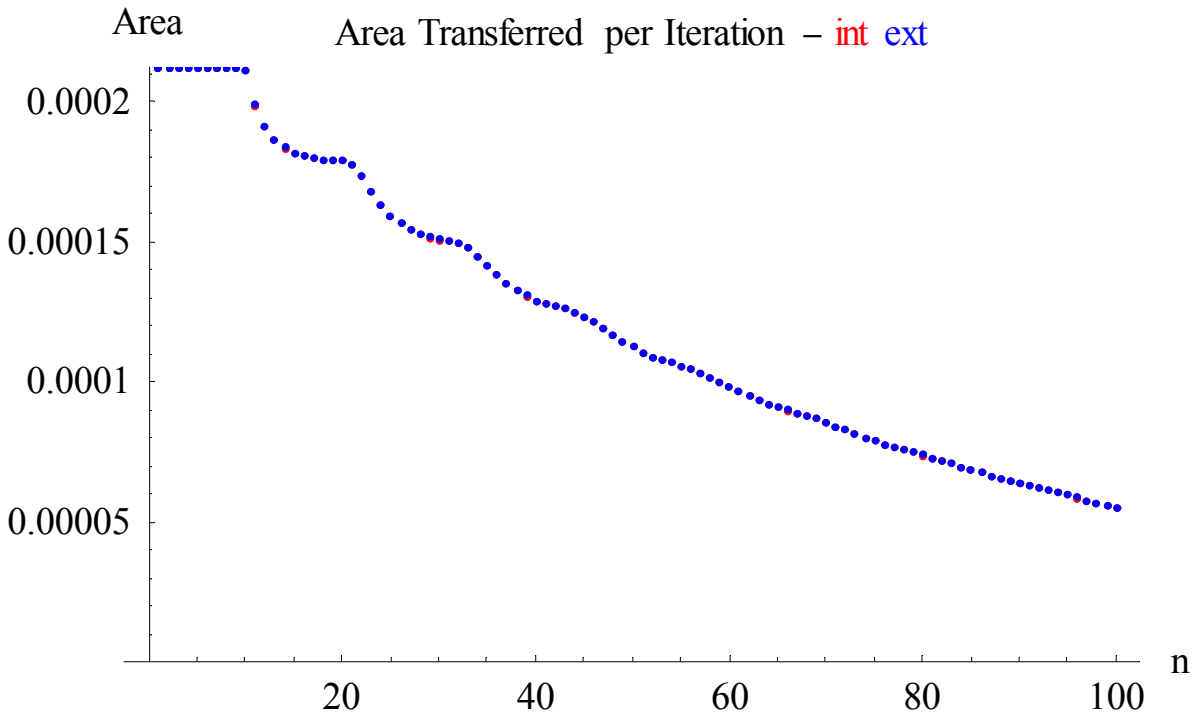

Figure 6.1.18 L = 10, ε = 0.01, ω = 3.27532 Area transferred per iteration

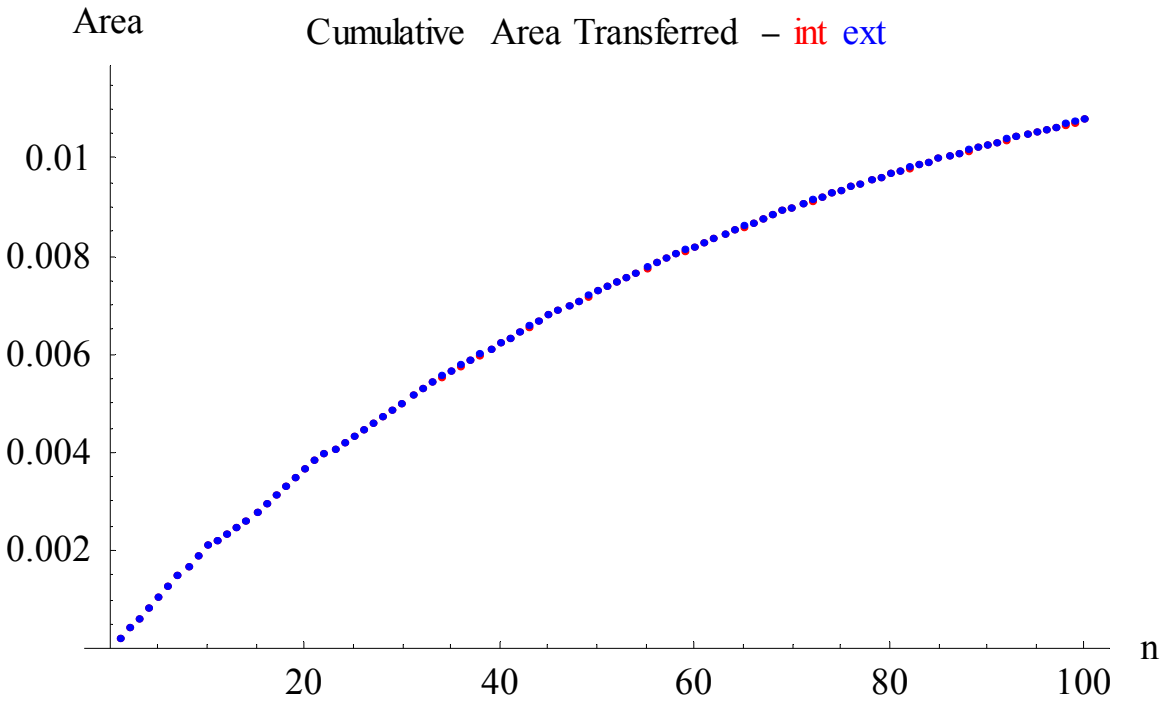

Figure 6.1.19 L = 10, ε = 0.01, ω = 3.27532 Cumulative area transferred

6.2 Comparison to Theoretical Flux Function

Glancing at the area (vertical) axes of the graphs in the preceding section indicates that the area transferred decreases as L, and thus, ω increases. Setting $\varepsilon = 1$ to remove the scaling effect of the perturbation strength, the lobe area is plotted as a function of frequency in figure 6.2.1.

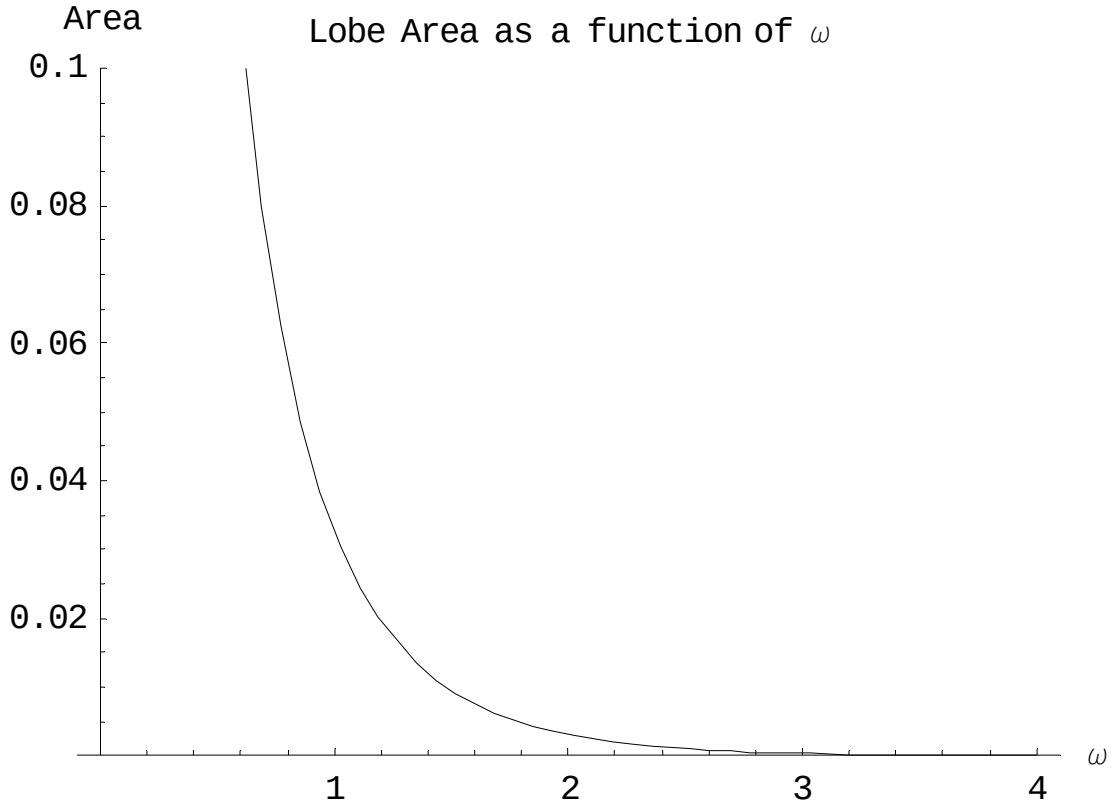

Figure 6.2.1 Lobe area as a function of ω

As can be seen from the graph, lobe area exponentially decreases as perturbation frequency ω increases. This can be deduced from the shape of the Melnikov amplitude function, since lobe area is calculated as $\left|\int_0^{\pi/\omega} M(t)\,dt\right|$, which equals $\frac{2}{\omega}C(\omega)$, where $C(\omega)$ is the Melnikov amplitude function shown if Figure 3.1.3.

Rom-Kedar and Poje (1999) proposed a theoretical flux function for one iteration cycle as (lobe area)/(perturbation period). Dividing the area of a lobe, $\frac{2}{\omega}C(\omega)$, by one perturbation period T = $\frac{2\pi}{\omega}$ leads to a theoretical flux function of

$$F(\omega) = \frac{1}{\pi}C(\omega) \qquad \text{Eq.6.2.1a}$$

Since all of the results graphed in section 6.1 are for 100 iterations, the theoretical flux function was modified to include iteration number n:

$$F(\omega) = n\frac{1}{\pi}C(\omega) \qquad \text{Eq.6.2.1b}$$

This modified theoretical flux function is graphed in Figure 6.2.2.

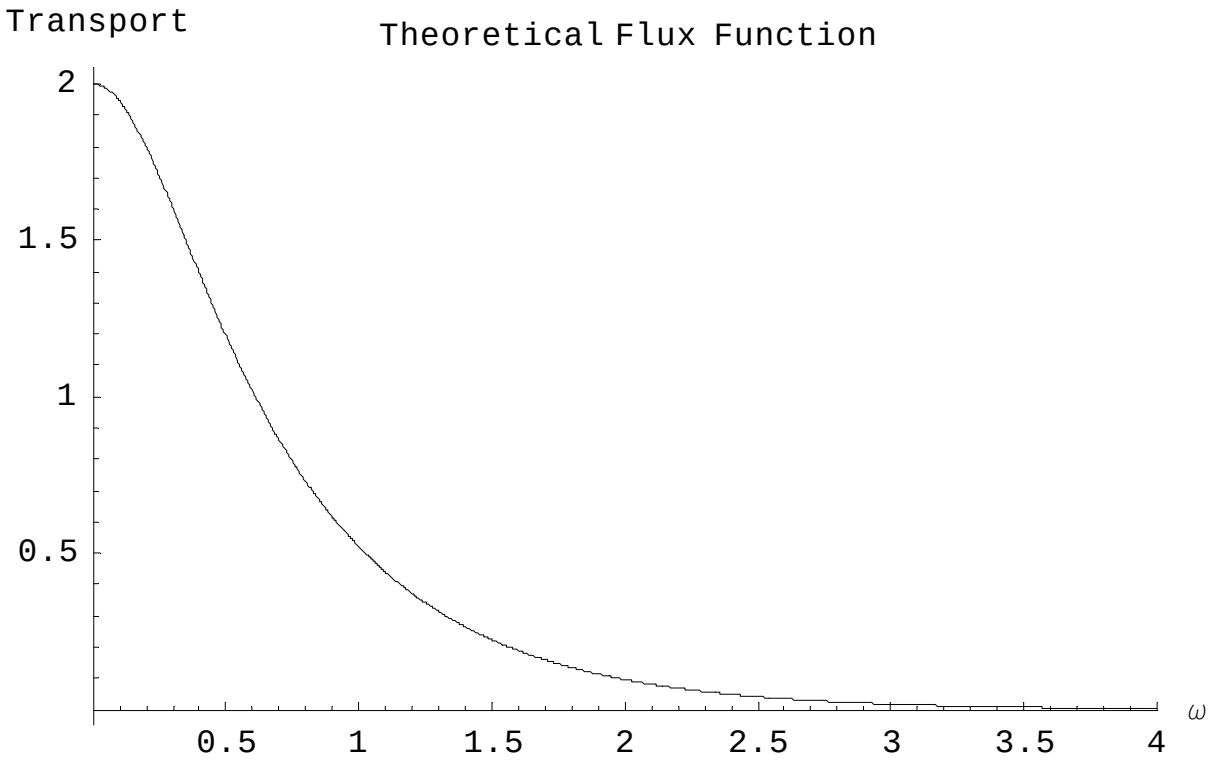

Figure 6.2.2 Theoretical flux function for n = 100 iterations

It is not surprising that the shape of the theoretical flux function is exactly that of Melnikov amplitude function *C*(ω) found in Figure 3.1.3.

The cumulative area transferred by the 100th iteration for L = 2 to L = 10, for both the internal and external lobes, are compared to the theoretical flux function in Figure 6.2.3. The graph only shows the transport area in the range of the values calculated by the Topological Approximation Method, and does not include the low frequency portion of the theoretical flux function since there are no calculated values for L = 0 or L = 1 in the low frequency region.

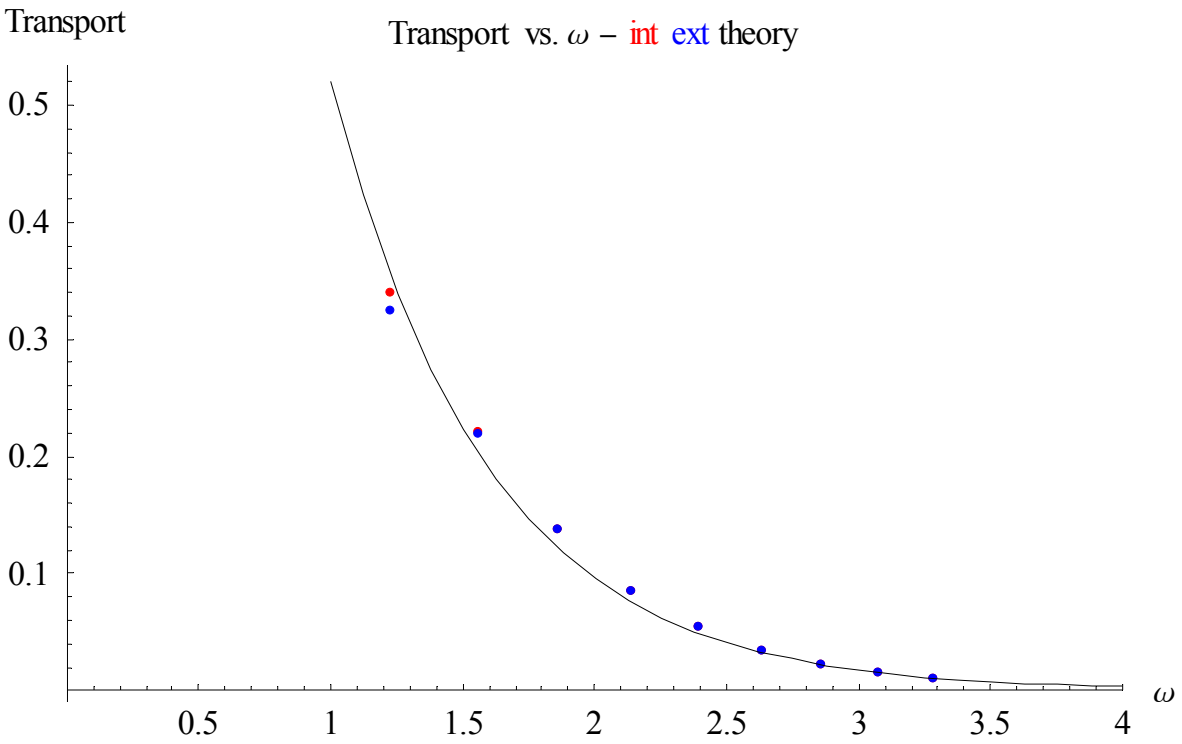

Figure 6.2.3 Comparison of TAM calculated values
and the theoretical flux function

The transport values calculated by the Topological Approximation Method fit the theoretical flux function extremely well for the application domain of the TAM. Transport depends on frequency, with a rapid decrease in transport rate as perturbation frequency increases. The size of the lobes decreases rapidly with an increase in the perturbation frequency, thus reducing the amount of phase space available to be transported. The lobes define a band or zone of transport near the unperturbed heteroclinic orbit. At lower frequencies, a broad band enveloping the heteroclinic orbit allows for relatively large transfer rates, but as the frequency increases, this band becomes narrower, thus restricting the amount of transport. A complete

description of the frequency dependence of the transport would indicate the choice of the optimal perturbation frequency for maximizing transport at given values of the perturbation strength.

