

Asymptotic Statistics of Nodal Domains of
Quantum Chaotic Billiards in the Semiclassical
Limit

Kyle Konrad
Dartmouth College
Computer Science Department
Advisor: Alex Barnett

May 31, 2012

Abstract

Quantum chaos concerns eigenfunctions of the Laplace operator in a domain where a billiard ball would bounce chaotically. Such chaotic eigenfunctions have been conjectured to share statistical properties of their nodal domains with a simple percolation model, from which many interesting quantities can be computed analytically. We numerically test conjectures on the number and size of nodal domains of quantum chaotic eigenfunctions at very high energies, approaching the semiclassical limit. We use a highly efficient scaling method to quickly compute eigenfunctions at low resolution and interpolate to higher resolution. We computed 10^5 eigenfunctions and counted 10^9 nodal domains. Our results agree with the conjectured size nodal domains but disagree with the conjectured mean and variance of the number of nodal domains.

Chapter 1

Introduction

1.1 Motivation

Nodal domains characterize regions of a vibrational surface (e.g. a drum head) that move in phase. The boundaries between nodal domains, known as nodal lines, are the regions which do not vibrate at all (fig. 1.1). Understanding the characteristics of nodal domains has applications in spectral geometry, mathematical physics, mechanical engineering, geophysics, astrophysics, and many other area dealing with wave behavior [13].

We study nodal domains of eigenfunctions of the Laplace operator on non-integrable Euclidean domains, or chaotic billiards. These eigenfunctions are known as quantum chaotic eigenfunctions because they are solutions of the quantum mechanical wave equation, the Schrödinger equation. Quantum chaotic eigenfunctions are a canonical example of quantum chaos, which lies at the intersection of quantum mechanics and chaos theory. The primary signature of chaos in classical systems is a nonlinear (exponential) divergence of trajectories in the phase space of a system. Quantum mechanics however, is entirely linear and quantum chaos deals with energy eigenfunctions of systems, which are constant in time. Thus chaos in quantum systems is manifest in different ways, one of the most studied being wavefunctions in chaotic domains. Figure 1.2 shows a classical orbit and a quantum eigenfunction on the same billiard.

The goal of this project is to numerically test conjectures on the mean and variance of the number and sizes of nodal domains in quantum chaotic eigenfunctions in the high energy, or semiclassical, limit. These tests are motivated by questions from number theorist Peter Sarnak, who inspired us to test the conjecture on the mean number of nodal domains. Obtaining relevant data requires solving new computational challenges due to the computational intensity of evaluating eigenfunctions at very high energies. We hope these results will enable further mathematical investigation of eigenfunctions and that the tools developed herein may be applied to related problems in quantum chaos.

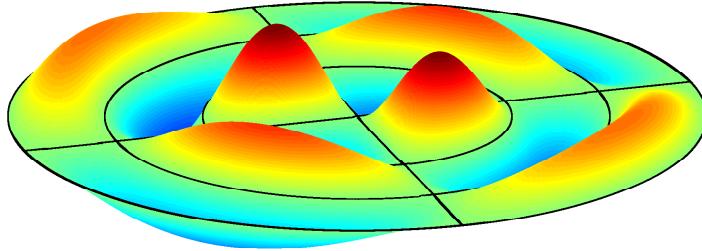


Figure 1.1: A circular vibrational surface with nodal lines shown in black. Nodal domains are regions between black lines

1.2 Classical Chaos in Billiards

A billiard is a compact domain $\Omega \subset \mathbb{R}^2$. By parameterizing the perimeter of a billiard by $s \in [0, L]$, where L is the perimeter length, we can construct a map $P : \mathbb{R} (\text{mod } L) \times \mathbb{R} (\text{mod } \pi) \rightarrow \mathbb{R} (\text{mod } L) \times \mathbb{R} (\text{mod } \pi)$ such that a trajectory incident on the boundary at parameter s and angle θ from the tangent to the boundary at s will next intersect the boundary at parameter and angle of incidence $(s', \theta') = P(s, \theta)$. This map describes the motion of a ball bouncing in the domain Ω .

Chaos in classical systems is characterized by the Lyapunov exponent λ of a system, which describes how quickly nearby trajectories diverge. It is computed as the long time ratio of the divergence of two initially close paths:

$$\lambda = \lim_{n \rightarrow \infty} \lim_{|\epsilon| \rightarrow 0} \frac{1}{n} \ln \frac{|P^n(\mathbf{x}_0) - P^n(\mathbf{x}_0 + \epsilon)|}{|\epsilon|}$$

Where $\mathbf{x}_0 = (s_0, \theta_0)$ and $|(s, \theta)| = s$. From this definition it follows that

$$|P^n(\mathbf{x}_0) - P^n(\mathbf{x}_0 + \epsilon)| \approx e^{\lambda n} |\epsilon|$$

for small ϵ and large n . Thus a tiny change ϵ in initial conditions produces a change that grows exponentially over time with growth rate λ . Chaotic systems have a positive Lyapunov exponent and therefore have unpredictable long-term behavior because arbitrarily small errors in measurements of initial conditions eventually become large. As these errors grow to the size of the domain, the position of a particle approaches a uniform distribution over the entire billiard. The property that small subsets of Ω eventually map to all of Ω is known as ergodicity (see appendix A for a formal definition).