6.3 Conclusions

An analytical expression for the Melnikov function for the Kelvin-Stuart driven flow was derived, along with expressions for the internal and external period functions, allowing the application of the Topological Approximation Method to the Kelvin-Stuart system. Details of these analytical calculations are found in Appendix A.

Mathematica® modules were developed to calculate the internal and external bifurcation curves and to perform all of the calculations necessary (selecting a ω value for a given ε, finding lobe intersection points, calculating lobe intersection areas, deriving values for the weighting factors) to generate the lobe dynamics transfer matrix and the initial vector in order to calculate lobe transport rates. These modules are flexible, and can be used with a wide range of ε, ω, L, and n values, not just those found in this paper. The modules can be adapted to other dynamical systems by changing the definitions of the Melnikov function and the period functions, found near the beginning of each module. These modules are included in Appendix B.

Bifurcation curves were plotted for L = 0 to L = 10 using the module in Appendix B.1 The program in Appendix B.2 was run for L values ranging from 2 to 10, with ε fixed at 0.01, ω varied with L according to the bifurcation curves, and n set to 100 iterations. Transport rates for each of these cases were calculated and graphed. The transport rates indicate a dependency upon frequency, decreasing rapidly as frequency increases. One primary reason for this rapid decline is that the lobe area decreases rapidly with increasing frequency and transport depends directly upon lobe area. The extent of the lobes determines an effective area of transport, a zone around the unperturbed heteroclinic orbit. As frequency increases, this effective area of transport becomes narrower as the areas of the lobes decrease, reducing the amount of phase space available for transport.

Transport values calculated with the Topological Approximation Method agree very well with the theoretical flux function proposed by Rom-Kedar and Poje, when modified to include the number of iterations, in the frequency range for L = 2 and larger.

6.4 Suggestions for Further Research

The Topological Approximation Method can only be applied as presently formulated in the region included between the minimum and the maximum bifurcation curves for any given L value. This leaves large frequency domains, gaps between the bifurcation curves for different L values, in which the TAM cannot be applied as of yet. In terms of the frequency at a given value of ε, as frequency is increased, a lobe goes from four intersection points down to two intersection points down to none within one set of bifurcation curves for a given value of the structure index L. A gap exists between one set of bifurcation curves for a given L and the adjacent set for the next value of L. Furthermore, the lobe jumps from zero intersection points, at the last bifurcation curve from the first set of bifurcation curves for the original L value, to four intersection points at the first curve encountered in the new set of bifurcation curves. Further research is necessary to understand the lobe evolution within these gaps, and to refine the Topological Approximation Method in order to apply the TAM to these regions.

The method described in section 4.5 to derive the expressions for individual weighting factors given in Eqs.4.5.4, 5 can only be used for $L \geq 2$. For $L = 0$ and $L = 1$ this approach can only solve for linear combinations of weighting factors, not the individual weighting factors. Further iterations keep generating the same linear combinations. Additional research is needed to find either a successful method for deriving the weighting factors for $L = 0$ and $L = 1$, or some other means of calculating transport for $L = 0$ and $L = 1$ frequencies.

The present calculations indicate a strong dependency of transport upon frequency for the frequency range investigated, but no information has been obtained regarding low frequency dependence. Research of low frequency transport will lead to a more complete description of the frequency dependence of transport.

Many field studies consist of numerical data, with no analytical streamfunction or Hamiltonian available to perform the dynamical analysis required by the Topological Approximation Method in its present form. Research into adapting the TAM to numerical studies would lead to much greater applicability.

The form of the Whisker map h iterates, which essentially reproduce the entire heteroclinic tangle between two consecutive zeroes of the previous iterate, the bifurcation curves, and the lobe intersection point locations all suggest some form of scaling and self-similarity,

which is beyond the scope of the present work, but may be of some significance and worth investigating.

6.5 Final Thoughts

Henri Poincaré stated (1892) that "…when we try to represent the figure formed by these two curves (the stable and unstable manifolds of a hyperbolic fixed point) and their intersections, … , these intersections form a type of trellis, tissue, or grid with infinitely serrated mesh. Neither of the two curves must ever cut across itself again, but must bend back upon itself in a very complex manner in order to cut across all of the meshes in the grid an infinite number of times. The complexity of this figure will be striking, and I shall not even try to draw it. Nothing is more suitable for providing us with an idea of the complex nature of the three body problem, and of all the problems of dynamics in general …" Developments such as lobe dynamics transport theory and the Topological Approximation Method are beginning to provide glimpses into the complex world of heteroclinic tangles, and inspire hope that we may be able to achieve deeper understanding of the problems of dynamics.

References

Abraham, Ralph H., and Shaw, Christopher D. (1988) *Dynamics-The Geometry of Behavior*

Part One: Periodic Behavior

Part Two: Chaotic Behavior

Part Three: Global Behavior

Part Four: Bifurcation Behavior

Aerial Press, San Diego (ISBN 0-942344-00-6 for four-volume set)

Aref, Hassan (1984), Stirring by chaotic advection, *J. Fluid. Mech.*, **143**, 1-21

Aref, Hassan (2002), The development of chaotic advection, *Physics of Fluids*, **14** no. 4, 1315-1325

Arfken, George (1985), *Mathematical Methods for Physicists*, 3rd ed., Academic Press, San Diego (ISBN 0-12-059820-5)

Arrowsmith, D.K., and Place, C.M. (1990), *An Introduction to Dynamical Systems*, Cambridge University Press (ISBN 0521303621)

Chirikov, B.V. (1979), A Universal instability of many-dimensional oscillator systems, *Phys. Rep.* **52**, 263-379

Coulliette, C. and Wiggins, S. (2000), Intergyre Transport in a Wind-Driven, Quasigeostrophic Double Gyre: An Application of Lobe Dynamics, *Nonlinear Processes in Geophysics*, **7**, no.1/2, 59-86

Duan, J., and Wiggins, S. (1997), Lagrangian Transport and Chaos in the Near Wake of the Flow Around an Obstacle: A Numerical Implementation of Lobe Dynamics, *Nonlinear Processes in Geophysics*, **4**, 125-136

Easton, Robert W. (1986), Trellises formed by stable and unstable manifolds in the plane, *Trans. Am. Math. Soc.*, **294**, 719-732

Gradshteyn, I.S., and Ryzhik, I.M. (1994), *Table of Integrals, Series, and Products*, 5th ed., Alan Jeffrey, editor, Academic Press, San Diego (ISBN 0-12-294755-X)

Guckenheimer, John, and Holmes, Philip (1983), *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Applied Mathematical Sciences vol. 42, Springer-Verlag, New York (ISBN 0-387-90819-6)

Hilborn, Robert C. (1994), *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*, Oxford University Press, New York (ISBN 0-19-5057560-0)

Lamb, Sir Horace (1932), *Hydrodynamics*, 6th ed., Dover Publications

Litvak-Hinenzon, Anna (1996), The homoclinic tangle of slightly dissipative, forced, two dimensional systems, M.S. thesis (Feb. 1996) Weizmann Institute of Science, Israel

Litvak-Hinenzon, Anna, and Rom-Kedar, Vered (1997), Symmetry breaking perturbations and strange attractors, *Phys. Rev.* E **55** no.5, 4964-4978

Nayfeh, Ali F., and Balachandram, Balakumar (1995), *Applied Nonlinear Dynamics: Analytical, Computational and Experimental Methods*, John Wiley and Sons (ISBN 0-471-59348-6)

Ottino, J.M. (1989), *The kinematics of mixing: stretching, chaos, and transport*, Cambridge Texts in Applied Mathematics, Cambridge University Press (ISBN 0 521 36335 7)

Ottino, Julio M., and Wiggins, Stephen (2004), Introduction: mixing in microfluidics, *Phil. Trans. R. Soc. Lond.* A, **362**, 923-935

Poincare, H. (1892), *Les Methods Nouvelles de la Mechanique Celeste*, Gauthier-Villars, Paris

Rom-Kedar, V. (1990), Transport rates of a class of two-dimensional maps and flows, *Physica* D, **43**, 229-268

Rom-Kedar V. (1993), The topological approximation method, in *Transport, Chaos and Plasma Physics (Proc., Marseille)* editors S. Benkadda, F. Doveil, and Y. Elskens, World Scientific Press, River Edge, NJ (ISBN 981-02-1619-X) 39-57

Rom-Kedar, V. (1994), Homoclinic tangles – classification and applications, *Nonlinearity*, **7**, 441-473

Rom-Kedar, Vered (1995), Secondary homoclinic bifurcation theorems, *Chaos*, **5** no.2, 385-401

Rom-Kedar, V., Leonard, A., and Wiggins, S. (1990), An analytical study of transport, mixing and chaos in an unsteady vortical flow, *J. Fluid Mech.,* **214**, 347-394

Rom-Kedar, V., and Poje, A.C. (1999), Universal properties of chaotic transport in the presence of diffusion, *Physics of Fluids*, **11** no.8, 2044-2057

Rom-Kedar, V., and Wiggins, S. (1990), Transport in Two-Dimensional Maps, *Arch. Rat. Mech. Anal.*, **109**, 239-298

Rom-Kedar, Vered, and Wiggins, Stephen (1991), Transport in two-dimensional maps: Concepts, examples, and a comparison of the theory of Rom-Kedar and Wiggins with the Markov model of MacKay, Meiss, Ott, and Percival, *Physica* D, **51**, 248-266 (1991)

Ruskeepää, Heikki (2004), *Mathematica® Navigator: Mathematics, Statistics, and Graphics*, 2nd ed., Elsevier Academic Press (ISBN 12603642X)

Stuart, J.T. (1967), On finite amplitude oscillations in laminar mixing layers, *J. Fluid Mech.*, **29** (part 3), 417-440

Stuart, J.T. (1971), Stability Problems in Fluids, in *Mathematical Problems in the Geophysical Sciences*, Lectures in Applied Mathematics, ed. William H. Reid, American Mathematical Society 139-155

Tsega, Yamlak, Michaelides, Efstathios E., and Eschenazi, Elia V. (2001), Particle dynamics and mixing in the frequency driven "Kelvin cat eyes" flow, *Chaos*, **11** no. 2, 351-358

Wiggins, S. (1990), *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Springer-Verlag, New York (ISBN 0-387-97003-7)

Wiggins, S. (1992), *Chaotic Transport in Dynamical Systems*, Springer-Verlag, New York (ISBN 0-387-97522-5)

Wiggins, Stephen (2005), The dynamical systems approach to Lagrangian transport in oceanic flows, *Annual Review of Fluid Mechanics*, **37**, 295-328

Wiggins, Stephen, and Ottino, Julio M. (2004), Foundations of chaotic mixing, *Phil. Trans. R. Soc. Lond.* A, **362**, 937-970

Wolfram, Stephen (1999), *The Mathematica® Book*, 4th ed., Wolfram Media/Cambridge University Press (ISBN 1-57955-004-5 and 0-521-64314-7)

Appendix A

Analytical Calculations

A.1 Fixed Point Analysis – Eigenvalues and Eigenvectors

The unperturbed Hamiltonian for the Kelvin-Stuart Cat Eyes dynamical system is

$$H(x,y) = \ln(\cosh y + A\cos x) - \ln(1+A) \qquad \text{Eq. A.1.1}$$

The ln(1+A) term is subtracted so as to have $H = 0$ on the heteroclinic orbit (separatrix). For H to be real for all values of x and y, the absolute value of A must be less than 1. The dynamical equations of motion for the unperturbed system are:

$$\dot{x} = \frac{\partial H}{\partial y} = \frac{\sinh y}{\cosh y + A\cos x} \qquad \text{Eq.A.1.2.a}$$

$$\dot{y} = -\frac{\partial H}{\partial x} = \frac{A\sin x}{\cosh y + A\cos x} \qquad \text{Eq.A.1.2.b}$$

Critical (fixed) points for the dynamical system occur when $\dot{x} = 0, \Rightarrow \sinh y = 0 \Rightarrow y = 0$ and when $\dot{y} = 0, \Rightarrow A\sin x = 0 \Rightarrow x = n\pi$. To find the eigenvalues and eigenvectors for the system, linearize about the critical (fixed) points:

$$DF(x,y) = \begin{pmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{A\sin x \sinh y}{(\cosh y + A\cos x)^2} & \frac{1 + A\cos x\cosh y}{(\cosh y + A\cos x)^2} \\ \frac{A^2 + A\cos x\cosh y}{(\cosh y + A\cos x)^2} & \frac{-A\sin x\sinh y}{(\cosh y + A\cos x)^2} \end{pmatrix} \qquad \text{Eq.A.1.3a}$$

Solve the characteristic equation $|DF - \lambda I| = 0$ for both $(2n\,\pi, 0)$ and $((2n+1)\,\pi, 0)$.

For $A > 0$, $DF(2n\pi, 0) = \begin{pmatrix} 0 & \frac{1}{1+A} \\ \frac{A}{1+A} & 0 \end{pmatrix}$, yielding characteristic equation $\lambda^2 - \frac{A}{(1+A)^2} = 0$,

with resulting eigenvalues and eigenvectors:

$$\lambda_{\pm} = \pm\frac{\sqrt{A}}{1+A} \qquad \text{Eq.A.1.4a}$$

Eigenvector for λ_+ is $\vec{x}_u = \frac{1}{\sqrt{1+A}}\begin{pmatrix} 1 \\ \sqrt{A} \end{pmatrix}$ Eq.A.1.4b

Eigenvector for λ_- is $\vec{x}_s = \frac{1}{\sqrt{1+A}}\begin{pmatrix} 1 \\ -\sqrt{A} \end{pmatrix}$ Eq.A.1.4c

The points ($2n\pi$, 0) are thus hyperbolic fixed points, or saddle points. The eigenvector for λ_+ indicates the direction of the unstable manifold, and the eigenvector for λ_- indicates the direction of the stable manifold.

$DF((2n+1)\pi,0) = \begin{pmatrix} 0 & \frac{1}{1-A} \\ \frac{-A}{1-A} & 0 \end{pmatrix}$, yielding characteristic equation $\lambda^2 + \frac{A}{(1-A)^2} = 0$,

with resulting eigenvalues:

$$\lambda_{\pm} = \pm i\frac{\sqrt{A}}{1-A} \qquad \text{Eq.A.1.5}$$

The points ((2n + 1) π, 0) are thus centers of rotation.

In the case of $A < 0$, the locations of the hyperbolic fixed points and the centers of rotation are reversed, i.e., the system is translated in the x-direction by $\pm\pi$.

A.2 Heteroclinic Orbit (Separatrix), Including Parametric Form

The heteroclinic orbit connects two adjacent hyperbolic fixed points, so the value of $H(x, y)$ is:

$$H(2n\pi,0) = \ln(\cosh(0) + A\cos(2n\pi) - \ln(1 + A) = 0 \qquad \text{Eq.A.2.1a}$$

Therefore, on the heteroclinic orbit,

$$\cosh y + A\cos x = 1 + A \qquad \text{Eq.A.2.1b}$$

To find the parametric form of the heteroclinic orbit, substitute Eq.A.2.1b into Eqs.A.1.2a, b, and use the Pythagorean identities for sin x and sinh y, to give:

$$\dot{x} = \frac{1}{1+A}\sqrt{A(1-\cos x)[2 + A(1-\cos x)]} \qquad \text{Eq.A.2.2a}$$

$$\dot{y} = \frac{1}{1+A}\sqrt{(\cosh y - 1)[2A - (\cosh y - 1)]} \qquad \text{Eq.A.2.2b}$$

Square each equation (Eq.A.2.2a and b), and either substitute $u = A(1-\cos x)$ into the square of Eq.A.2.2a, or substitute $u = \cosh y - 1$ into the square of Eq.A.2.2b. Taking the derivative of u with respect to time in each case and writing both equations in terms of $\dot{u}^2$ leads to the following result in both cases:

$$\dot{u}^2 = \lambda_{\pm}{}^2 u^2(4 + 2(1 - \tfrac{1}{A}) - \tfrac{1}{A}u^2) \qquad \text{Eq.A.2.3a}$$

Take the square root of Eq.A.2.3, separate the variables, and integrate to achieve the following:

$$\int \frac{du}{u\sqrt{4+2(1-\frac{1}{A})-\frac{1}{A}u^2}} = \lambda_{\pm}\int dt \qquad \text{Eq.A.2.3b}$$

Let $R = a + bu + cu^2$, with $a = 4$, $b = 2(1-\frac{1}{A})$, $c = -\frac{1}{A}$. Since $a = 4 > 0$, formula 2.266 on p.101 of the 5th edition of Gradshteyn and Ryzhik (1994), gives the following result:

$$-\frac{1}{2}\ln\left[\frac{8+2(1-\frac{1}{A})u+2\sqrt{16+8(1-\frac{1}{A})u-\frac{4}{A}u^2}}{u}\right] = \lambda_{\pm}t \qquad \text{Eq.A.2.4}$$

Rearranging Eq.A.2.4 to solve for u eventually leads to

$$\left(\left\{\left[\exp(2\lambda_{\mp}t)+2(1+\tfrac{1}{A})\right]^2-8\exp(2\lambda_{\mp}t)\right\}u-16\exp(2\lambda_{\mp}t)\right)u = 0 \qquad \text{Eq.A.2.5}$$

Solving for x, using $u = A(1-\cos x)$, gives either $x = 2n\pi$, or

$$x = Arc\cos\left(\frac{[\exp(\lambda_{\mp}t)-2(1+\frac{1}{A})\exp(\lambda_{\pm}t)]^2-\frac{8}{A}}{[\exp(\lambda_{\mp}t)+2(1+\frac{1}{A})\exp(\lambda_{\pm}t)]^2-8}\right) \qquad \text{Eq.A.2.6a}$$

Solving for y, using $u = \cosh y - 1$, gives either $y = 0$, or

$$y = Arc\cosh\left(\frac{[\exp(\lambda_{\mp}t)+2(1+\frac{1}{A})\exp(\lambda_{\pm}t)]^2+8}{[\exp(\lambda_{\mp}t)+2(1+\frac{1}{A})\exp(\lambda_{\pm}t)]^2-8}\right) \qquad \text{Eq.A.2.6b}$$

The denominators in Eqs.A.2.6a, b are equal. An equivalent form can be found for this denominator, which will be useful in finding the time when $x = \pi$.

$$[\exp(\lambda_{\mp}t)+2(1+\tfrac{1}{A})\exp(\lambda_{\pm}t)]^2-8 = [\exp(\lambda_{\mp}t)-2(1+\tfrac{1}{A})\exp(\lambda_{\pm}t)]^2+\tfrac{8}{A} \qquad \text{Eq.A.2.7}$$

The heteroclinic orbit is symmetric about $x = \pi$, so find the time t_π when $x = \pi$. Substitute the equivalent denominator shown in Eq.A.2.7 into Eq.A.2.a, set $x = \pi$ and $t = t_\pi$, and solve for t_π:

$$t_\pi = \frac{1}{2\lambda_{\mp}}\ln\left[\frac{2}{\sqrt{A}\lambda_{+}}\right] \qquad \text{Eq.A.2.8}$$

Define $t = t' + t_\pi$, and substitute, along with Eq.A.2.8, into Eqs.A.2.6a and b, which yields the following results for x and y:

$$x = Arc\cos\left(\frac{\sinh^2(\lambda_{+}t')-\lambda_{+}/\sqrt{A}}{\cosh^2(\lambda_{+}t')-\sqrt{A}\,\lambda_{+}}\right) \qquad \text{Eq.A.2.9a}$$

$$y = Arc\cosh\left(\frac{\cosh^2(\lambda_+ t') + \sqrt{A}\,\lambda_+}{\cosh^2(\lambda_+ t') - \sqrt{A}\,\lambda_+}\right) \qquad \text{Eq.A.2.9b}$$

Since both t and t' have domains of (-∞, ∞), the prime can be dropped. The + subscript on λ_+ will also be dropped, and $\lambda = \sqrt{A}/(1+A)$ will be used from this point forward. The heteroclinic orbit can thus be expressed as the vector:

$$\vec{\Gamma}_0(t) = \left(Arc\cos\left(\frac{\sinh^2(\lambda t) - \lambda/\sqrt{A}}{\cosh^2(\lambda t) - \sqrt{A}\,\lambda}\right),\ Arc\cosh\left(\frac{\cosh^2(\lambda t) + \sqrt{A}\,\lambda}{\cosh^2(\lambda t) - \sqrt{A}\,\lambda}\right)\right) \qquad \text{Eq.A.2.10}$$

A.3 Melnikov Function

The dynamical equations of motion for the Kelvin-Stuart Cat Eyes driven flow (perturbed system) are:

$$\dot{x} = \frac{\partial H}{\partial y} = \frac{\sinh y}{\cosh y + A\cos x} + \varepsilon\sin(\omega t) \qquad \text{Eq.A.3.1a}$$

$$\dot{y} = -\frac{\partial H}{\partial x} = \frac{A\sin x}{\cosh y + A\cos x} + \varepsilon\sin(\omega t) \qquad \text{Eq.A.3.1b}$$

Define vectors $\vec{f}$ and $\vec{g}$ as:

$$\vec{f} = \begin{pmatrix} \dfrac{\sinh y}{\cosh y + A\cos x} \\ \dfrac{A\sin x}{\cosh y + A\cos x} \end{pmatrix}, \quad \vec{g} = \begin{pmatrix} \sin(\omega t) \\ \sin(\omega t) \end{pmatrix} \qquad \text{Eq.A.3.2}$$

The Melnikov function is defined as:

$$M(t_0) = \int_{-\infty}^{\infty} dt \exp\left(-\int \vec{\nabla}\cdot\vec{f}(\vec{\Gamma}_0(s))\,ds\right)\vec{f}(\vec{\Gamma}_0(t)) \wedge \vec{g}(\vec{\Gamma}_0(t), t + t_0) \qquad \text{Eq.A.3.3}$$

The exponential term drops out of Eq.A.3.3 since $\vec{\nabla}\cdot\vec{f} = 0$ for the Kelvin Stuart Cat Eyes dynamical system.

The wedge product is defined as $\vec{f} \wedge \vec{g} = f_x g_y - f_y g_x$, so along the heteroclinic orbit, using the Pythagorean identities for sin *x* and sinh *y*, and Eqs.A.2.9a, b, the wedge product becomes

$$\vec{f}(\vec{\Gamma}_0(t)) \wedge \vec{g}(\vec{\Gamma}_0(t), t+t_0) =$$
$$\frac{1}{1+A}\left(\sqrt{\left(\frac{\cosh^2(\lambda t)+\sqrt{A}\,\lambda}{\cosh^2(\lambda t)-\sqrt{A}\,\lambda}\right)^2 - 1} - A\sqrt{1-\left(\frac{\sinh^2(\lambda t)-\lambda/\sqrt{A}}{\cosh^2(\lambda t)-\sqrt{A}\,\lambda}\right)^2}\right)\times \qquad \text{Eq.A.3.4}$$
$$(\sin(\omega t)\cos(\omega t_0)+\cos(\omega t)\sin(\omega t_0))$$

Use the equivalent denominator $\cosh^2(\lambda t) - \sqrt{A}\,\lambda = \sinh^2(\lambda t) + \lambda/\sqrt{A}$ in the second square root to perform the subtraction, take each square root, and then combine into one fraction with the original denominator to achieve the following result for the wedge product:

$$\vec{f}(\vec{\Gamma}_0(t)) \wedge \vec{g}(\vec{\Gamma}_0(t), t+t_0) =$$
$$\frac{2\sqrt{\lambda\sqrt{A}}}{1+A}\left(\frac{\cosh(\lambda t)-\sqrt{A}\sinh(\lambda t)}{\cosh^2(\lambda t)-\lambda\sqrt{A}}\right)(\sin(\omega t)\cos(\omega t_0)+\cos(\omega t)\sin(\omega t_0)) \qquad \text{Eq.A.3.5}$$

Integrate the wedge product with respect to time to get the Melnikov function:

$$M(t_0) = \frac{2\sqrt{\lambda\sqrt{A}}}{1+A}\int_{-\infty}^{\infty}\left(\frac{\cosh(\lambda t)-\sqrt{A}\sinh(\lambda t)}{\cosh^2(\lambda t)-\lambda\sqrt{A}}\right)(\sin(\omega t)\cos(\omega t_0)+\cos(\omega t)\sin(\omega t_0))\,dt \qquad \text{Eq.A.3.6}$$

Using the even and odd symmetries of the trigonometric and hypberbolic functions, and the definition of λ, reduces the Melnikov function to the difference of two integrals:

$$M(t_0) = I_1 - I_2, \quad \begin{cases} I_1 = 2\lambda\sqrt{\lambda/\sqrt{A}}\,\sin(\omega t_0)\displaystyle\int_{-\infty}^{\infty}\frac{\cosh(\lambda t)\cos(\omega t)}{\cosh^2(\lambda t)-\lambda\sqrt{A}}\,dt \\ I_2 = 2\lambda\sqrt{\lambda\sqrt{A}}\,\cos(\omega t_0)\displaystyle\int_{-\infty}^{\infty}\frac{\sinh(\lambda t)\sin(\omega t)}{\cosh^2(\lambda t)-\lambda\sqrt{A}}\,dt \end{cases} \qquad \text{Eq.A.3.7}$$

Use the identity $\cosh^2(\lambda t) = \frac{1}{2}[\cosh(2\lambda t) + 1]$ and the fact that $\lambda\sqrt{A} = A/(1+A)$ to change the form of the denominator in each integral of Eq.A.3.7 to $\frac{1}{2}[\cosh(2\lambda t) + \frac{1-A}{1+A}]$. Redefine variables so that $t' = 2\lambda t$, substitute into both integrals in Eq.A.3.7, and simplify, giving:

$$I_1 = 2\sqrt{\lambda/\sqrt{A}}\sin(\omega t_0)\int_{-\infty}^{\infty}\frac{\cosh(t/2)\cos(\omega t/2\lambda)}{\cosh(t)+\frac{1-A}{1+A}}dt \qquad \text{Eq.A.3.8a}$$

$$I_2 = 2\sqrt{\lambda\sqrt{A}}\cos(\omega t_0)\int_{-\infty}^{\infty}\frac{\sinh(t/2)\sin(\omega t/2\lambda)}{\cosh(t)+\frac{1-A}{1+A}}dt \qquad \text{Eq.A.3.8b}$$

Solve each integral using contour integration in the complex plane. Since the integrand denominators are the same, the locations of the poles for the two integrands will be identical. Setting the integrand denominator to 0 to find the poles implies that since cosh(*t*) is > 0 for all real *t*, the poles must be strictly imaginary, i.e.,

$\cosh(t_{pole}) = -\frac{1-A}{1+A} \Rightarrow \cosh(i\tau_{pole}) = -\frac{1-A}{1+A} \Rightarrow \cos(\tau_{pole}) = -\frac{1-A}{1+A}$, so that

$$t_{pole} = i((2n+1)\pi \pm \delta), \qquad \delta = Arc\cos(\frac{1-A}{1+A}) \qquad \text{Eq.A.3.9}$$

Calculate the residues (using L'Hopital's rule) for each integral using $(z\text{-}z_{pole})\,f(z)$ evaluated at $z = z_{pole}$, where $f(z)$ is the integrand for each integral, to get for I_1:

$$2\pi i\sum residues = \frac{2\pi}{\cos(\delta/2)}\cosh\left(\frac{\omega\pi}{2\lambda}\right)\cosh\left(\frac{\omega\delta}{2\lambda}\right) \qquad \text{Eq.A.3.10a}$$

and for I_2:

$$2\pi i\sum residues = \frac{2\pi}{\sin(\delta/2)}\cosh\left(\frac{\omega\pi}{2\lambda}\right)\sinh\left(\frac{\omega\delta}{2\lambda}\right) \qquad \text{Eq.A.3.10b}$$

Choose the following contour for both integrals:

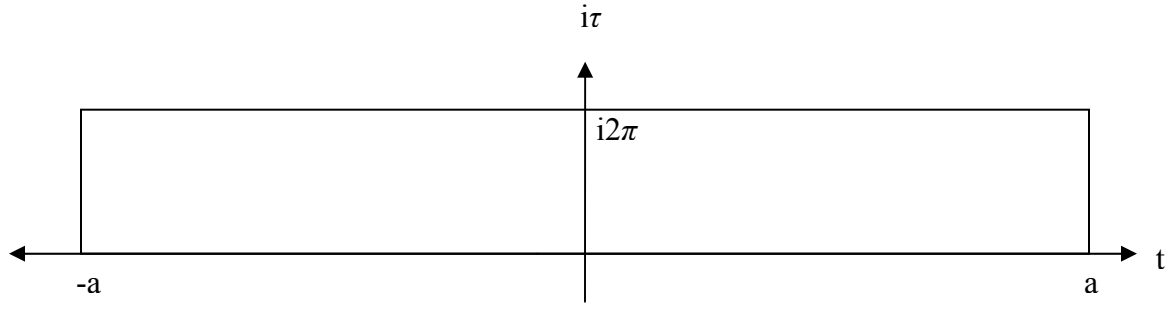

Figure A.3.1 Contour used for contour integration in the complex plane

Thus the contour integral for each integral takes the form

$$\oint_C f(z) = \int_{-a}^{a} f(t)\,dt + i\int_{0}^{2\pi} f(a+i\tau)\,d\tau + \int_{a}^{-a} f(t+i2\pi)\,dt + i\int_{2\pi}^{0} f(-a+i\tau)\,d\tau \qquad \text{Eq.A.3.11}$$

For both f_1 and f_2, and thus I_1 and I_2, the second and fourth integrals in Eq.A.3.11 can be shown to approach 0 in the limit as $a \to \infty$. Also, for both f_1 and f_2, and thus I_1 and I_2, the third integral in Eq.A.3.11 can be shown to equal $\cosh(\omega\pi/\lambda)$ times the first integral in Eq.A.3.11. Combining these results with Eq.A.3.8a and Eq.A.3.10a for I_1 gives:

$$I_1 = 2\sqrt{\lambda/\sqrt{A}}\,\sin(\omega t_0)\frac{2\pi\cosh(\omega\delta/2\lambda)\cosh(\omega\pi/2\lambda)}{\cos(\delta/2)(1+\cosh(\omega\pi/\lambda))} \qquad \text{Eq.A.3.12a}$$

For I_2 the result is:

$$I_2 = 2\sqrt{\lambda\sqrt{A}}\,\cos(\omega t_0)\frac{2\pi\sinh(\omega\delta/2\lambda)\cosh(\omega\pi/2\lambda)}{\sin(\delta/2)(1+\cosh(\omega\pi/\lambda))} \qquad \text{Eq.A.3.12b}$$

Using the half-angle formulas for cos (δ/2) and for sin (δ/2), as well as the identity $[\cosh(2x)+1] = 2\cosh^2 x$, Eqs.A.3.12a, b can be simplified to:

$$I_1 = \pm\sin(\omega t_0)2\pi\frac{\cosh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}, \qquad \delta = Arc\cos\left(\frac{1-A}{1+A}\right), \qquad \lambda = \frac{\sqrt{A}}{1+A} \qquad \text{Eq.A.3.13a}$$

$$I_2 = \pm\cos(\omega t_0)2\pi\frac{\sinh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}, \qquad \delta = Arc\cos\left(\frac{1-A}{1+A}\right), \qquad \lambda = \frac{\sqrt{A}}{1+A} \qquad \text{Eq.A.3.13b}$$

Thus, the overall Melnikov function is:

$$M(t_0) = 2\pi\left[\pm\frac{\cosh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}\sin(\omega t_0) \pm \frac{\sinh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}\cos(\omega t_0)\right] \qquad \text{Eq.A.3.14}$$

The graph of *M (t)* vs. ω and vs. *t* is shown on the next page in Figure A.3.2.