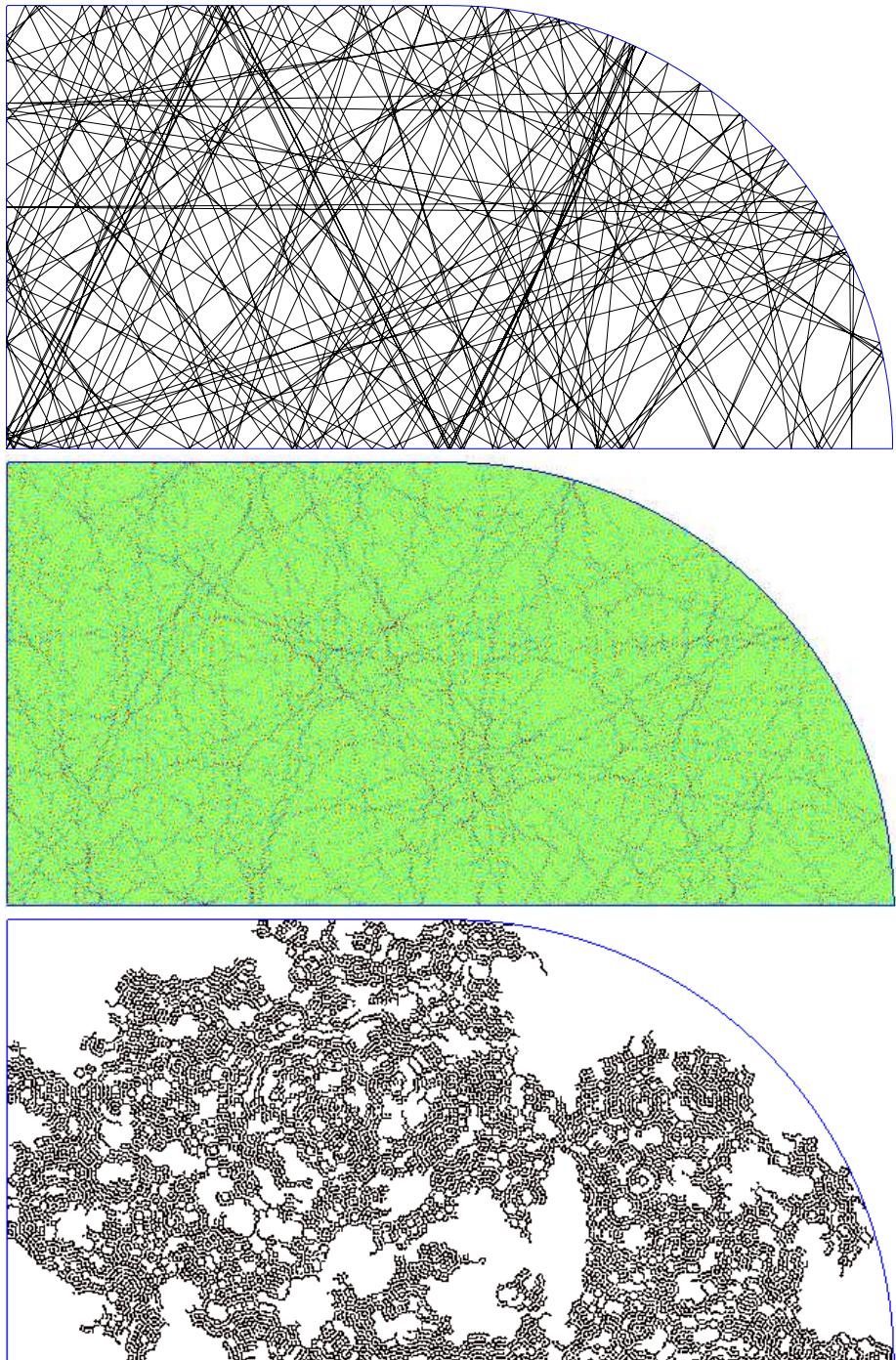


Figure 1.2: Above: A classical orbit in a billiard; center: a quantum eigenfunction in the same billiard; below: the largest nodal domain in the center eigenfunction.

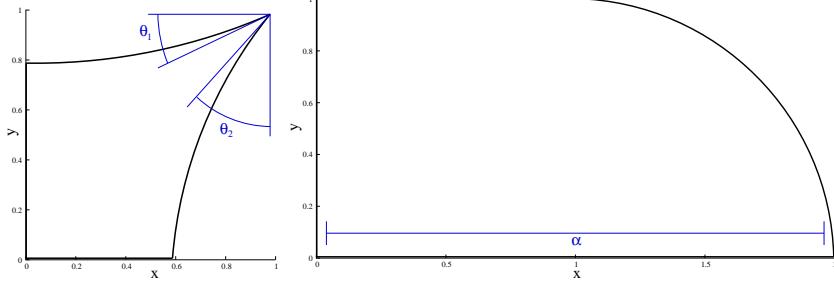


Figure 1.3: Left: quarter generalized rectangular Sinai billiard; right: quarter stadium billiard

1.3 Quantum Chaos in Billiards

A quantum wave-particle in a billiard Ω obeys the Schrödinger equation

$$Eu(\mathbf{r}) = -\frac{\hbar^2}{2m} \Delta u(\mathbf{r}) + V(\mathbf{r})u(\mathbf{r})$$

Where $\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian differential operator in two dimensions. Setting $\hbar = 2m = 1$ and $V(\mathbf{r}) = 0$ for $\mathbf{r} \in \Omega$ while enforcing Dirichlet boundary conditions $u(\mathbf{r}) = 0$ for $\mathbf{r} \in \Gamma = \partial\Omega$ simplifies this to the Helmholtz equation

$$\begin{cases} (\Delta + k^2)u(\mathbf{r}) = 0 & \text{if } \mathbf{r} \in \Omega \\ u(\mathbf{r}) = 0 & \text{if } \mathbf{r} \in \Gamma \end{cases} \quad (1.1)$$

where $k^2 = E$ is the corresponding eigenvalue for eigenfunction $u(\mathbf{r})$ and E is kinetic energy of the quantum wave-particle.

We focus our investigation on two billiard shapes: the generalized rectangular Sinai billiard and the Stadium billiard (fig. 1.3). In both cases we desymmetrize the billiard shapes by considering only a quarter of the full shape. This restricts our basis set to functions that are odd as functions of x and y , i.e., $f(-x, y) = f(x, -y) = -f(x, y)$.

The Sinai billiard is constructed from circular arcs that meet at $(1, 1)$ and is parameterized by two angles, θ_1 and θ_2 , the angles from horizontal and vertical, respectively, of the arcs at $(1, 1)$. The Sinai billiard is said to demonstrate “hard chaos” because there are no stable or neutrally stable orbits. We use values of $\theta_1 = 0.4$ and $\theta_2 = 0.7$.

The stadium billiard is constructed from a rectangular region and a quarter circle and is parameterized by the the horizontal length of the billiard α . We use a value of $\alpha = 2$ here. The stadium billiard contains neutrally stable orbits, specifically those with vertical momentum in the rectangular region. Neutrally stable orbits have zero Lyapunov exponent but any perturbation will cause them to have positive Lyapunov exponent. The existence of such orbits implies

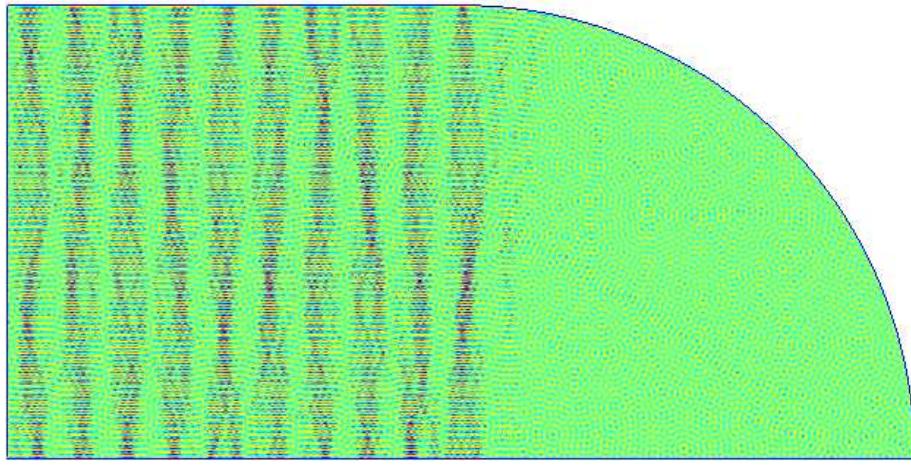


Figure 1.4: A bouncing ball mode in the stadium billiard with $k = 500.3881$

that the stadium does not demonstrate hard chaos but because these orbits occupy a measure zero subset of phase space, the stadium still has chaotic properties. These classical orbits are manifest in quantum eigenfunctions as so-called “bouncing-ball” modes (fig. 1.4).