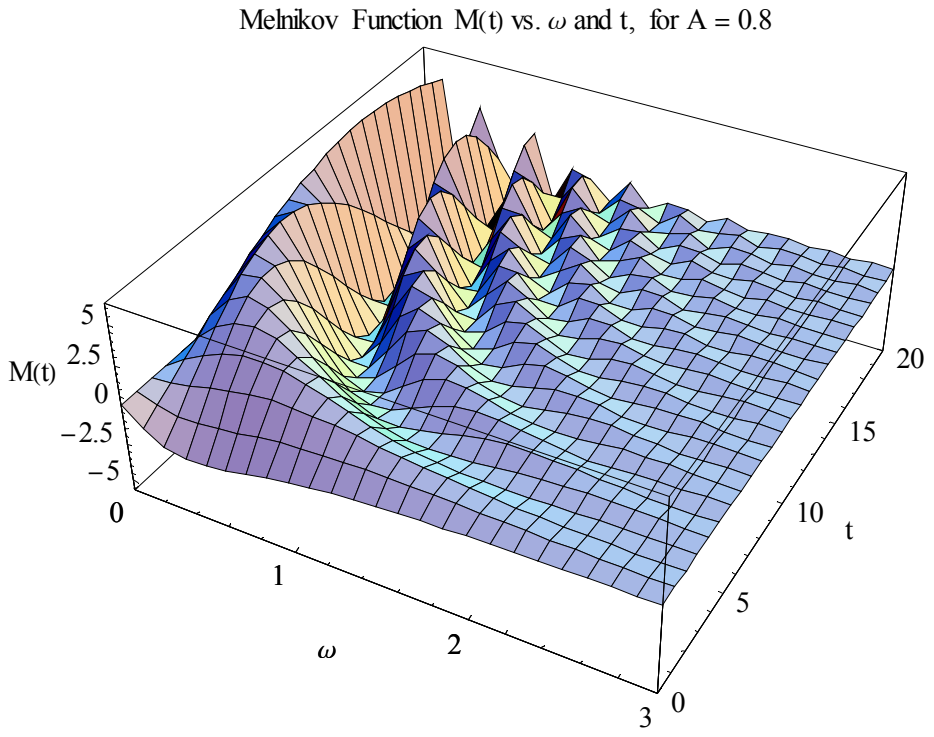

Figure A.3.2 Melnikov Function M (t) vs. ω and t, for A = 0.8, with ε = 1

The Melnikov function is graphed for several values of ω in Figure A.3.3.

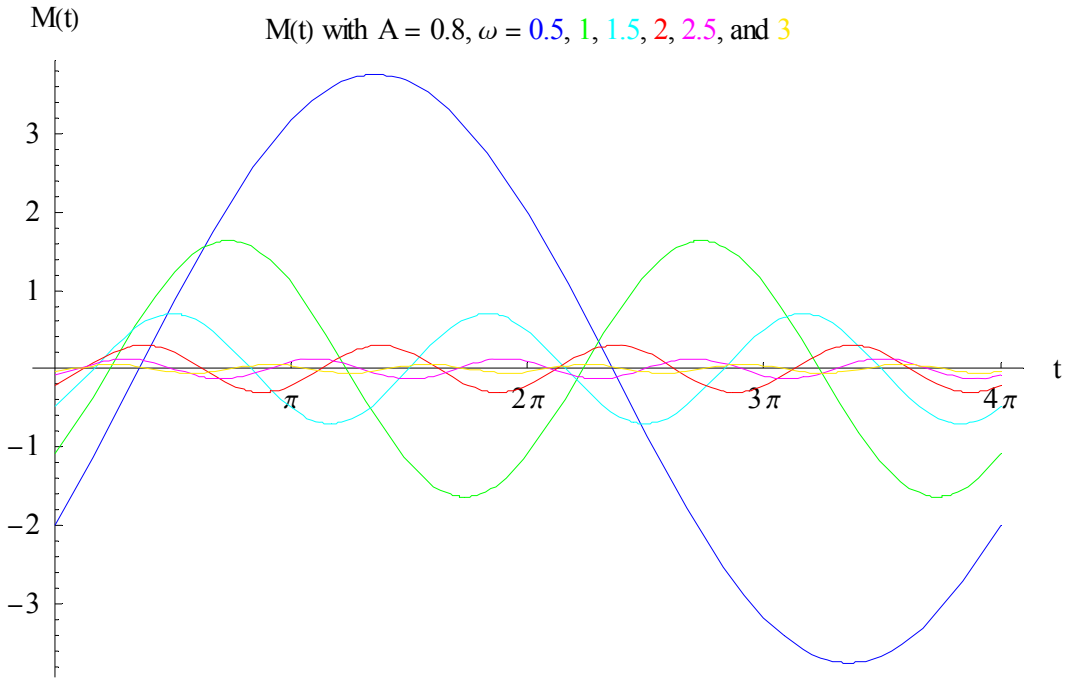

Figure A.3.3 Melnikov Function as a function of t with fixed ω, and ε = 1

The Melnikov function can be written in a simpler and more illuminating form by making the following substitutions:

$$F_1(\omega) = \frac{\cosh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}, \quad F_2(\omega) = \frac{\sinh(\omega\delta/2\lambda)}{\cosh(\omega\pi/2\lambda)}, \quad \varphi = Arc\tan\left(\frac{F_2(\omega)}{F_1(\omega)}\right),$$
$$\sin\varphi = \frac{F_2(\omega)}{\sqrt{(F_1(\omega))^2 + (F_2(\omega))^2}}, \quad \cos\varphi = \frac{F_1(\omega)}{\sqrt{(F_1(\omega))^2 + (F_2(\omega))^2}} \qquad \text{Eq.A.3.15}$$

Applying the substitutions given in Eq.A.3.15 into the expression for the Melnikov function in Eq.A.3.14, and using the trigonometric identity for the sine of the sum of an angle results in:

$$M(t_0) = \pm\, 2\pi\sqrt{(F_1(\omega))^2 + (F_2(\omega))^2}\,\sin(\omega t \pm \varphi) \qquad \text{Eq.A.3.16}$$

A graph of this simplified form of the Melnikov function is shown below in Figure A.3.4. As expected, it is exactly the same as the graph of the original Melnikov function shown in Figure A.3.3.

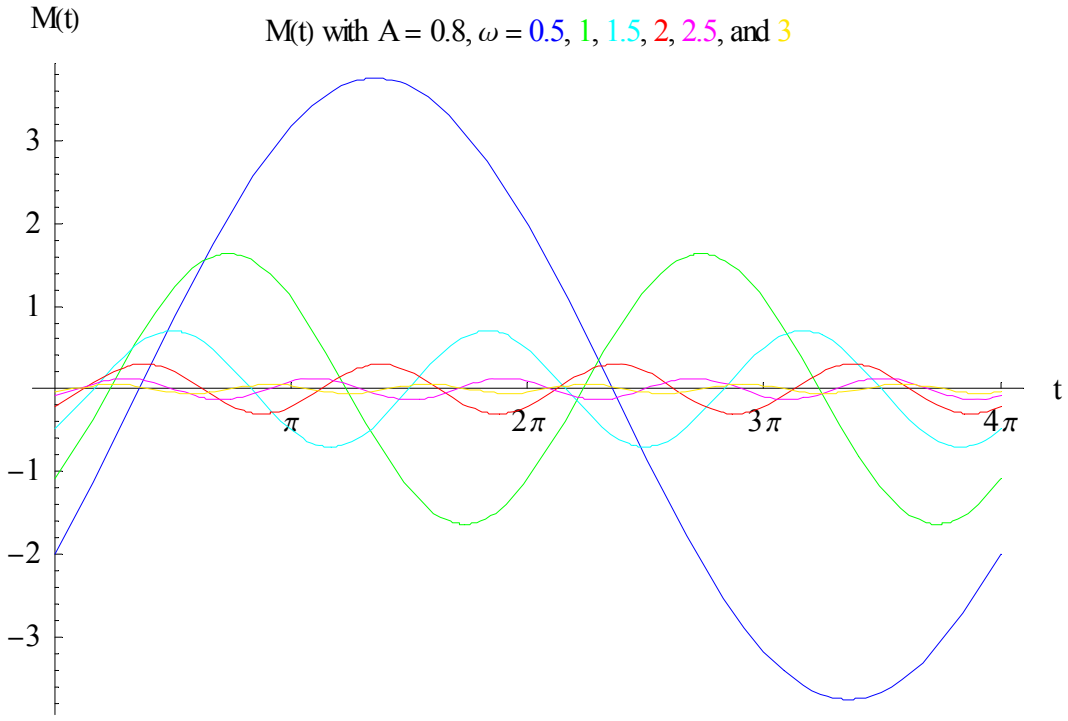

Figure A.3.4 Simplified Melnikov Function as a function of t with fixed ω, and ε = 1

The amplitude function $C(\omega) = 2\pi\sqrt{(F_1(\omega))^2 + (F_2(\omega))^2}$ for the simplified Melnikov function is graphed below in Figure A.3.5.

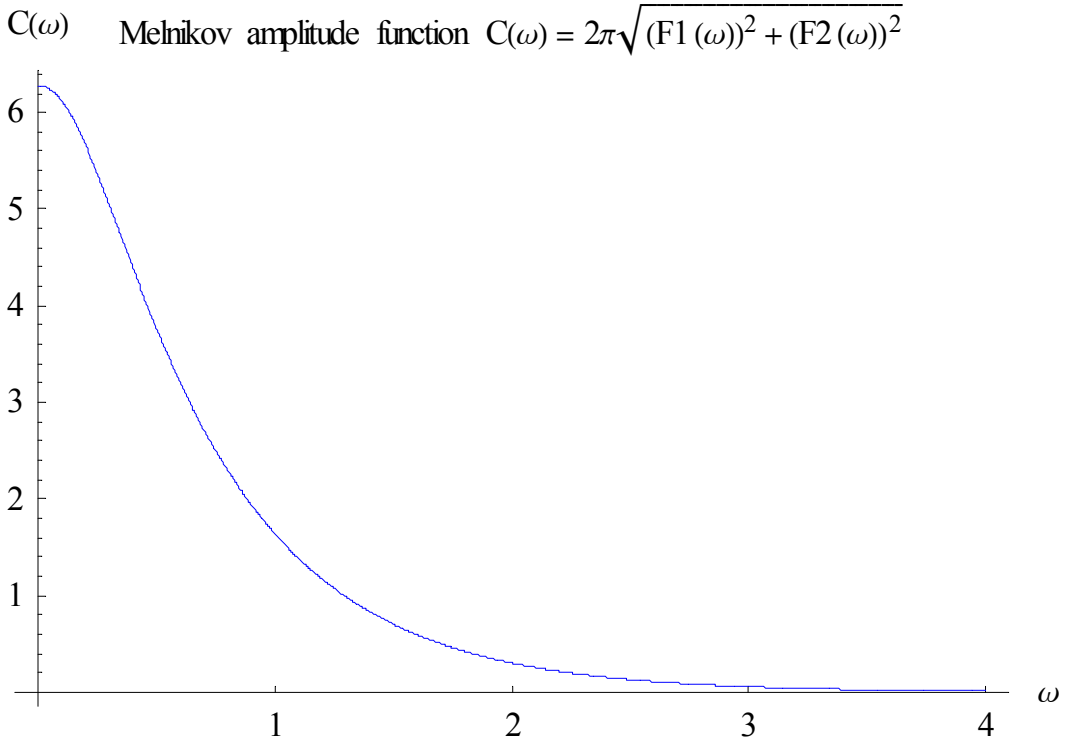

Figure A.3.5 Amplitude function for the simplified Melnikov function

A.4 Internal and External Periodic Functions

Periodic orbits are foliated inside the heteroclinic orbit of the unperturbed system, while external to the heteroclinic orbit, orbits exhibit a spatial periodicity due to the spatially periodic nature of the Kelvin-Stuart Cat Eyes dynamical system, as shown on the next page in Figure A.4.1.

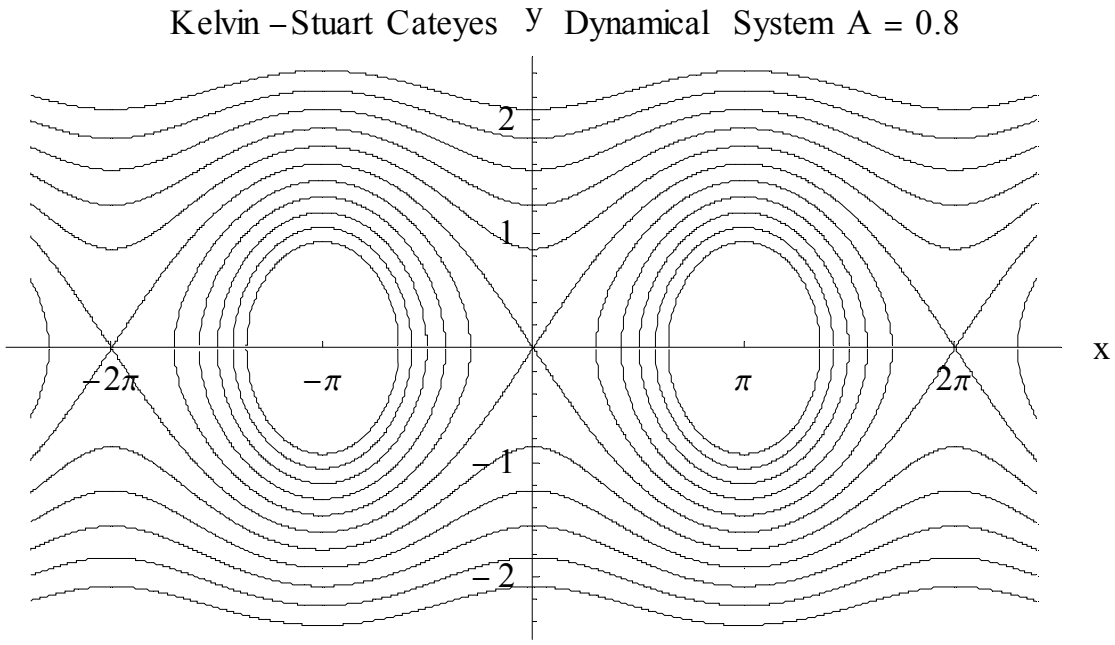

Figure A.4.1 Kelvin-Stuart Cat Eyes Dynamical System for A = 0.8, showing foliated internal periodic orbits, and external orbits exhibiting the periodicity of the system.