1.4 Percolation Model

Bogomolny and Schmit [5] have argued that nodal domains of random functions (which are considered an accurate proxy for eigenfunctions of chaotic systems [11]) can be modeled by nodal domains of a percolation model. Their percolation model is formed by creating a checkerboard of positive and negative regions with a grid size given by the average spacing of zeros of random functions along a particular axis. This checkerboard pattern can be realized as an eigenfunction $\bar{u}(x, y) = \sin(\frac{kx}{\sqrt{2}})\sin(\frac{ky}{\sqrt{2}})$ of a square billiard $\Omega = [0, 1]^2$. A random eigenfunction can be modelled as this mean eigenfunction $\bar{u}(x, y)$ plus another term representing deviation from the mean $u(x, y) = \bar{u}(x, y) + \delta u(x, y)$. The deviation term $\delta u(x, y)$ is modelled by perturbing each nodal line crossing by connecting two diagonal regions (fig. 1.5). The decision of which nodal domains to connect is made randomly with equal probability for either possibility.

Bogomolny and Schmit apply results from graph theory and statistical physics to compute the distribution of nodal domains in this percolation model. These results comprise the conjectures we seek to test numerically.

Conjecture 1 (Mean of Nodal Domain Count). *The number of nodal domains $\nu(E)$ in a quantum chaotic eigenfunction with energy E is normally distributed*

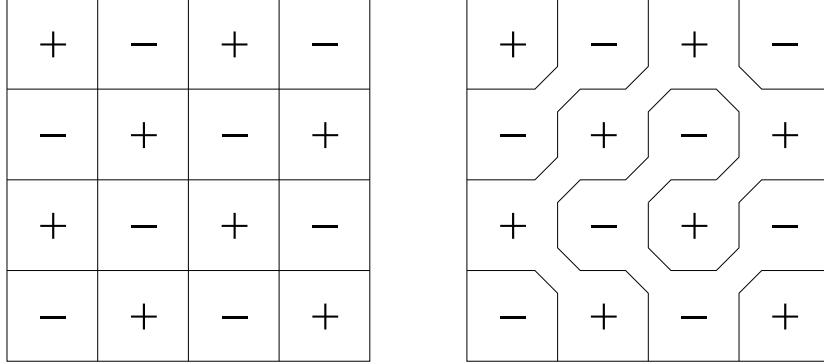


Figure 1.5: Left: checkerboard pattern; right: perturbed checkerboard pattern

with mean

$$\frac{\bar{\nu}(E)}{\bar{N}(E)} = \frac{3\sqrt{3} - 5}{\pi} \approx 0.0624 \quad (1.2)$$

Conjecture 2 (Variance of Nodal Domain Count). *The number of nodal domains $\nu(E)$ in a quantum chaotic eigenfunction with energy E is normally distributed with variance*

$$\frac{\sigma^2(\nu(E))}{\bar{N}(E)} = \frac{18}{\pi^2} + \frac{4\sqrt{3}}{\pi} - \frac{25}{2\pi} \approx 0.0502 \quad (1.3)$$

In both conjectures, $\bar{N}(E)$ is the mean number of eigenvalues less than E which has asymptotic behavior given by Weyl's law [7]

$$\bar{N}(E) \sim \frac{|\Omega|E}{4\pi} \quad (1.4)$$

Bogolmony and Schmit also obtain a prediction for the distribution of areas of nodal domains.

Conjecture 3 (Area of Nodal Domains). *The area s of nodal domains in a quantum chaotic eigenfunction follows the distribution*

$$f(s) \propto s^{-\tau} \quad (1.5)$$

where $\tau = \frac{187}{91}$ is the Fisher exponent.

1.4.1 Implementation

The percolation model was implemented in code by creating a checkerboard pattern with each square being two pixels by two pixel that was then perturbed at each nodal line crossing by changing the sign of a pixel to either connect the

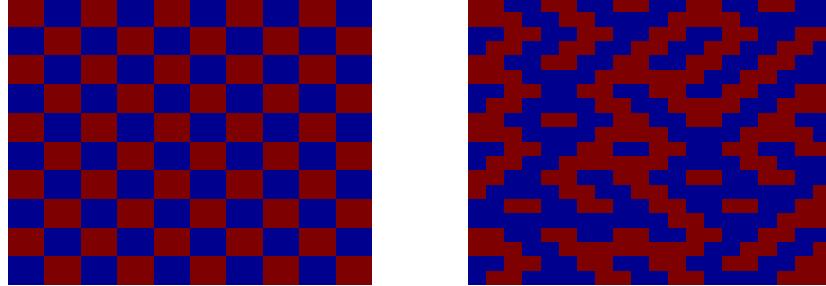


Figure 1.6: Left: checkerboard pattern implementation; right: perturbed checkerboard pattern implementation

top right square to the bottom left square or the top left square to the bottom right square (fig. 1.6). The number of squares m in one direction on a unit grid is determined by k to be [5].

$$m = \frac{k}{\sqrt{2}\pi}$$

Making each square two pixels by two pixels uses a grid of size $(2m)^2 = \frac{2k^2}{\pi^2}$. This construction allows us to numerically validate the conjectures of the percolation model via the nodal domain counting algorithm described in 2.4.

1.5 Random Plane Waves

A random superposition of plane waves,

$$u_{rand}(\mathbf{r}; k) = \Re \left[\lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \sum_{n=1}^N \omega_n \exp \{ik\hat{n}_n \cdot \mathbf{r}\} \right] \quad (1.6)$$

where $\omega_n \sim \mathcal{N}(0, 1)$ are complex and independent and identically distributed and $\hat{n}_n = (\cos \frac{2\pi n}{N}, \sin \frac{2\pi n}{N})$ are evenly spaced vectors around the unit circle, is considered an accurate model for a quantum chaotic eigenfunctions [11]. This random superposition of plane waves (hereafter “random plane wave”) is built from plane waves with constant wavenumber k with random phase and amplitude, oriented in all directions.

Numerically, random plane waves can be computed quickly using a nonuniform fast Fourier transform on vectors of length k with random phase. When testing new methods or obtaining general statistics, we often use random plane waves in place of actual eigenfunctions to save computation time.

Chapter 2

Methods

2.1 Computation of Eigenfunctions via Scaling Method

2.1.1 Theory

Vergini and Saraceno [12] developed a method of computing high energy eigenfunctions of chaotic billiards using a scaling method. The method simultaneously finds all eigenfunctions $u_i(\mathbf{r})$ with wavenumber in a given window $[k - \Delta k, k + \Delta k]$ by scaling each eigenfunction. The scaled eigenfunctions $\chi_i(k, \mathbf{r})$ are computed as

$$\chi_i(k, \mathbf{r}) = u_i\left(\frac{k}{k_i} \mathbf{r}\right)$$

This scaling causes all eigenfunctions to fall approximately in a linear subspace of a single basis set $\{\xi_l\}_{l=1}^B$,

$$\chi_i(k, \mathbf{r}) = \sum_{l=1}^B h_l^{(i)} \xi_l(k, \mathbf{r}) + \epsilon_i(\mathbf{r})$$

where $\epsilon_i(\mathbf{r}) \ll 1$ for sufficiently large B , the number of basis functions used. Values of B of approximately $1.5 \frac{k|\Gamma|}{\pi}$ have been shown to produce $\epsilon < 10^{-4}$ [4]. This single basis set provides a significant efficiency gain over prior methods because many eigenfunctions can be found by solving a single linear system and evaluating a single basis set on the domain.

The choice of basis functions $\xi_l(k, \mathbf{r})$ depend on the billiard shape being used. For the quarter generalized rectangular Sinai billiard the basis set consists of fundamental solutions, or irregular Bessel functions, (which satisfy $(\Delta + k^2)\xi_l(k, \mathbf{r}) = 0$) placed along Γ^+ , the set of points outside Ω whose nearest distance to Γ is D where kD is taken to be 7 so that D is approximately one wavelength. For the quarter stadium, the basis set is plane waves with orientations evenly spaced around the circle.

The numerical implementation of the scaling method is based on solving a generalized eigenvalue problem from which one can reconstruct eigenvalues and eigenfunctions of the original Dirichlet boundary value problem. The basic approach is to construct B by B matrices \mathbf{F} and \mathbf{G} where

$$\mathbf{F}_{ij} = \langle \xi_i, \xi_j \rangle$$

and

$$\mathbf{G}_{ij} = (\langle \xi_i, x \cdot \nabla \xi_j \rangle + \langle x \cdot \nabla \xi_i, \xi_j \rangle) / k$$

By solving the generalized eigenvalue problem

$$\mathbf{F}h = \mu \mathbf{G}h$$

one obtains approximations of the eigenvalues of the original problem with

$$\hat{k}_i = k - 2/\mu_i$$

and approximate eigenfunctions by “undilating” the generalized eigenvectors

$$\hat{u}_i(\mathbf{r}) = \sum_{l=1}^B h_l \xi_l(\hat{k}, \mathbf{r}) \quad (2.1)$$

The scaling method runs in $O(B^3) = O(k^3)$ time [3] and approximates eigenvalues and eigenfunctions with error $O(\Delta k^3)$ [4, p. 32].

2.1.2 Eigenfunction evaluation

The scaling method only computes coefficients of basis functions, which must be evaluated in order to obtain eigenfunction values at arbitrary $\mathbf{r} \in \Omega$. One method is to evaluate $\hat{u}(\mathbf{r})$ as in (2.1), but this requires evaluating the basis functions $\xi_l(\hat{k}, \mathbf{r})$ for each approximate eigenvalue \hat{k} . For efficiency, we instead evaluate only $\xi_l(k, \mathbf{r})$, which produces a dilated eigenfunction that can be transformed back to \hat{u} by dilating the coordinate system. Thus we can evaluate all eigenfunctions in the energy window with NB basis function evaluations and nNB coefficient multiplications, where n is the number of eigenfunctions in the energy window, N is the number of evaluation points, and B is the number of basis functions. Basis evaluation are much more expensive than multiplications; table 2.1 summarizes the ratios of these costs.

As noted above, errors in eigenfunctions produced by the scaling method scale like $O(\Delta k^3)$ and are independent of E so we use a constant Δk . We estimate errors by calculating the tension

$$t = \left(\int_{\Gamma} u(\mathbf{r})^2 d\mathbf{r} \right)^{\frac{1}{2}}$$

ΔE is chosen such that $t \lesssim 10^{-4}$. Using Weyl’s law 1.4 we can obtain an estimate of the number of eigenfunctions in an energy window to be

$$n \approx \frac{|\Omega|}{4\pi} ((E + \Delta E) - (E - \Delta E)) = \frac{|\Omega|}{2\pi} \Delta E$$

Billiard	Ratio	k_c
Sinai	75	3860
Stadium	31	550

Table 2.1: Ratios of basis function evaluation time to coefficient multiplication time for various billiards. Column k_c contains the value of k for which n is equal to the given ratio with $\Delta k = 0.1$ and coefficient multiplications take as long as evaluating basis functions

We choose Δk to be as large as possible while keeping errors small in order to maximize n , allowing more eigenfunctions to be computed from each evaluation of basis functions. We use $\Delta k = 0.1$ for the quarter generalized rectangular Sinai billiard and $\Delta k = 0.05$ for the quarter stadium billiard.

For both billiards considered here $B \sim k$. Thus, in practice $N \gg n$ and $N \gg B$ so it is primarily N (which scales like α^{-2}) that determines the time to compute eigenfunctions. Maximum values of N and B were $N = |\Omega|(k_{max}/\alpha)^2 = 5.25e6$ and $B = 1.5k_{max}|\Gamma|/\pi = 2333$.

2.2 Interpolation

Because sampling eigenfunctions is expensive (requiring $O(k^3)$ basis function evaluations), we are limited by the total number of pixels N , which scales like h^{-2} where h is the distance between adjacent sample points, or the width (and height) of a pixel. As a consequence, we must work with relatively coarsely sampled eigenfunctions, causing us to encounter scenarios where the connectivity of nodal domains may be ambiguous (fig. 2.1).

We prevent miscounting by interpolating the computed eigenfunction using the functions

$$J_n(kr) \sin(n\theta)$$

and

$$J_n(kr) \cos(n\theta)$$

where J_n is a regular Bessel function. These functions form a complete basis for solutions of (1.1) (see appendix B). We fix a value M to be the order of the highest order Bessel function, restricting our basis to the finite set

$$\zeta_j(r, \theta) = \begin{cases} J_j(kr) & \text{if } j = 0 \\ J_j(kr) \sin(j\theta) & \text{if } 1 \leq j \leq M \\ J_{j-M}(kr) \cos((j - M)\theta) & \text{if } M + 1 \leq j \leq 2M \end{cases} \quad (2.2)$$

We construct a surrogate function

$$\tilde{u}(\mathbf{r}) = \sum_{i=0}^{2M} c_i \zeta_i(\mathbf{r})$$

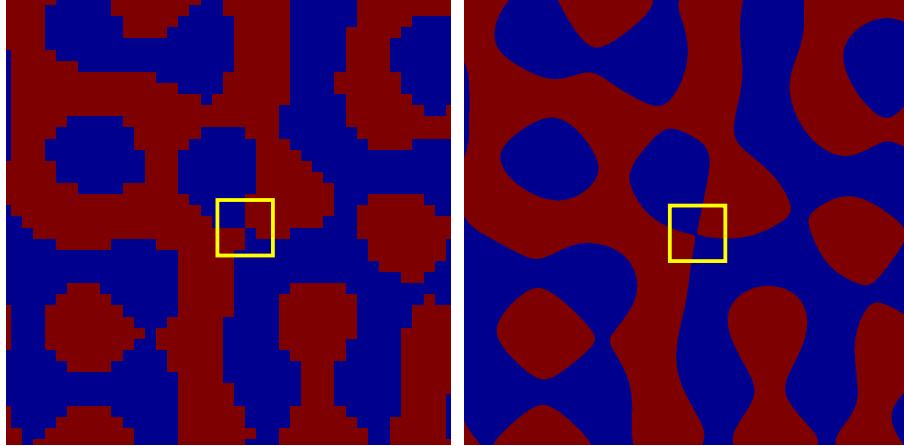


Figure 2.1: An ambiguous nodal domain connection due to coarse sampling. Left: A random plane wave sampled at 10 points per wavelength; right: the same random plane wave sampled at 200 points per wavelength. The ambiguous region at low resolution is highlighted

as a local approximation to the eigenfunction $u(\mathbf{r})$ by fitting the coefficients $c_i \in \mathbb{R}$ to minimize the error $\|u - \tilde{u}\|_2$. This function can then be sampled at a higher resolution within the region in question. We define the sampling ratio of this surrogate function to the original eigenfunction to be ρ .