The Hamiltonian for the unperturbed system is

$$H(x,y) = \ln(\cosh y + A\cos x) - \ln(1+A) \qquad \text{Eq.A.4.1}$$

All orbits of the unperturbed system can be uniquely characterized by their value of $H(x, y)$. The heteroclinic orbit has $H = 0$, and separates the internal orbits, which have $H < 0$, from the external orbits, which have $H > 0$. The period for any orbit can be expressed as a function of H for both the internal and external orbits, which will be called the internal and external period functions, respectively. Separate the variables in the x equation of motion (Eq.A.1.2a) and integrate to get:

$$P_{\text{int}}(H) = \int_{t_1}^{t_2} dt = 2\int_{x_1}^{x_2} \frac{(1+A)\exp(H)}{\sqrt{[(1+A)\exp(H) - A\cos x]^2 - 1}}\,dx, \quad H < 0 \qquad \text{Eq.A.4.2a}$$

$$P_{ext}(H) = \int_{t_1}^{t_2} dt = \int_{0}^{2\pi} \frac{(1+A)\exp(H)}{\sqrt{[(1+A)\exp(H) - A\cos x]^2 - 1}}\,dx, \quad H > 0 \qquad \text{Eq.A.4.2b}$$

Figure A.4.2 on the next page shows the heteroclinic orbit and representative internal and external orbits, with the corresponding integration limits of x_1 and x_2, and 0 and 2π, respectively.

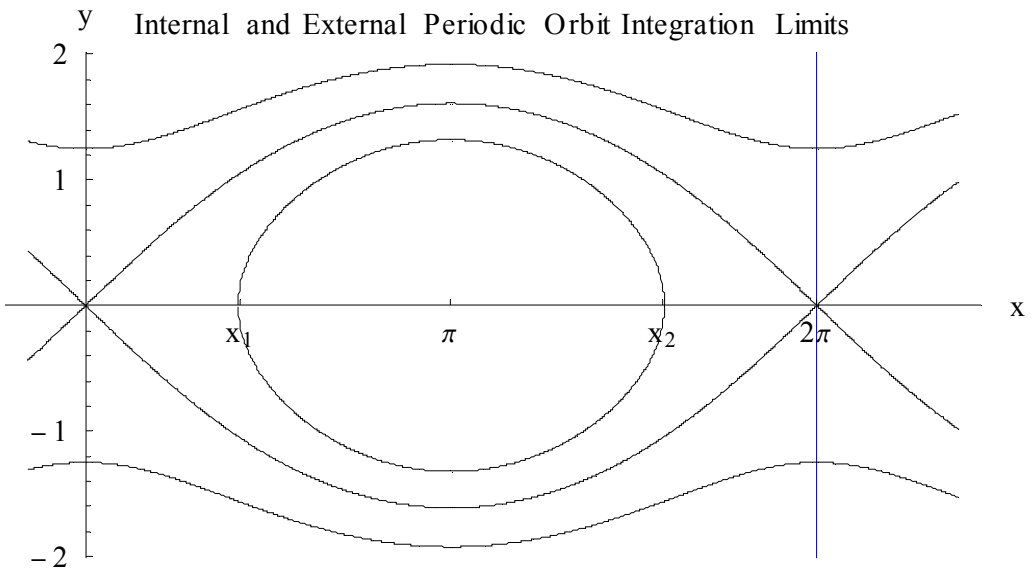

Figure A.4.2 The heteroclinic orbit, and representative internal and external orbits with the associated limits of integration for each case.

Note that both the internal and external orbits are symmetric about $x = \pi$, so that both integrals can be rewritten:

$$P_{\text{int}}(H) = 4\int_{x_1}^{\pi} \frac{(1+A)\exp(H)}{\sqrt{[(1+A)\exp(H) - A\cos x]^2 - 1}}\,dx, \quad H < 0 \qquad \text{Eq.A.4.2a}'$$

$$P_{ext}(H) = 2\int_{0}^{\pi} \frac{(1+A)\exp(H)}{\sqrt{[(1+A)\exp(H) - A\cos x]^2 - 1}}\,dx, \quad H > 0 \qquad \text{Eq.A.4.2b}'$$

Let $u = (1+A)\exp(H) - A\cos x$ for both period functions, find $\sin x$ in terms u, and express dx in terms of du. For each integral, find the limits of integration as values of u. The two integrals can now be written as:

$$P_{\text{int}}(H) = 4(1+A)\exp(H) \int_{1}^{(1+A)\exp(H)+A} \frac{du}{\sqrt{(u^2-1)\{A^2 - [(1+A)\exp(H) - u]^2\}}}, \quad H < 0$$

Eq.A.4.3a

$$P_{ext}(H) = 2(1+A)\exp(H) \int_{(1+A)\exp(H)-A}^{(1+A)\exp(H)+A} \frac{du}{\sqrt{(u^2-1)\{A^2 - [(1+A)\exp(H) - u]^2\}}}, \quad H > 0$$

Eq.A.4.3b

Note that both integrands are equal, and only the limits of integration differ for the two integrals.

The square root expression in the integrands may be rewritten as

$$\sqrt{\{[(1+A)\exp(H)+A]-u\}\{u-[(1+A)\exp(H)-A]\}(u-1)(u-(-1))} \qquad \text{Eq.A.4.4}$$

In the case of $H < 0$, the following inequalities hold for the four factors in Eq.A.4.4:

$$[(1+A)\exp(H)+A] > 1 > [(1+A)\exp(H)-A] > -1 \qquad \text{Eq.A.4.5a}$$

In the case of $H > 0$, the following inequalities hold for the four factors in Eq.A.4.4:

$$[(1+A)\exp(H)+A] > [(1+A)\exp(H)-A] > 1 > -1 \qquad \text{Eq.A.4.5b}$$

The integral for $H < 0$ can be rewritten as

$$P_{\text{int}}(H) = 4(1+A)\exp(H)\int_b^a \frac{du}{\sqrt{(a-u)(u-b)(u-c)(u-d)}}, \qquad \text{Eq.A.4.6a}$$
$$a=[(1+A)\exp(H)+A], \quad b=1, \quad c=[(1+A)\exp(H)-A], \quad d=-1$$

The integral for $H > 0$ can be rewritten as

$$P_{ext}(H) = 2(1+A)\exp(H)\int_b^a \frac{du}{\sqrt{(a-u)(u-b)(u-c)(u-d)}}, \qquad \text{Eq.A.4.6b}$$
$$a=[(1+A)\exp(H)+A], \quad b=[(1+A)\exp(H)-A], \quad c=1, \quad d=-1$$

The integral for each case is the same, and can be found in section 3.147, either formula 6 or 7 on p.290 of Gradshteyn and Ryzhik (1994), as

$$\int_b^a \frac{du}{\sqrt{(a-u)(u-b)(u-c)(u-d)}} = \frac{2}{\sqrt{(a-c)(b-d)}} F\left(\frac{\pi}{2}, \sqrt{\frac{(a-b)(c-d)}{(a-c)(b-d)}}\right) = \frac{2}{\sqrt{(a-c)(b-d)}} K\left(\sqrt{\frac{(a-b)(c-d)}{(a-c)(b-d)}}\right) \qquad \text{Eq.A.4.7}$$

K (k) is the complete elliptic integral of the first kind. Using result Eq.A.4.7 with Eqs.A.4.6a, b and the definition of $\lambda = \sqrt{A}/(1+A)$ gives the period functions as

$$P_{\text{int}}(H) = \frac{4}{\lambda}\exp(H) K\left(\frac{1}{2\lambda}\sqrt{\exp(2H)-\left(\frac{1-A}{1+A}\right)^2}\right), \quad H<0 \qquad \text{Eq.A.4.8a}$$

$$P_{ext}(H) = \frac{4\exp(H)}{\sqrt{\exp(2H)-\left(\frac{1-A}{1+A}\right)^2}} K\left(\frac{2\lambda}{\sqrt{\exp(2H)-\left(\frac{1-A}{1+A}\right)^2}}\right), \quad H>0 \qquad \text{Eq.A.4.8b}$$

It is important to note that the definitions of the elliptic integrals used in Gradshteyn and Ryzhik (1994) differ from those used by Mathematica® (1999) and many other sources, such as Arfken (1985), in that Gradshteyn and Ryzhik define the integrals in terms of the modulus, while Arfken and Mathematica® use the square of the modulus as a parameter.

The period functions are written in the definition used by Mathematica® as

$$P_{\text{int}}(H) = \frac{4}{\lambda}\exp(H)K\left(\frac{1}{4\lambda^2}\exp(2H) - \left(\frac{1-A}{1+A}\right)^2\right), \quad H < 0 \qquad \text{Eq.A.4.8a'}$$

$$P_{ext}(H) = \frac{4\exp(H)}{\sqrt{\exp(2H) - \left(\frac{1-A}{1+A}\right)^2}}K\left(\frac{4\lambda^2}{\exp(2H) - \left(\frac{1-A}{1+A}\right)^2}\right), \quad H > 0 \qquad \text{Eq.A.4.8b'}$$

It is necessary to find approximations for the two period functions so as to find approximate inverse functions. The approximations will use the definition found in Gradshteyn and Ryzhik, since the form of the approximation also comes from Gradshteyn and Ryzhik. Formula 3 of section 8.113 on p.909 in Gradshteyn and Ryzhik gives the following approximation for $K(k)$:

$$K(k) = \ln\frac{4}{k'} + \left(\frac{1}{2}\right)^2\left(\ln\frac{4}{k'} - \frac{2}{1\cdot 2}\right)k'^2 + \cdots, \quad k' = \sqrt{1-k^2} \qquad \text{Eq.A.4.9}$$

Using the first term of the approximation in Eq.A.4.9 with the two period functions in Eqs.A.4.8a, b, gives the following approximations for the period functions:

$$P_{\text{int}}(H) = \frac{2}{\lambda}\left[\ln(32\lambda^2) - \ln(-H)\right], \quad H < 0 \qquad \text{Eq.A.4.10a}$$

$$P_{ext}(H) = \frac{1}{\lambda}\left[\ln(32\lambda^2) - \ln(H)\right], \quad H > 0 \qquad \text{Eq.A.4.10b}$$

Inverting Eqs.A.4.10a, b results in the following approximate inverse functions:

$$P_{\text{int}}^{-1}(t) = -32\lambda^2\exp\left(\frac{-t}{2\lambda}\right), \quad H < 0 \qquad \text{Eq.A.4.11a}$$

$$P_{ext}^{-1}(t) = -32\lambda^2\exp\left(\frac{-t}{\lambda}\right), \quad H < 0 \qquad \text{Eq.A.4.11b}$$

Derivatives of the period function approximations are

$$P'_{\text{int}}(H) = -\frac{2}{\lambda H}, \quad H < 0 \text{ and } P'_{ext}(H) = -\frac{1}{\lambda H}, \quad H > 0 \qquad \text{Eq.A.4.12a, b}$$

Appendix B

Mathematica® Programs

B.1 Module for Calculating and Plotting Bifurcation Curves

```
(**********************************************************************
Module for calculating and plotting bifurcation curves for the Kelvin-Stuart Cat Eyes system.
The system may be symmetric about p0, i.e., the unstable manifold is at a greater distance than
the stable manifold for the first lobe, or the system may be symmetric about q0, i.e., the stable
manifold is at a greater distance than the unstable manifold for the first lobe. The variable
melsign selects which case to run, -1 for symmetric about p0, or 1 for symmetric about q0.  The
module can be modified for use with other dynamical systems by changing the functions in
the module, located just after the Functions list comment.
**********************************************************************)


runbifurcation = A = 0.8; melsign = −1; lmax = 10
(*** set initial values for A, melsign, and maximum l value ***);


(**********************************************************************************
    Module to find ϵmin and ϵmax bifurcation values for both internal and external
    lobes.  Plots are created for the min and max internal bifurcation curves
    (red is the internal min, orange the internal max),
   the min and max external curves (blue is the external min, light blue the external max),
  as well as the approximated min and max values used as beginning values to search for
    the internal and external min and max
    (magenta is the approximation max, green is the approximation min.)
      **********************************************************************)

bifurcationcurves[A_, a_, b_, l_, bifurinternalmin_, bifurinternalmax_, bifurexternalmin_,
    bifurexternalmax_, ϵapprox_, melsign_] :=
  Module[{c = A, xinit = a, xfinal = b, int = l, ω, points, ϵinternal1, ϵinternal2, ϵexternal1,
      ϵexternal2, ϵinternalmin, ϵinternalmax, ϵexternalmin, ϵexternalmax},


   (**********************************************************************
                              Module call variable list:
        A is the vorticity strength parameter in the Kelvin−Stuart Cat Eyes system,
      a is the left endpoint of the search interval, b is the right endpoint of the search interval,
      l is the lobe iteration that first intersects lobe 0, bifurinternalmin, etc.,
      are the various bifurcation curves calculated and outputted by the module for each value of l.
      **********************************************************************)
```

(**
Module internal variable list:
c = A, xinit = a, xfinal = b, int = 1, ω = frequency, points = {ω, calculated ϵ},
ϵinternal1 = approximate internal min ϵ, ϵinternal2 = approximate internal max ϵ,
ϵexternal1 = approximate external min ϵ, ϵexterna2 = approximate external max ϵ,
ϵinternalmin = calculated internal min ϵ, ϵinternalmax = calculated internal max ϵ,
ϵexternalmin = calculated external min ϵ, ϵexternamax = calculated external max ϵ.
**)

(**
Function list:
F1[ω], F2[ω] are the functions that constitute the amplitude function of the Kelvin–Stuart Cat Eyes Melnikov function,
melnikov[ω, t] is the Kelvin–Stuart Cat Eyes Melnikov function,
periodinternal[h] and periodexternal[h] are the internal and external period functions,
ϵapproxinternal[ω, t] and ϵapproxexternal[ω, t] calculate approximate ϵ values for the internal and external lobes.
**)

$$F1[\omega\_] := \frac{\text{Cosh}\left[\frac{1+c}{2\sqrt{c}}\,\omega\,\text{ArcCos}\left[\frac{1-c}{1+c}\right]\right]}{\text{Cosh}\left[\frac{1+c}{2\sqrt{c}}\,\omega\pi\right]};$$

$$F2[\omega\_] := \frac{\text{Sinh}\left[\frac{1+c}{2\sqrt{c}}\,\omega\,\text{ArcCos}\left[\frac{1-c}{1+c}\right]\right]}{\text{Cosh}\left[\frac{1+c}{2\sqrt{c}}\,\omega\pi\right]};$$

$$\text{melnikov}[\omega\_, t\_] := \text{melsign}\, 2\pi\sqrt{(F1[\omega])^2 + (F2[\omega])^2}\,\text{Sin}[\omega t];$$

$$\text{periodinternal}[h\_] := \frac{2(1+c)}{\sqrt{c}}\,\text{Exp}[h]\,\text{EllipticK}\left[\frac{(1+c)^2}{4c}\left(\text{Exp}[2h] - \left(\frac{1-c}{1+c}\right)^2\right)\right];$$

$$\text{periodexternal}[h\_] := \frac{4\,\text{Exp}[h]}{\sqrt{\left(\text{Exp}[2h] - \left(\frac{1-c}{1+c}\right)^2\right)}}\,\text{EllipticK}\left[\frac{4c}{(1+c)^2}\,\frac{1}{\left(\text{Exp}[2h] - \left(\frac{1-c}{1+c}\right)^2\right)}\right];$$

$$\text{extraint} = (1 + \text{melsign})/2;$$

$$\text{extraext} = (1 - \text{melsign})/2;$$

$$\epsilon\text{approxinternal}[\omega\_, t\_] := \left(-32\,\frac{c}{(1+c)^2}\,\text{Exp}\left[-\frac{\sqrt{c}}{(1+c)}\left((\text{int} + \text{extraint} + 1)\,\frac{2\pi}{\omega} - 2t\right)\right]\right) \Big/ \text{melnikov}[\omega, t];$$

$$\epsilon\text{approxexternal}[\omega\_, t\_] := \left(32\,\frac{c}{(1+c)^2}\,\text{Exp}\left[-\frac{\sqrt{c}}{(1+c)}\left((\text{int} + \text{extraext} + 1)\,\frac{2\pi}{\omega} - 2t\right)\right]\right) \Big/ \text{melnikov}[\omega, t];$$

```
(***************** Calculate the internal minimum bifurcation curve *****************)


ϵinternal1 = ϵapproxinternal[ω, 1/ω (extraint * π + ArcTan[(1 + c)/√c, ω/2])];
points =
  Table[
    {ω,
      ϵinternalmin /.
        FindRoot[
          periodinternal[(ϵinternalmin) melnikov[ω, 1/ω (extraint * π + ArcTan[(1 + c)/√c, ω/2])]] ==
            (int + extraint + 1) 2π/ω - 2(1/ω (extraint * π + ArcTan[(1 + c)/√c, ω/2])),
          {ϵinternalmin, ϵinternal1, 0, 125.0}]}, {ω, xinit, xfinal, 0.001}];

bifurinternalmin = ListPlot[points, PlotJoined → True, AxesOrigin → {0, 0}, PlotRange → {0, 1.0},
    PlotStyle → {RGBColor[1, 0, 0]}, DisplayFunction → Identity];


(***************** Calculate the internal maximum bifurcation curve *****************)


ϵinternal2 = ϵapproxinternal[ω, extraint * π/ω + π/(2ω)];
points =
  Table[
    {ω,
      ϵinternalmax /.
        FindRoot[periodinternal[(ϵinternalmax) melnikov[ω, extraint * π/ω + π/(2ω)]] ==
            (int + extraint + 1) 2π/ω - 2(extraint * π/ω + π/(2ω)), {ϵinternalmax, ϵinternal2, 0, 125.0}]},
    {ω, xinit, xfinal, 0.001}];

bifurinternalmax = ListPlot[points, PlotJoined → True, AxesOrigin → {0, 0}, PlotRange → {0, 1.0},
    PlotStyle → {RGBColor[1, 0.75, 0]}, DisplayFunction → Identity];
```

```
(***************** Calculate the external minimum bifurcation curve ****************)


ϵexternal1 = ϵapproxexternal[ω, 1/ω (extraext * π + ArcTan[(1 + c)/Sqrt[c] ω/2])];
points =
  Table[
    {ω,
      ϵexternalmin /.
        FindRoot[
          periodexternal[(ϵexternalmin) melnikov[ω, 1/ω (extraext * π + ArcTan[(1 + c)/Sqrt[c] ω/2])]] ==
            (int + extraext + 1) 2π/ω - 2/ω (extraext * π + ArcTan[(1 + c)/Sqrt[c] ω/2]),
          {ϵexternalmin, ϵexternal1, 0, 125.0}]}, {ω, xinit, xfinal, 0.001}];

bifurexternalmin = ListPlot[points, PlotJoined → True, AxesOrigin → {0, 0}, PlotRange → {0, 1.0},
    PlotStyle → {RGBColor[0, 0, 1]}, DisplayFunction → Identity];


(***************** Calculate the external maximum bifurcation curve ****************)


ϵexternal2 = ϵapproxexternal[ω, extraext * π/ω + π/(2ω)];
points =
  Table[
    {ω,
      ϵexternalmax /.
        FindRoot[periodexternal[(ϵexternalmax) melnikov[ω, extraext * π/ω + π/(2ω)]] ==
            (int + extraext + 1) 2π/ω - 2(extraext * π/ω + π/(2ω)), {ϵexternalmax, ϵexternal2, 0, 125.0}]},
    {ω, xinit, xfinal, 0.001}];

bifurexternalmax = ListPlot[points, PlotJoined → True, AxesOrigin → {0, 0}, PlotRange → {0, 1.0},
    PlotStyle → {RGBColor[0, 0.75, 1]}, DisplayFunction → Identity];
```

```
(****************** Calculate the approximation bifurcation curves ******************)

   ϵapprox  =  Plot[{ϵexternal2,ϵexternal1},{ω,0,xfinal}, PlotRange→{0,1.0},
PlotStyle→{{RGBColor [1,0,1]},{RGBColor [0,1,0]}}, PlotPoints→1000,
DisplayFunction→Identity];


(****************** Print message indicating module finished ********************)

   Print ["Finished bifurcation module for l = ", int]];


(****************** Loop to calculate search intervals for each l value ******************)

Table[{xinit[i] = (Floor [i]+1.0)/10.0, xfinal[i] = Ceiling[(i+0.5)]/2+0.5}, {i, 0, lmax}];


(****************** Clear variable values before new module call ********************)

Clear[bifurinternalmin, bifurinternalmax, bifurexternalmin, bifurexternalmax, ϵapprox];


(******** Loop to call module for all l values from 0 to chosen maximum l (lmax) *********)

Do [bifurcationcurves[0.8, xinit[l], xfinal[l], l, bifurinternalmin[l], bifurinternalmax[l],
bifurexternalmin[l], bifurexternalmax[l], ϵapprox[l], melsign], {l, 0, lmax}]; //Timing


(******************** Plot all bifurcation curves for all l values **********************)

Show [Table[{bifurinternalmin[l], bifurinternalmax[l], bifurexternalmin[l], bifurexternalmax[l],
ϵapprox[l]},{l,0,lmax}], PlotRange→{{0,4},{0,0.1}}, DisplayFunction→$DisplayFunction,
AxesOrigin→{0,0}, Frame→True, PlotLabel→"Bifurcation Curves for l = 0 to l = 10",
TextStyle→{FontSize→16}, FrameLabel→{"ω","ϵ"}, RotateLabel→False]
```

B.2 Topological Approximation Method Module

(**********

Module for applying the Topological Approximation Method (TAM) to a dynamical system , here, the Kelvin−Stuart Cat Eyes system..

The system may be symmetric about p0, i.e., the unstable manifold is at a greater distance than the stable manifold for the first lobe, or the system may be symmetric about q0, i.e., the stable manifold is at a greater distance than the unstable manifold for the first lobe.

The variable melsign selects which case to run, −1 for symmetric about p0, or 1 for symmetric about q0.

The module performs several calculations, in the following order:

1. For a given ϵ, l, and initial value for ω (called ω_0) calculate the minimum (using the minimum internal value) and maximum (using the maximum external value) ω values and an ω value anywhere in the interval between the minimum and maximum for a given partition value (called ω int).
2. Find the intersection points (times and iterates) for internal lobes L, L + 1, L+ 2 and the outer and inner strips of lobe 2 L + 1 that intersect with external lobe 0.
3. Find the intersection points (times and iterates) for external lobes L, L + 1, L+ 2 and the outer and inner strips of lobe 2 L + 1 that intersect with internal lobe 0.
4. Calculate the area of a lobe, and the areas of intersection for lobes L, L + 1, L+ 2 and the outer and inner strips of lobe 2 L + 1, for both internal and external lobes.
5. Use the areas calculated in step 4 to calculate the internal and external weighting factors.
6. Calculate the lobe dynamics transfer matrix and the initial vector.
7. Use the l matrix and the initial vector to calculate the escape rate at iteration n.

Note that if an asymmetry term is introduced into the dynamical system , most of the steps above must be modified to include the effects of the asymmetry term.

The module can be modified for use with other dynamical systems by changing the functions in the module located just after the Functions list comment.