2.2.1 Triggering Interpolation

A region is interpolated if and only if, when counting nodal domains, we encounter a point whose sign matches that of a point diagonally adjacent to it, but differs from the signs of the two points adjacent to both it and its diagonal neighbor (fig. 2.2). In such a case we fill a vector \mathbf{v} with the eigenfunction values at the stencil points surrounding the four points comprising the ambiguity. We then compute $\mathbf{w} = P\mathbf{v}$ where P is the interpolation matrix described above. This vector \mathbf{w} contains estimated eigenfunction values with a spacing of $\frac{h}{\rho}$ between the four points comprising the ambiguous region. We can use these values to determine the connectivity of the nodal domains by traversing pixel-by-pixel from the top-left pixel, in the same manner as above, until we either reach the bottom-right pixel or finish exploring the nodal domain. In the former case the nodal domain containing the top-left pixel connects to the nodal domain containing the bottom-right pixel and in the latter case the nodal domain containing the top-right pixel connects to the nodal domain containing the bottom-left pixel.

To check the validity of only interpolating at these configurations, we counted nodal domains of 10^4 eigenfunctions that were interpolated everywhere to $\alpha =$

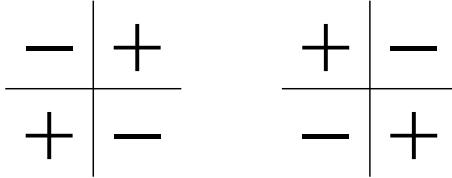


Figure 2.2: Configurations that will result in interpolation

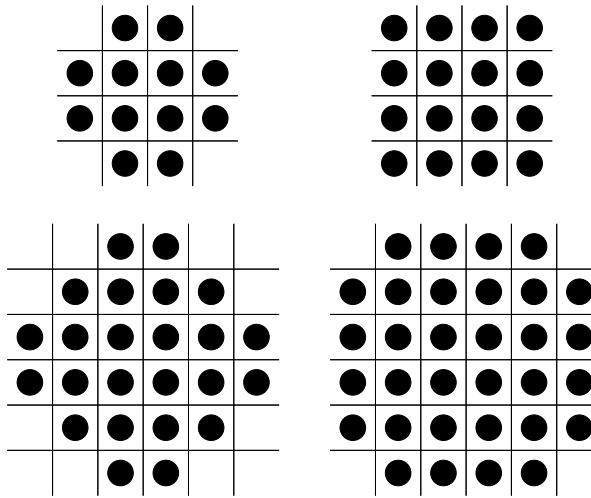


Figure 2.3: The four stencil shapes considered for interpolation

.023 and found that mean nodal domains counts agreed within 1%.

2.2.2 Parameter Selection

Each two pixel by two pixel square defines a region in which interpolation produces upsampled eigenfunction values. The surrogate function however, uses more than just four pixels to fit the coefficients c_i . The selection of how many and which surrounding values to use is termed a stencil. Only stencil shapes that are symmetric about the four central points were considered. The four shapes considered are shown in figure 2.3. The most important consideration in choosing a stencil was the accuracy of the interpolation. The size of the stencil affects the computational cost of interpolation but this difference is trivial.

The accuracy of interpolation also depends on M and $\alpha = kh$, which must be considered simultaneously with the choice of stencil. Figure 2.4 shows a comparison of the infinity norm of the interpolation error over various values of M and α for each of the stencils shown above.

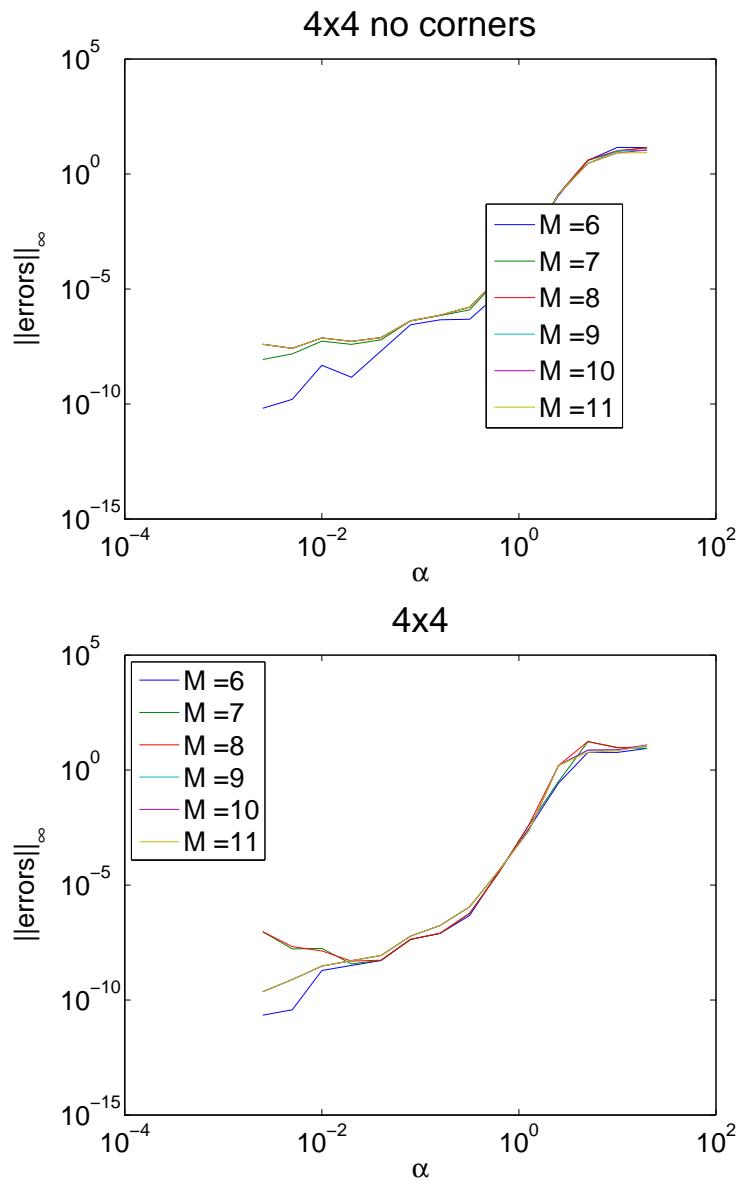
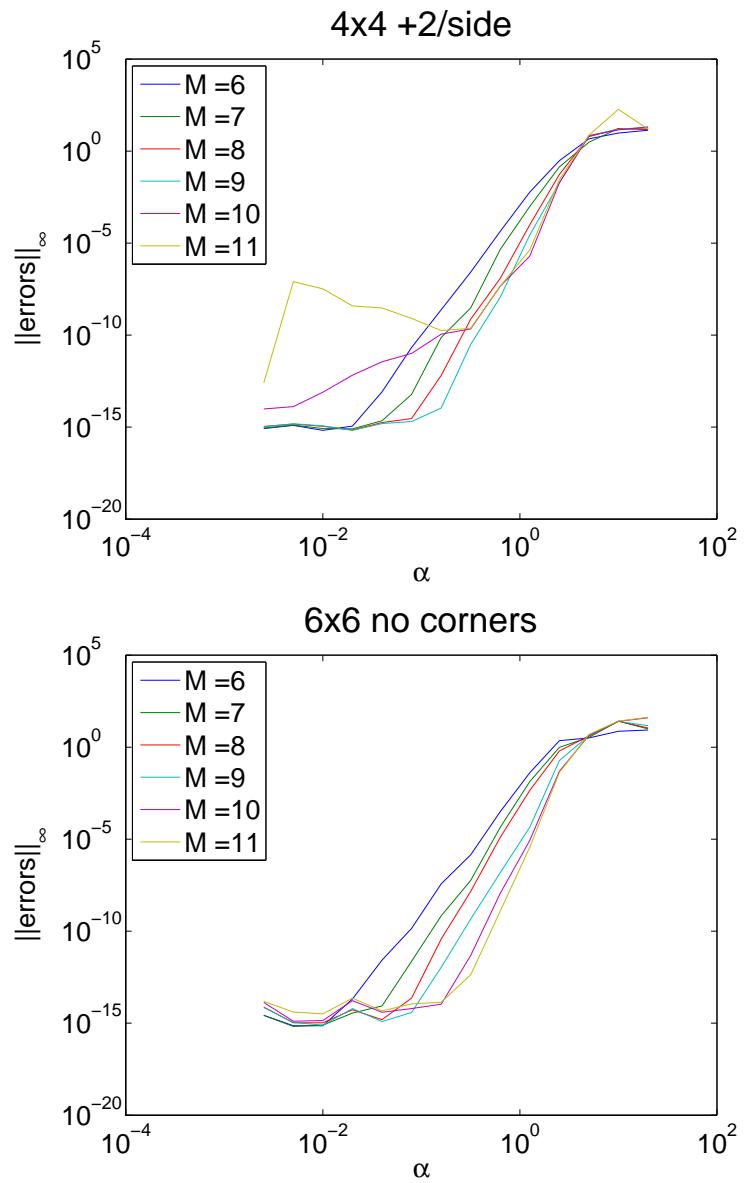