**********)

```
(*************************************************************************)
(* Setting of global parameters – A is the vorticity strength parameter,
  lmax = maximum value of l where l is the value of the first lobe that intersects with lobe 0,
  melsign selects the orientation of the lobes, ϵinternal0 = ϵexternal0 is the chosen value of ϵ,
  nend is the number of iterations and ω int is an integer between 0 and 100 that selects
      the value of ω – 0 is the min, 100 is the max. Clear ω and ϵinternal. *)
(*************************************************************************)

A = 0.8; lmax = 10; melsign = −1; ϵinternal0 = 0.01; ϵexternal0 = 0.01; nend = 100;
ωint = 50; Clear[ω, ϵinternal]; l = 2; ω0 = 1.25

(*************************************************************************
                              Function list:
      F1[ω] , F2 [ω] are the functions that constitute the amplitude function of the Kelvin–
    Stuart Cat Eyes Melnikov function,
  melnikov[ω , t] is the Kelvin–Stuart Cat Eyes Melnikov function,
  periodinternal[h] and periodexternal[h] are the internal and external period functions,
  ϵapproxinternal [ω , t] and ϵapproxexternal[ω ,t] calculate approximate ϵ values for the
    internal and external lobes.
 *************************************************************************)

F1[ω_] := Cosh[(1+A)/(2√A) ω ArcCos[(1−A)/(1+A)]] / Cosh[(1+A)/(2√A) ω π];

F2[ω_] := Sinh[(1+A)/(2√A) ω ArcCos[(1−A)/(1+A)]] / Cosh[(1+A)/(2√A) ω π];

melnikov[ω_, t_] := melsign 2π √((F1[ω])^2 + (F2[ω])^2) Sin[ω t];

periodinternal[h_] := 2(1 + A)/√A Exp[h] EllipticK[(1 + A)^2/(4 A) (Exp[2 h] − ((1 − A)/(1 + A))^2)];

periodexternal[h_] := 4 Exp[h] / √(Exp[2 h] − ((1−A)/(1+A))^2) EllipticK[4 A/(1 + A)^2, 1/(Exp[2 h] − ((1−A)/(1+A))^2)];
```

```
extraint = (1 + melsign) / 2;

extraext = (1 − melsign) / 2;

ϵapproxinternal[t_] := (−32 A/(1 + A)^2 Exp[−√A/(1 + A) (1 + extraint + 1) 2π/ω0 − 2t]) / melnikov[ω0, t];

ϵapproxexternal[t_] := (32 A/(1 + A)^2 Exp[−√A/(1 + A) (1 + extraext + 1) 2π/ω0 − 2t])
    / melnikov[ω0, t];

(*****************************************************************************)
(* Calculate the maximum, minimum, and working values of ω *)
(*****************************************************************************)

ωmax =
  ω /. FindRoot[periodexternal[(ϵexternal0) melnikov[ω, 1/ω (extraext * π + ArcTan[(1 + A)/√A ω/2])]]
      == (1 + extraint + 1) 2π/ω − 2(1/ω (extraint * π + ArcTan[(1 + A)/√A ω/2])), {ω, ω0, 0, 5.0}];
ωmin =
  ω /. FindRoot[periodinternal[(ϵinternal0) melnikov[ω, extraint * π/ω + π/(2ω)]] ==
      (1 + extraint + 1) 2π/ω − 2(extraint * π/ω + π/(2ω)), {ω, ω0, 0, 5.0}];

ωdelta = (ωmax − ωmin) / 100;
ω = ωmin + ωint * ωdelta;
ϵinternal = ϵinternal0; ϵexternal = ϵexternal0;

Print["ω min = ", ωmin, "  ω max = ", ωmax, "  ω = ", ω, "  ϵ = ", ϵinternal]
```

```
(*************************************************************************)
(* Find intersection points for internal lobe l *)
(*************************************************************************)

Print ["Intersection points for internal lobes"]

{intlzero10, intlzero11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, (extraint) π/ω + π/(18 ω), (extraint) π/ω, (extraint) π/ω + π/(2 ω)},
    {t1, ((extraint) 1/2 + 1 + 5/6) 2π/ω, ((extraint) 1/2 + 1) 2π/ω, ((extraint) 1/2 + 1 + 1) 2π/ω}];
{intlzero10, intlzero11}

{intlzero20, intlzero21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(2 ω), extraint * π/ω, extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 1 + 3/4) 2π/ω, (extraint * 1/2 + 1) 2π/ω, (extraint * 1/2 + 1 + 1) 2π/ω}];
{intlzero20, intlzero21}

(*************************************************************************)
(* Find intersection points for internal lobe l+1 *)
(*************************************************************************)

{intl1zero10, intl1zero11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(18 ω), extraint * π/ω, extraint * π/ω + π/(2 ω)},
    {t1, (extraint * 1/2 + 1 + 15/8) 2π/ω, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω}];
intl1h1 = (ϵinternal) melnikov[ω, intl1zero10];
{intl1zero10, intl1zero11, intl1h1}
```

```
{intl1zero20, intl1zero21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(18 ω), extraint * π/ω, extraint * π/ω + π/(2 ω)},
    {t1, (extraint * 1/2 + 1 + 13/8) 2π/ω, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω}];
intl1h2 = (ϵinternal) melnikov[ω, intl1zero20]; {intl1zero20, intl1zero21, intl1h2}

{intl1zero30, intl1zero31} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + 11π/(18 ω), extraint * π/ω + π/(2 ω), extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 1 + 13/8) 2π/ω, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω}];
intl1h3 = (ϵinternal) melnikov[ω, intl1zero30];
{intl1zero30, intl1zero31, intl1h3}

{intl1zero40, intl1zero41} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + 11π/(18 ω), extraint * π/ω + π/(2 ω), extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 1 + 15/8) 2π/ω, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω}];
intl1h4 = (ϵinternal) melnikov[ω, intl1zero40]; {intl1zero40, intl1zero41, intl1h4}

(**************************************************************************)
(* Find intersection points for internal lobe l+2 *)
(**************************************************************************)
```

```
{intl2zero10, intl2zero11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(18 ω), extraint * π/ω, extraint * π/ω + π/(9 ω)},
    {t1, (extraint * 1/2 + 1 + 17/6) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω, (extraint * 1/2 + 1 + 3) 2π/ω}];
intl2h1 = (ϵinternal) melnikov[ω, intl2zero10]; {intl2zero10, intl2zero11, intl2h1}

{intl2zero20, intl2zero21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(18 ω), extraint * π/ω, extraint * π/ω + π/(9 ω)},
    {t1, (extraint * 1/2 + 1 + 5/2) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω, (extraint * 1/2 + 1 + 3) 2π/ω}];
intl2h2 = (ϵinternal) melnikov[ω, intl2zero20]; {intl2zero20, intl2zero21, intl2h2}

{intl2zero30, intl2zero31} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + 11π/(18 ω), extraint * π/ω + π/(2 ω), extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 1 + 5/2) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω, (extraint * 1/2 + 1 + 3) 2π/ω}];
intl2h3 = (ϵinternal) melnikov[ω, intl2zero30]; {intl2zero30, intl2zero31, intl2h3}

{intl2zero40, intl2zero41} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + 11π/(18 ω), extraint * π/ω + π/(2 ω), extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 1 + 17/6) 2π/ω, (extraint * 1/2 + 1 + 2) 2π/ω, (extraint * 1/2 + 1 + 3) 2π/ω}];
intl2h4 = (ϵinternal) melnikov[ω, intl2zero40];
{intl2zero40, intl2zero41, intl2h4}
```

```
(*************************************************************************)
(* Find outer intersection points for internal lobe 2l+1 *)
(*************************************************************************)

{int2l1outer10, int2l1outer11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(10000 ω), extraint * π/ω, extraint * π/ω + π/(2 ω)},
    {t1, (extraint * 1/2 + 2 l + 11/6) 2π/ω, (extraint * 1/2 + 2 l + 1) 2π/ω,
      (extraint * 1/2 + 2 l + 2) 2π/ω}]; int2l1outerh1 = (ϵinternal) melnikov[ω, int2l1outer10];
{int2l1outer10, int2l1outer11, int2l1outerh1}

{int2l1outer20, int2l1outer21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + π/(10000 ω), extraint * π/ω, extraint * π/ω + π/(2 ω)},
    {t1, (extraint * 1/2 + 2 l + 3/2) 2π/ω, (extraint * 1/2 + 2 l + 1) 2π/ω,
      (extraint * 1/2 + 2 l + 2) 2π/ω}]; int2l1outerh2 = (ϵinternal) melnikov[ω, int2l1outer20];
{int2l1outer20, int2l1outer21, int2l1outerh2}

{int2l1outer30, int2l1outer31} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + 9999 π/(10000 ω), extraint * π/ω + π/(2 ω), extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 2 l + 3/2) 2π/ω, (extraint * 1/2 + 2 l + 1) 2π/ω,
      (extraint * 1/2 + 2 l + 2) 2π/ω}]; int2l1outerh3 = (ϵinternal) melnikov[ω, int2l1outer30];
{int2l1outer30, int2l1outer31, int2l1outerh3}
```

```
{int2l1outer40, int2l1outer41} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraint * π/ω + 9999π/(10000ω), extraint * π/ω + π/(2ω), extraint * π/ω + π/ω},
    {t1, (extraint * 1/2 + 21 + 11/6) 2π/ω, (extraint * 1/2 + 21 + 1) 2π/ω,
      (extraint * 1/2 + 21 + 2) 2π/ω}]; int2l1outerh4 = (ϵinternal) melnikov[ω, int2l1outer40];
{int2l1outer40, int2l1outer41, int2l1outerh4}

(*************************************************************************)
(* Find inner intersection points for internal lobe 2l+1 *)
(*************************************************************************)

{int2l1inner10, int2l1inner11, int2l1inner12} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
      t2 == t1 + periodinternal[(ϵinternal) melnikov[ω, t0] + (ϵinternal) melnikov[ω, t1]],
      melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
     {t0, 0.5 (intl1zero20 + intlzero10), intl1zero20, intlzero10},
     {t1, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 1/2) 2π/ω,
      (extraint * 1/2 + 1 + 3/2) 2π/ω}, {t2, (extraint * 1/2 + 21 + 13/8) 2π/ω, (extraint * 1/2 + 21 + 3/2) 2π/ω,
      (extraint * 1/2 + 21 + 7/4) 2π/ω}]; int2l1innerh11 = (ϵinternal) melnikov[ω, int2l1inner10];
int2l1innerh21 = int2l1innerh11 + (ϵinternal) melnikov[ω, int2l1inner11];
{int2l1inner10, int2l1inner11, int2l1inner12, int2l1innerh11, int2l1innerh21}
```

```
{int2l1inner20, int2l1inner21, int2l1inner22} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodinternal[(∈internal) melnikov[ω, t0]],
        t2 == t1 + periodinternal[(∈internal) melnikov[ω, t0] + (∈internal) melnikov[ω, t1]],
        melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
      {t0, 0.5 (intl1zero20 + intlzero10), intl1zero20, intlzero10},
      {t2, (extraint * 1/2 + 21 + 15/8) 2π/ω, (extraint * 1/2 + 21 + 7/4) 2π/ω,
        (extraint * 1/2 + 21 + 2) 2π/ω}, {t1, (extraint * 1/2 + 1 + 1) 2π/ω,
        (extraint * 1/2 + 1 + 1/2) 2π/ω, (extraint * 1/2 + 1 + 3/2) 2π/ω}];
int2l1innerh12 = (∈internal) melnikov[ω, int2l1inner20];
int2l1innerh22 = int2l1innerh12 + (∈internal) melnikov[ω, int2l1inner21];
{int2l1inner20, int2l1inner21, int2l1inner22, int2l1innerh21, int2l1innerh22}

{int2l1inner30, int2l1inner31, int2l1inner32} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodinternal[(∈internal) melnikov[ω, t0]],
        t2 == t1 + periodinternal[(∈internal) melnikov[ω, t0] + (∈internal) melnikov[ω, t1]],
        melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
      {t0, 0.5 (intl1zero30 + intlzero20), intlzero20, intl1zero30},
      {t1, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 1/2) 2π/ω,
        (extraint * 1/2 + 1 + 3/2) 2π/ω}, {t2, (extraint * 1/2 + 21 + 15/8) 2π/ω, (extraint * 1/2 + 21 + 7/4) 2π/ω,
        (extraint * 1/2 + 21 + 2) 2π/ω}]; int2l1innerh13 = (∈internal) melnikov[ω, int2l1inner30];
int2l1innerh23 = int2l1innerh13 + (∈internal) melnikov[ω, int2l1inner31];
{int2l1inner30, int2l1inner31, int2l1inner32, int2l1innerh13, int2l1innerh23}
```

```
{int2l1inner40, int2l1inner41, int2l1inner42} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodinternal[(ϵinternal) melnikov[ω, t0]],
        t2 == t1 + periodinternal[(ϵinternal) melnikov[ω, t0] + (ϵinternal) melnikov[ω, t1]],
        melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
      {t0, 0.5 (intl1zero30 + intlzero20), intlzero20, intl1zero30},
      {t1, (extraint * 1/2 + 1 + 1) 2π/ω, (extraint * 1/2 + 1 + 1/2) 2π/ω,
        (extraint * 1/2 + 1 + 3/2) 2π/ω}, {t2, (extraint * 1/2 + 21 + 13/8) 2π/ω, (extraint * 1/2 + 21 + 3/2) 2π/ω,
        (extraint * 1/2 + 21 + 7/4) 2π/ω}]; int2l1innerh14 = (ϵinternal) melnikov[ω, int2l1inner40];
int2l1innerh24 = int2l1innerh14 + (ϵinternal) melnikov[ω, int2l1inner41];
{int2l1inner40, int2l1inner41, int2l1inner42, int2l1innerh14, int2l1innerh24}

(*************************************************************************)
(* Find intersection points for external lobe 1 *)
(*************************************************************************)

Print ["Intersection points for external lobes"]

{extlzero10, extlzero11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(18 ω), extraext * π/ω, extraext * π/ω + π/(2 ω)},
    {t1, (extraext * 1/2 + 1 + 5/6) 2π/ω, (extraext * 1/2 + 1) 2π/ω, (extraext * 1/2 + 1 + 1) 2π/ω}];
{extlzero10, extlzero11}

{extlzero20, extlzero21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(2 ω), extraext * π/ω, extraext * π/ω + π/ω},
    {t1, (extraext * 1/2 + 1 + 3/4) 2π/ω, (extraext * 1/2 + 1) 2π/ω, (extraext * 1/2 + 1 + 1) 2π/ω}];
{extlzero20, extlzero21}
```

```
(*************************************************************************)
(* Find intersection points for external lobe l+1 *)
(*************************************************************************)

{extl1zero10, extl1zero11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(18 ω), extraext * π/ω, extraext * π/ω + π/(2 ω)},
    {t1, (extraext * 1/2 + 1 + 15/8) 2π/ω, (extraext * 1/2 + 1 + 1) 2π/ω, (extraext * 1/2 + 1 + 2) 2π/ω}];
extl1h1 = (ϵexternal) melnikov[ω, extl1zero10]; {extl1zero10, extl1zero11, extl1h1}

{extl1zero20, extl1zero21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(18 ω), extraext * π/ω, extraext * π/ω + π/(2 ω)},
    {t1, (extraext * 1/2 + 1 + 13/8) 2π/ω, (extraext * 1/2 + 1 + 1) 2π/ω, (extraext * 1/2 + 1 + 2) 2π/ω}];
extl1h2 = (ϵexternal) melnikov[ω, extl1zero20]; {extl1zero20, extl1zero21, extl1h2}

{extl1zero30, extl1zero31} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + 11π/(18 ω), extraext * π/ω + π/(2 ω), extraext * π/ω + π/ω},
    {t1, (extraext * 1/2 + 1 + 13/8) 2π/ω, (extraext * 1/2 + 1 + 1) 2π/ω, (extraext * 1/2 + 1 + 2) 2π/ω}];
extl1h3 = (ϵexternal) melnikov[ω, extl1zero30]; {extl1zero30, extl1zero31, extl1h3}

{extl1zero40, extl1zero41} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + 11π/(18 ω), extraext * π/ω + π/(2 ω), extraext * π/ω + π/ω},
    {t1, (extraext * 1/2 + 1 + 15/8) 2π/ω, (extraext * 1/2 + 1 + 1) 2π/ω, (extraext * 1/2 + 1 + 2) 2π/ω}];
extl1h4 = (ϵexternal) melnikov[ω, extl1zero40];
{extl1zero40, extl1zero41, extl1h4}
```

```
(*************************************************************************)
(* Find intersection points for external lobe l+2 *)
(*************************************************************************)

{extl2zero10, extl2zero11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(18 ω), extraext * π/ω, extraext * π/ω + π/(9 ω)},
    {t1, {extraext * 1/2 + 1 + 17/6} 2π/ω, {extraext * 1/2 + 1 + 2} 2π/ω, {extraext * 1/2 + 1 + 3} 2π/ω}];
extl2h1 = (ϵexternal) melnikov[ω, extl2zero10]; {extl2zero10, extl2zero11, extl2h1}

{extl2zero20, extl2zero21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(18 ω), extraext * π/ω, extraext * π/ω + π/(9 ω)},
    {t1, {extraext * 1/2 + 1 + 5/2} 2π/ω, {extraext * 1/2 + 1 + 2} 2π/ω, {extraext * 1/2 + 1 + 3} 2π/ω}];
extl2h2 = (ϵexternal) melnikov[ω, extl2zero20]; {extl2zero20, extl2zero21, extl2h2}

{extl2zero30, extl2zero31} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + 11π/(18 ω), extraext * π/ω + π/(2 ω), extraext * π/ω + π/ω},
    {t1, {extraext * 1/2 + 1 + 5/2} 2π/ω, {extraext * 1/2 + 1 + 2} 2π/ω, {extraext * 1/2 + 1 + 3} 2π/ω}];
extl2h3 = (ϵexternal) melnikov[ω, extl2zero30]; {extl2zero30, extl2zero31, extl2h3}

{extl2zero40, extl2zero41} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + 11π/(18 ω), extraext * π/ω + π/(2 ω), extraext * π/ω + π/ω},
    {t1, {extraext * 1/2 + 1 + 17/6} 2π/ω, {extraext * 1/2 + 1 + 2} 2π/ω, {extraext * 1/2 + 1 + 3} 2π/ω}];
extl2h4 = (ϵexternal) melnikov[ω, extl2zero40];
{extl2zero40, extl2zero41, extl2h4}
```

```
(*************************************************************************)
(* Find outer intersection points for external lobe 2l+1 *)
(*************************************************************************)

{ext2l1outer10, ext2l1outer11} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(10000 ω), extraext * π/ω, extraext * π/ω + π/(2 ω)},
    {t1, (extraext * 1/2 + 2 l + 11/6) 2 π/ω, (extraext * 1/2 + 2 l + 1) 2 π/ω,
      (extraext * 1/2 + 2 l + 2) 2 π/ω}]; ext2l1outerh1 = (ϵexternal) melnikov[ω, ext2l1outer10];
{ext2l1outer10, ext2l1outer11, ext2l1outerh1}