Figure 2.4: Comparison of interpolation error norms over M and α for each stencil.



Based on these data, we chose to use the 24-point stencil with $M = 9$ and $\alpha \in [0.5, 0.7]$ for interpolation. For this choice of parameters we expect errors in interpolated eigenfunction values of order 10^{-6} . We use an upsampling ratio of $\rho = 30$, which we have found to be sufficient to resolve ambiguities in nearly all cases.

2.2.3 Numerical Implementation

We construct a matrix to transform values on a stencil to interpolated values in the center of the stencil, performing the process described above with a single matrix multiplication. We construct the interpolation matrix as

$$P = HL^+$$

Where L is an S by $2M + 1$ matrix (where S is the number of stencil points) which contains evaluations of Bessel functions at stencil points, H is a $(\rho + 1)^2$ by $2M + 1$ contains evaluations of Bessel functions at high resolution and $+$ denotes the pseudoinverse. Specifically,

$$L_{ij} = \begin{cases} J_j(r_i) & \text{for } j = 0 \text{ and } 0 \leq i < T \\ J_j(r_i) \sin(j\theta_i) & \text{for } 1 \leq j \leq M \text{ and } 0 \leq i < T \\ J_{j-M}(r_i) \cos((j-M)\theta_i) & \text{for } M + 1 \leq j \leq 2M + 1 \text{ and } 0 \leq i < T \end{cases}$$

and

$$H_{ij} = \begin{cases} J_j(\tilde{r}_i) & \text{for } j = 0 \text{ and } 0 \leq i < \gamma \\ J_j(\tilde{r}_i) \sin(j\tilde{\theta}_i) & \text{for } 1 \leq j \leq M \text{ and } 0 \leq i < \gamma \\ J_{j-M}(\tilde{r}_i) \cos((j-M)\tilde{\theta}_i) & \text{for } M + 1 \leq j \leq 2M + 1 \text{ and } 0 \leq i < \gamma \end{cases}$$

where T is the number of points in the stencil, $\gamma = (\rho + 1)^2$ and (r_i, θ_i) and $(\tilde{r}_i, \tilde{\theta}_i)$ are points in the stencil and inner square, respectively, expressed in polar coordinates.

P acts on a vector of eigenfunction values in a stencil and produces interpolated values in the center square of the stencil (fig. 2.5).

The pseudoinverse L^+ is computed using a singular value decomposition as follows,

$$L^+ = V\Sigma^+U^*$$

where $L = U^*\Sigma V$ is a singular value decomposition of L and Σ is a diagonal matrix of singular values with

$$\Sigma_{ii}^+ = \begin{cases} \Sigma_{ii}^{-1} & \text{if } \Sigma_{ii} > \epsilon \\ \Sigma_{ii} & \text{otherwise} \end{cases}$$

where $\epsilon = \gamma\epsilon_{double}\Sigma_{11}$ where $\epsilon_{double} = 1.11e-16$ is the difference between one and the smallest IEEE double precision floating point number greater than one. Singular values less than ϵ are considered to be zero within working precision.

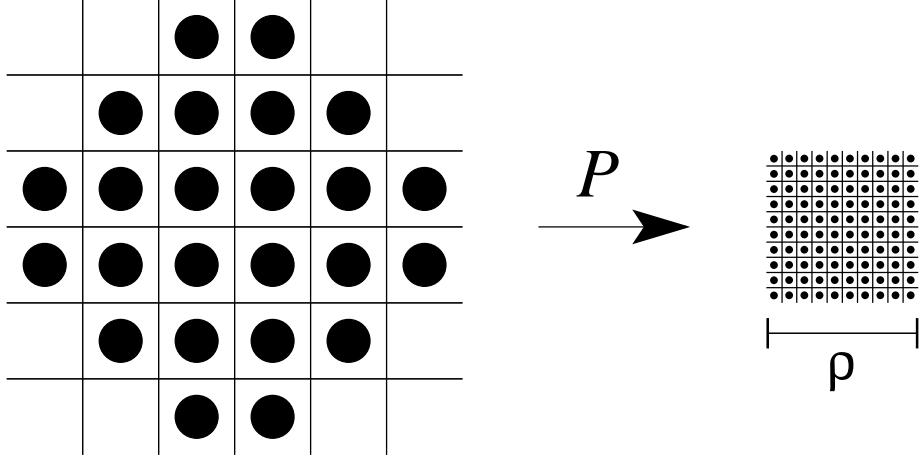


Figure 2.5: Visualization of the action of P as a mapping. P maps eigenfunction values on a coarse grid to eigenfunction values on a fine grid.

2.3 Analysis

2.3.1 Probability of ambiguous region

Here we derive an upper bound on the probability of a trouble region using random plane waves 1.6. Applying the Jacobi-Anger expansion [2]

$$\exp(ikr \cos \theta) = \sum_{l \in \mathbb{Z}} i^l \exp\{il\theta\} J_l(kr)$$

and the fact that $\hat{n}_n \cdot \mathbf{r} = r \cos(\theta - \frac{2\pi n}{N})$ produces

$$u(\mathbf{r}) = \Re \left[\sum_{l \in \mathbb{Z}} i^l \tilde{\omega}_l \exp\{il\theta\} J_l(kr) \right] \quad (2.3)$$

where $\mathbf{r} = (r, \theta)$ in polar coordinates and $\tilde{\omega}_l$ are given by

$$\tilde{\omega}_l = \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \sum_{n=1}^N \omega_n i^l \exp\left\{-il\frac{2\pi n}{N}\right\}$$

Fixing N , we can express the transformation which takes $\vec{\omega}$ to $\vec{\tilde{\omega}}$ (as vectors) as a matrix

$$\vec{\tilde{\omega}}^{(N)} = A^{(N)} \vec{\omega}^{(N)}$$

where entries of $A^{(N)}$ are given by

$$a_{mn} = \frac{1}{\sqrt{N}} i^m \exp\left\{-im\frac{2\pi n}{N}\right\} = \frac{1}{\sqrt{N}} \exp\left\{-im\left(\frac{2\pi n}{N} - \frac{\pi}{2}\right)\right\}$$

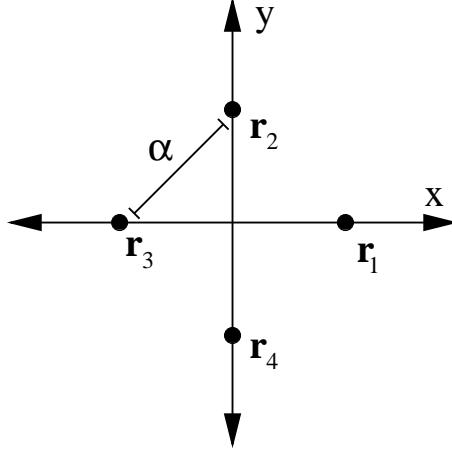


Figure 2.6: Four evaluation points in coordinate system

The operator A is simply a discrete Fourier transform and therefore acts on $\omega_n \sim \mathcal{N}(0, 1)$ i.i.d. to produce $\tilde{\omega}_n \sim \mathcal{N}(0, 1)$ i.i.d.

Expanding the first three terms of 2.3 gives

$$u(\mathbf{r}) = a_0 J_0(kr) + J_1(kr)(a_1 \cos \theta + b_1 \sin \theta) + J_2(kr)(a_2 \cos 2\theta + b_2 \sin 2\theta) + \dots$$

where $a_0 = \Re[\tilde{\omega}_0]$, $a_{1,2} = \Re[\tilde{\omega}_1 - \tilde{\omega}_{-1}]$, and $b_{1,2} = \Im[\tilde{\omega}_1 - \tilde{\omega}_{-1}]$. Note that $a_0 \sim \mathcal{N}(0, 1)$ and $a_1, b_1, a_2 \sim \mathcal{N}(0, \sqrt{2})$.

We now consider four points forming a square of side length α . We define a coordinate system with origin at the center of the square and axes rotated such that each corner of the square falls on either the x - or y -axis (figure 2.6). We label the four points \mathbf{r}_1 through \mathbf{r}_4 . Evaluating the three term expansion of $u(\mathbf{r})$ at the four points gives

$$\begin{aligned} u(\mathbf{r}_1) &= a_0\beta_0 + a_1\beta_1 + a_2\beta_2 \\ u(\mathbf{r}_2) &= a_0\beta_0 + b_1\beta_1 - a_2\beta_2 \\ u(\mathbf{r}_3) &= a_0\beta_0 - a_1\beta_1 + a_2\beta_2 \\ u(\mathbf{r}_4) &= a_0\beta_0 - b_1\beta_1 - a_2\beta_2 \end{aligned}$$

where $\beta_i = J_i\left(k \frac{\alpha}{\sqrt{2}}\right)$. Interpolation is required if $u(\mathbf{r}_1), u(\mathbf{r}_3) < 0$ and $u(\mathbf{r}_2), u(\mathbf{r}_4) > 0$ (or the reverse case) which gives the system of inequalities

$$\begin{aligned} a_0\beta_0 + a_1\beta_1 + a_2\beta_2 &< 0 \\ a_0\beta_0 + b_1\beta_1 - a_2\beta_2 &> 0 \\ a_0\beta_0 - a_1\beta_1 + a_2\beta_2 &< 0 \\ a_0\beta_0 - b_1\beta_1 - a_2\beta_2 &> 0 \end{aligned}$$

which is equivalent to

$$\begin{aligned} a_2 &< 0 \\ a_0 &< a_2 \frac{\beta_2}{\beta_0} \\ a_1 &< a_2 \frac{2\beta_2}{\beta_1} \\ b_1 &< 0 \end{aligned}$$

Thus the probability of a configuration of four points requiring interpolation is given by

$$2 \int_{-\infty}^0 \int_{-\infty}^0 \int_{-\infty}^{a_2 \frac{\beta_2}{\beta_0}} \int_{-\infty}^{a_2 \frac{2\beta_2}{\beta_1}} f(a_0, a_1, b_1, a_2) da_1 da_0 db_1 da_2 \quad (2.4)$$

where the factor of two is inserted to account for the possibility of the inequalities being reversed and $f(a_0, a_1, b_1, a_2)$ is a four dimensional Gaussian distribution

$$f(a_0, a_1, a_2) = \frac{1}{(2\pi)^2 |\Sigma|^{\frac{1}{2}}} \exp \left\{ \mathbf{a}^T \Sigma^{-1} \mathbf{a} \right\}$$

where

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

applying the linear transformation

$$\begin{aligned} x_1 &= a_0 \\ x_2 &= a_0 - \frac{\beta_2}{\beta_0} a_2 \\ x_3 &= a_1 - \frac{2\beta_2}{\beta_1} a_2 \\ x_4 &= b_1 \end{aligned}$$

produces the integral

$$2 \int_{-\infty}^0 \int_{-\infty}^0 \int_{-\infty}^0 \int_{-\infty}^0 f_X(x_0, x_1, x_2, x_3) dx_3 dx_2 dx_1 dx_0 \quad (2.5)$$

where

$$f_X(x_0, x_1, x_2, x_3) = \frac{1}{(2\pi)^2 |(J \Sigma J^T)^{-1}|^{\frac{1}{2}}} \exp \left\{ \mathbf{x}^T J \Sigma J^T \mathbf{x} \right\}$$

is the Gaussian distribution f under the linear transformation and J is the Jacobian of this transformation

$$J = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -\frac{\beta_2}{\beta_0} \\ 0 & 1 & 0 & \frac{2\beta_2}{\beta_1} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