{ext2l1outer20, ext2l1outer21} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + π/(10000 ω), extraext * π/ω, extraext * π/ω + π/(2 ω)},
    {t1, (extraext * 1/2 + 2 l + 3/2) 2 π/ω, (extraext * 1/2 + 2 l + 1) 2 π/ω,
      (extraext * 1/2 + 2 l + 2) 2 π/ω}]; ext2l1outerh2 = (ϵexternal) melnikov[ω, ext2l1outer20];
{ext2l1outer20, ext2l1outer21, ext2l1outerh2}

{ext2l1outer30, ext2l1outer31} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + 9999 π/(10000 ω), extraext * π/ω + π/(2 ω), extraext * π/ω + π/ω},
    {t1, (extraext * 1/2 + 2 l + 3/2) 2 π/ω, (extraext * 1/2 + 2 l + 1) 2 π/ω,
      (extraext * 1/2 + 2 l + 2) 2 π/ω}]; ext2l1outerh3 = (ϵexternal) melnikov[ω, ext2l1outer30];
{ext2l1outer30, ext2l1outer31, ext2l1outerh3}
```

```
{ext2l1outer40, ext2l1outer41} =
  {t0, t1} /. FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      melnikov[ω, t0] + melnikov[ω, t1] == 0},
    {t0, extraext * π/ω + 9999π/(10000ω), extraext * π/ω + π/(2ω), extraext * π/ω + π/ω},
    {t1, (extraext * 1/2 + 21 + 11/6) 2π/ω, (extraext * 1/2 + 21 + 1) 2π/ω,
      (extraext * 1/2 + 21 + 2) 2π/ω}]; ext2l1outerh4 = (ϵexternal) melnikov[ω, ext2l1outer40];
{ext2l1outer40, ext2l1outer41, ext2l1outerh4}

(*************************************************************************)
(* Find inner intersection points for external lobe 2l+1 *)
(*************************************************************************)

{ext2l1inner10, ext2l1inner11, ext2l1inner12} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
      t2 == t1 + periodexternal[(ϵexternal) melnikov[ω, t0] + (ϵexternal) melnikov[ω, t1]],
      melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
     {t0, 0.5 (extl1zero20 + extlzero10), extl1zero20, extlzero10},
     {t1, (extraext * 1/2 + 1 + 1) 2π/ω, (extraext * 1/2 + 1 + 1/2) 2π/ω,
      (extraext * 1/2 + 1 + 3/2) 2π/ω}, {t2, (extraext * 1/2 + 21 + 13/8) 2π/ω, (extraext * 1/2 + 21 + 3/2) 2π/ω,
      (extraext * 1/2 + 21 + 7/4) 2π/ω}]; ext2l1innerh11 = (ϵexternal) melnikov[ω, ext2l1inner10];
ext2l1innerh21 = ext2l1innerh11 + (ϵexternal) melnikov[ω, ext2l1inner11];
{ext2l1inner10, ext2l1inner11, ext2l1inner12, ext2l1innerh11, ext2l1innerh21}
```

```
{ext2l1inner20, ext2l1inner21, ext2l1inner22} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
       t2 == t1 + periodexternal[(ϵexternal) melnikov[ω, t0] + (ϵexternal) melnikov[ω, t1]],
       melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
      {t0, 0.5 (extl1zero20 + extlzero10), extl1zero20, extlzero10},
      {t2, {extraext * 1/2 + 21 + 15/8} 2π/ω, {extraext * 1/2 + 21 + 7/4} 2π/ω,
       {extraext * 1/2 + 21 + 2} 2π/ω}, {t1, {extraext * 1/2 + 1 + 1} 2π/ω,
       {extraext * 1/2 + 1 + 1/2} 2π/ω, {extraext * 1/2 + 1 + 3/2} 2π/ω}];
ext2l1innerh12 = (ϵexternal) melnikov[ω, ext2l1inner20];
ext2l1innerh22 = ext2l1innerh12 + (ϵexternal) melnikov[ω, ext2l1inner21];
{ext2l1inner20, ext2l1inner21, ext2l1inner22, ext2l1innerh12, ext2l1innerh22}

{ext2l1inner30, ext2l1inner31, ext2l1inner32} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
       t2 == t1 + periodexternal[(ϵexternal) melnikov[ω, t0] + (ϵexternal) melnikov[ω, t1]],
       melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
      {t0, 0.5 (extl1zero30 + extlzero20), extlzero20, extl1zero30},
      {t1, {extraext * 1/2 + 1 + 1} 2π/ω, {extraext * 1/2 + 1 + 1/2} 2π/ω,
       {extraext * 1/2 + 1 + 3/2} 2π/ω}, {t2, {extraext * 1/2 + 21 + 15/8} 2π/ω, {extraext * 1/2 + 21 + 7/4} 2π/ω,
       {extraext * 1/2 + 21 + 2} 2π/ω}]; ext2l1innerh13 = (ϵexternal) melnikov[ω, ext2l1inner30];
ext2l1innerh23 = ext2l1innerh13 + (ϵexternal) melnikov[ω, ext2l1inner31];
{ext2l1inner30, ext2l1inner31, ext2l1inner32, ext2l1innerh13, ext2l1innerh23}
```

```
{ext2l1inner40, ext2l1inner41, ext2l1inner42} =
  {t0, t1, t2} /.
    FindRoot[{t1 == t0 + periodexternal[(ϵexternal) melnikov[ω, t0]],
        t2 == t1 + periodexternal[(ϵexternal) melnikov[ω, t0] + (ϵexternal) melnikov[ω, t1]],
        melnikov[ω, t0] + melnikov[ω, t1] + melnikov[ω, t2] == 0},
      {t0, 0.5 (extl1zero30 + extlzero20), extlzero20, extl1zero30},
      {t1, (extraext * 1/2 + 1 + 1) 2π/ω, (extraext * 1/2 + 1 + 1/2) 2π/ω,
        (extraext * 1/2 + 1 + 3/2) 2π/ω}, {t2, (extraext * 1/2 + 21 + 13/8) 2π/ω, (extraext * 1/2 + 21 + 3/2) 2π/ω,
        (extraext * 1/2 + 21 + 7/4) 2π/ω}]; ext2l1innerh14 = (ϵexternal) melnikov[ω, ext2l1inner40];
ext2l1innerh24 = ext2l1innerh14 + (ϵexternal) melnikov[ω, ext2l1inner41];
{ext2l1inner40, ext2l1inner41, ext2l1inner42, ext2l1innerh14, ext2l1innerh24}

(*************************************************************************)
(* Calculation of areas of intersection of internal lobes l, l+1, l+2,
  and 2l+1 with external lobe 0.  Note that lobe 2l+1 has two intersection strips,
  inner and outer, the areas of which have been combined into one term. *)
(*************************************************************************)

Print ["Areas of intersections of internal lobes l, l+1, l+2, and 2l+1 with external lobe 0"]

  areaintl = Abs [NIntegrate[(-ϵinternal * (melnikov[ω, t] + melnikov[ω, t +
                periodinternal[(ϵinternal) melnikov[ω, t]]])), {t, intlzero10, intlzero20}]]

areaintl1 = Abs [ϵinternal * NIntegrate[(melnikov[ω, t] + melnikov[ω, t +
                periodinternal[(ϵinternal) melnikov[ω, t]]]), {t, intl1zero10, intl1zero20}] +
               (ϵinternal) NIntegrate[(melnikov[ω, t] + melnikov[ω, t +
                periodinternal[(ϵinternal) melnikov[ω, t]]]), {t, intl1zero30, intl1zero40}]]

      areaintl2 = Abs [(ϵinternal) NIntegrate[(melnikov[ω, t]
          + melnikov[ω, t + periodinternal[(ϵinternal) melnikov[ω, t]]]),
            {t, intl2zero10, intl2zero20}] + ϵinternal * NIntegrate[
          (melnikov[ω, t] + melnikov[ω, t + periodinternal[(ϵinternal) melnikov[ω, t]]]),
          {t, intl2zero30, intl2zero40}]]
```

```
  areaint2l1 = Abs [ϵinternal*NIntegrate[(melnikov[ω,t]
          + melnikov [ω, t + periodinternal [(ϵinternal) melnikov [ω, t]]]),
             {t, int2l1outer10, int2l1outer20}]  +  ϵinternal*NIntegrate[
             (melnikov [ω, t] + melnikov [ω, t + periodinternal [(ϵinternal) melnikov [ω, t]]]),
             {t, int2l1outer30, int2l1outer40}] + ϵinternal*NIntegrate[
             (melnikov [ω, t]  +  melnikov [ω, t +
             periodinternal [(ϵinternal) melnikov [ω, t]]] +
          melnikov [ω, t + periodinternal [(ϵinternal) melnikov [ω, t]] +
             periodinternal [(ϵinternal) melnikov [ω, t] +
                (ϵinternal) melnikov [ω, t +
                    periodinternal [(ϵinternal) melnikov [ω, t]]]]]),
                              {t, int2l1inner10, int2l1inner20}] +
                 ϵinternal*NIntegrate [(melnikov [ω, t] +
          melnikov [ω, t + periodinternal [(ϵinternal) melnikov [ω, t]]] +
          melnikov [ω, t + periodinternal [(ϵinternal) melnikov [ω, t]] +
             periodinternal [(ϵinternal) melnikov [ω, t] +
                (ϵinternal) melnikov [ω, t + periodinternal [(ϵinternal) melnikov [ω, t]]]]),
                        {t, int2l1inner30, int2l1inner40}]]


(*****************************************************************)
(* Calculation of areas of intersection of external lobes l, l+1, l+2, and 2l+1 with internal lobe 0.
Note that lobe 2l+1 has two intersection strips, inner and outer, the areas of which have been
combined into one term. *)
(*****************************************************************)


Print ["Areas of intersections of external lobes l, l+1, l+2, and 2l+1 with internal lobe 0"]

areaextl =  Abs [NIntegrate [(-ϵexternal* (melnikov [ω, t] +
          melnikov [ω, t + periodexternal [(ϵexternal) melnikov [ω, t]]])),
                {t, extlzero10, extlzero20}]]

areaextl1 = Abs [ϵexternal*NIntegrate [(melnikov [ω, t]  +
          melnikov [ω, t + periodexternal [(ϵexternal) melnikov [ω, t]]]),
              {t, extl1zero10, extl1zero20}]  + (ϵexternal) NIntegrate [
             (melnikov [ω, t]   + melnikov [ω, t +
             periodexternal [(ϵexternal) melnikov [ω, t]]]),
                    {t, extl1zero30, extl1zero40}]]

areaextl2  =  Abs [ϵexternal*NIntegrate [(melnikov[ω,t]  +
          melnikov[ω,t + periodexternal[(ϵexternal) melnikov[ω,t]]]) ,
             {t,extl2zero10,extl2zero20}] + (ϵexternal) NIntegrate [
            (melnikov[ω,t] + melnikov[ω,
           t +  periodexternal[(ϵexternal) melnikov[ω,t]]]),
              {t,extl2zero30,extl2zero40}]]
```

```
areaext2l1 = Abs[ϵexternal * NIntegrate[(melnikov[ω, t] +
                melnikov[ω, t + periodexternal[(ϵexternal) melnikov[ω, t]]]),
               {t, ext2l1outer10, ext2l1outer20}] +
              (ϵexternal) NIntegrate[(melnikov[ω, t] +
               melnikov[ω, t + periodexternal[(ϵexternal) melnikov[ω, t]]]),
               {t, ext2l1outer30, ext2l1outer40}] + ϵexternal *
         NIntegrate[(melnikov[ω, t] + melnikov[ω, t +
                  periodexternal[(ϵexternal) melnikov[ω, t]]] +
               melnikov[ω, t + periodexternal[(ϵexternal) melnikov[ω, t]] +
                  periodexternal[(ϵexternal) melnikov[ω, t] +
                     (ϵexternal) melnikov[ω, t + periodexternal[
                           (ϵexternal) melnikov[ω, t]]]]]),
           {t, ext2l1inner10, ext2l1inner20}] +
       ϵexternal * NIntegrate[(melnikov[ω, t] +
               melnikov[ω, t + periodexternal[(ϵexternal) melnikov[ω, t]]] +
               melnikov[ω, t + periodexternal[(ϵexternal) melnikov[ω, t]] +
                  periodexternal[(ϵexternal) melnikov[ω, t] +
                     (ϵexternal) melnikov[ω, t + periodexternal[
                           (ϵexternal) melnikov[ω, t]]]]]),
           {t, ext2l1inner30, ext2l1inner40}]]


Print["Areas of one lobe and the region between the unperturbed manifolds"]


arealobe = (ϵexternal) Abs[NIntegrate[((melnikov[ω, t])), {t, 0, π/ω}]]


areaS = 2 * NIntegrate[ArcCosh[(1 + A) Exp[0] − A Cos[x]], {x, 0, 2π}]


(*************************************************************************)
(* Calculation of the weighting factors s1, s2,
  and s3 for internal and external lobes.  Note that if an asymmetry  is introduced,
  the s3 factor for the internal lobes must be calculated separately for the top  and bottom halves,
  since this factor includes factors from both the top and the bottom. *)
(*************************************************************************)
```

```
Print ["Weighting factors s1, s2, and s3 for internal lobes"]

s1int = areaintl2 / areaintl1
s2int = 1 - areaintl1 / (arealobe - areaintl) - s1int
s3int = 1 / 2 + ((s1int)^1 areaintl1 - areaint2l1) / (2 * s2int * (arealobe - areaintl))

Print ["Weighting factors s1, s2, and s3 for external lobes"]

s1ext = areaextl2 / areaextl1
s2ext = 1 - areaextl1 / (arealobe - areaextl) - s1ext
s3ext = 1 / 2 + ((s1ext)^1 areaextl1 - areaext2l1) / (2 * s2ext * (arealobe - areaextl))

(*************************************************************************)
(* Calculate the lobe dynamics transfer matrix and the initial vector. *)
(*************************************************************************)

lmatrix = Table[0, {4 (l + 2)}, {4 (l + 2)}];
Do [lmatrix [[6 + j + i (l + 1), 7 + j + i (l + 1)]] = 1, {i, 0, 3}, {j, 0, l - 2}];
lmatrix [[5, 5]] = s1ext; lmatrix [[5 + 3 (l + 1), 5 + 3 (l + 1)]] = s1ext;
lmatrix [[5, 6]] = 2 s3ext; lmatrix [[5 + 3 (l + 1), 6 + 3 (l + 1)]] = 2 s3ext;
lmatrix [[4 + (l + 1), 5]] = s2ext; lmatrix [[4 (l + 2), 5 + 3 (l + 1)]] = s2ext;
lmatrix [[2, 5]] = 1 - s1ext - s2ext; lmatrix [[2, 6]] = 1 - 2 s3ext;
lmatrix [[3, 5 + 3 (l + 1)]] = 1 - s1ext - s2ext; lmatrix [[3, 6 + 3 (l + 1)]] = 1 - 2 s3ext;
lmatrix [[5 + (l + 1), 5 + (l + 1)]] = s1int; lmatrix [[5 + 2 (l + 1), 5 + 2 (l + 1)]] = s1int;
lmatrix [[5 + (l + 1), 6 + (l + 1)]] = 2 s3int; lmatrix [[5 + 2 (l + 1), 6 + 2 (l + 1)]] = 2 s3int;
lmatrix [[4 + 3 (l + 1), 5 + (l + 1)]] = s2int; lmatrix[ [4 + 2 (l + 1), 5 + 2 (l + 1)]] = s2int;
lmatrix [[1, 5 + 2 (l + 1)]] = 1 - s1int - s2int; lmatrix [[1, 6 + 2 (l + 1)]] = 1 - 2 s3int;
lmatrix [[4, 5 + (l + 1)]] = 1 - s1int - s2int; lmatrix [[4, 6 + (l + 1)]] = 1 - 2 s3int;

initialvector = Table[0, {4 (l + 2)}];
initialvector [[1]] = areaintl; initialvector [[4]] = areaintl;
initialvector [[2]] = areaextl; initialvector [[3]] = areaextl;
initialvector [[5]] = arealobe - areaextl; initialvector [[5 + (l + 1)]] = arealobe - areaintl;
initialvector [[5 + 2 (l + 1)]] = arealobe - areaintl;
initialvector [[5 + 3 (l + 1)]] = arealobe - areaextl;

(*************************************************************************)
(* Use the l matrix and the initial vector to calculate the escape rate at iteration n. *)
(*************************************************************************)
```

```
result [power_]:= MatrixPower [lmatrix, power] . initialvector;

pointsint = Table[{n,(If[n<l, tempescapeint = arealobe, tempescapeint = arealobe -
Sum[result[j],{j,0,n-l}] [[1]]])},{n,1,nend}];

pointsext = Table[{n,(If[n<l, tempescapeext = arealobe, tempescapeext = arealobe -
Sum[result[j],{j,0,n-l}] [[2]]])},{n,1,nend}];

internalplot = ListPlot [pointsint, PlotStyle→{RGBColor [1,0,0]}, AxesOrigin→{0,0},
PlotRange→{0,arealobe}];

externalplot = ListPlot [pointsext, PlotStyle→{RGBColor [0,0,1]}, AxesOrigin→{0,0},
PlotRange→{0,arealobe}];

Show[internalplot, externalplot, TextStyle→{FontSize→16, FontFamily→"Times New Roman"},
AxesLabel→{"n", "Area"}, PlotLabel→"Area Transferred per Iteration -
   StyleBox[\"int\",\nFontColor->RGBColor[1, 0, 0]]
   StyleBox[\" \",\nFontColor->GrayLevel[0]]
   StyleBox[\"ext\",\nFontColor->RGBColor[0, 0, 1]] "]

pointstotalint = Table[{n, If [n<l, totalescapeint = n*arealobe, (totalescapeint = n*arealobe -
Sum[(n-j)*result[j],{j,0,n-l}] [[1]])]},{n,1,nend}];
totalescapeint

internaltotalplot = ListPlot [pointstotalint, PlotStyle→{RGBColor [1,0,0]}, AxesOrigin→{0,0},
PlotRange→{0,1.1* totalescapeint}]

pointstotalext = Table[{n, If [n<l, totalescapeext = n * arealobe, (totalescapeext = n * arealobe -
Sum[(n-j)*result[j],{j,0,n-l}] [[2]])]},{n,1,nend}];
totalescapeext

externaltotalplot = ListPlot [pointstotalext, PlotStyle→{RGBColor [0,0,1]}, AxesOrigin→{0,0},
PlotRange→{0,1.1 * totalescapeext}]

Show[internaltotalplot, externaltotalplot, TextStyle→{FontSize→16, FontFamily→"Times New
Roman"}, AxesLabel→{"n", "Area"}, PlotLabel→"Cumulative Area Transferred -
   StyleBox[\"int\",\nFontColor->RGBColor[1, 0, 0]]
   StyleBox[\" \",\nFontColor->GrayLevel[0]]
   StyleBox[\"ext\",\nFontColor->RGBColor[0, 0, 1]] "]

totalescapeint2=nend*arealobe-Sum[(nend-j)*result[j],{j,0,nend-l}] [[1]];
totalescapeext2=nend*arealobe-Sum[(nend-j)*result[j],{j,0,nend-l}] [[2]]

ωplot [l]=ω
escapeint [l]=totalescapeint2
escapeext [l]=totalescapeext2
```

Vita

Stephen Michael Rodrigue was born in New Orleans on September 16, 1959. He attended Brother Martin High School and graduated valedictorian in May, 1977. He then attended the University of New Orleans, earning two B.S. degrees, in Physics and Mathematics, and graduated magna cum laude in May, 1982. Leaving academia for a period after briefly being a graduate student at the University of New Orleans and the University of Washington in Seattle, he taught high school at McDonogh 35 Senior High School and De La Salle High School in New Orleans, did research with the Naval Oceanographic Office, and worked with Richard Camus in actuarial consulting. After traveling around the country, he returned to New Orleans and worked as a contract tutor with Basics Plus for five years, in addition to entering the Graduate School at the University of New Orleans. Finishing with M.S. degrees in both Mathematics and Physics, upon graduation in August, 1994 he left for New York to work in the Physics Ph.D. program at State University of New York at Stony Brook. He briefly worked with the New York Public Interest Research Group, and began working at Kaplan while in New York. Upon returning to New Orleans in 1998, he entered the Engineering and Applied Science Ph.D. program at the University of New Orleans. He taught as an instructor in the Physics and Dual Degree Engineering Program at Xavier University of Louisiana for five years, 1998-2003. He will earn his Ph.D. in Engineering and Applied Science in August, 2006.