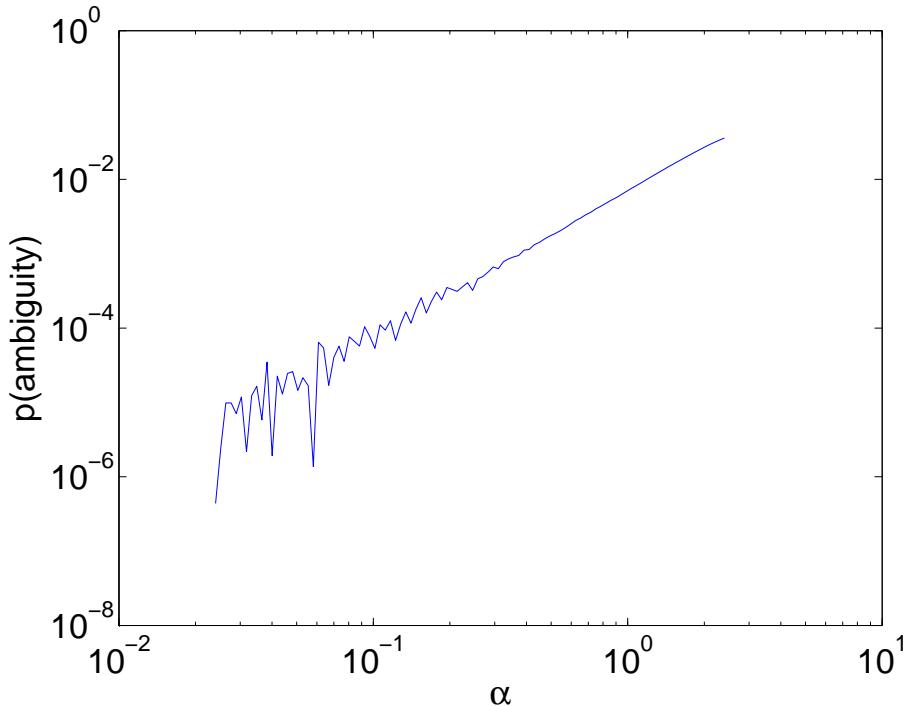


Figure 2.7: Theoretical probability of an ambiguity when sampling at α by 2.5. Small values of alpha produce highly correlated variables in the transformed coordinate system, which produce numerical instabilities in the integration of the Gaussian.

This integral can be computed numerically; figure 2.7 shows the probability of such an error occurring at a randomly chosen two pixel by two pixel square. These results appear to match empirical measures of interpolations per pixel to within a constant factor of approximately 2.5.

The heuristic we use when deciding where to interpolate misses some very rare cases where domains connect between sampled points. See [10] for a more complete characterization of sampling errors when computing nodal domains in two dimensions.

2.4 Counting Nodal Domains

Eigenfunctions are sampled on a regular grid with dimension n_y by n_x ; each point in this grid will be referred to as a “pixel.” Nodal domains are counted by exploring domains pixel-by-pixel, marking each pixel as “counted” in an n_y by n_x bit array once it has been seen. The searching algorithm used to explore each

domain is a hybrid depth- and breadth-first method where for each pixel, the sign of each uncounted neighboring pixel is compared to the sign of the nodal domain and if the sign matches, the neighboring pixel is pushed onto a stack. Exploration then continues by popping a pixel off of the stack.

This hybrid method was chosen because it requires fewer comparisons than a depth-first search (where a pixel may be popped from the stack up to four times, once for each neighbor, versus only once in this method) and allows an efficient stack implementation using a dynamically sized array, whereas a breadth-first search requires a queue, which cannot be implemented as efficiently as a stack. A linked-list implementation of a stack or queue was found to add approximately 15% runtime overhead due to frequent memory allocations. An array implementation of a queue can be accomplished by treating the array as circular but this requires additional comparisons when enqueueing and dequeuing as compared to an array implementation of a stack.

Algorithm 1 Nodal domain counting algorithm

Require: $grid$ is ny by nx matrix containing eigenfunction values on billiard
Require: $\alpha = kh$, M is highest order Bessel function to interpolate with, ρ is ratio to interpolate by
Require: $counted$ is ny by nx and $counted[i][j] = UNCOUNTED$ if the point at i, j is in Ω , else $counted[i][j] = COUNTED$

```

function COUNTNODALDOMAINS( $grid, counted, \alpha, M, \rho$ )
     $interp \leftarrow CreateInterpMatrix(\alpha, M, \rho)$ 
     $i, j, domains \leftarrow 0$ 
    while  $i, j \leftarrow FindNextUnseen(counted, i, j)$  do
         $domains \leftarrow domains + 1$ 
         $FindDomain(grid, counted, i, j, domain\_num, interp)$ 
    end while
    return  $domains$ 
end function

function FINDNEXTUNSEEN( $counted, y, x$ )
    for  $i \in y \dots ny$  do
        for  $j \in 1 \dots nx$  do
            if  $i = y$  and  $j \leq x$  then
                continue
            end if
            if  $counted[i][j] = UNCOUNTED$  then
                return  $i, j$ 
            end if
        end for
    end for
    return  $NULL$ 
end function

```

Algorithm 1 Nodal domain counting algorithm (continued)

```
function FINDDOMAIN(grid, counted, i, j, interp)
    left, above, right, below  $\leftarrow$  FALSE,
    s.push(j, i)
    counted[i][j] = COUNTED
    sign  $\leftarrow$  sign(grid[i][j])                                 $\triangleright s$  is a stack
    while x, y  $\leftarrow$  s.pop() do
        if InGrid(x - 1, y) then
            if sign(grid[y][x - 1]) = sign then
                left  $\leftarrow$  TRUE
                if counted[x - 1][y] = UNCOUNTED then
                    counted[x - 1][y] = COUNTED
                    s.push(x - 1, y)
                end if
            end if
        end if
        ...
         $\triangleright$  Same for above, right, and below
        if InGrid(x - 1, y - 1) and above and left then
            if sign(grid(x - 1, y - 1)) = sign then
                if not IsInterpolated(counted, x - 1, y - 1) then
                    Interpolate(grid, counted, x - 1, y - 1, interp)
                end if
                if ConnectedAboveLeft(counted, x, y) then s.push(x - 1, y - 1)
                end if
            end if
        end if
        ...
         $\triangleright$  Same for below and left, below and right, and above and right
    end while
end function
```

This algorithm has runtime $O(N)$. This is because the method performs a constant number of comparisons for each pixel in the eigenfunction plus $O(N)$ total comparisons searching for an unseen pixel after a nodal domain has been explored. Interpolation occurs $O(N)$ times and each matrix-vector multiply takes constant time because the number of stencil points S and the upsampling ratio ρ are constant.

This algorithm requires $O(N)$ space for the array *counted* which we use to store for each pixel, whether it is in the domain, has been counted already, and has been interpolated. In addition, this method uses a dynamically sized array as a stack whose size is (loosely) bounded above by the number of pixels in the nodal domain being explored.

Chapter 3

Results

3.1 Data Collected

We computed approximately 10^5 eigenfunctions amounting to over 800 GB of data and counted 1.5 billion nodal domains. This data required approximately 8000 CPU hours, running on a cluster with about 30 cores.

3.1.1 Run time

For both billiards considered here $B \sim k$. Thus, in practice $N \gg n$ and $N \gg B$ so it is primarily N (which scales like α^{-2}) that determines the time to compute eigenfunctions. Computation time is dominated by eigenfunction evaluation, which is dominated by basis function evaluations (fig. 3.1).

3.1.2 Prior Work

Bogomolny and Schmit [6] have conducted a small numerical study of their conjectures but investigated only the stadium billiard and collected only a few hundred eigenfunctions. These eigenfunctions were collected around $k = 100$ which we have shown is far from the asymptotic regime (fig. 3.2). Keating, Marklof and Williams [8] have performed a related study on quantum maps, for which they verified all three conjectures of Bogomolny and Schmit, using a few thousand eigenstates (which are much less costly to compute for quantum maps as compared to quantum billiards).

3.2 Mean of number of nodal domains

We find that the scaled mean number of nodal domains for the Sinai billiard converges to $0.0596 \pm 1.724e-5$, which is 4.61% below the value of conjecture 1. This result represents a deviation of 167σ and is therefore very statistically significant.

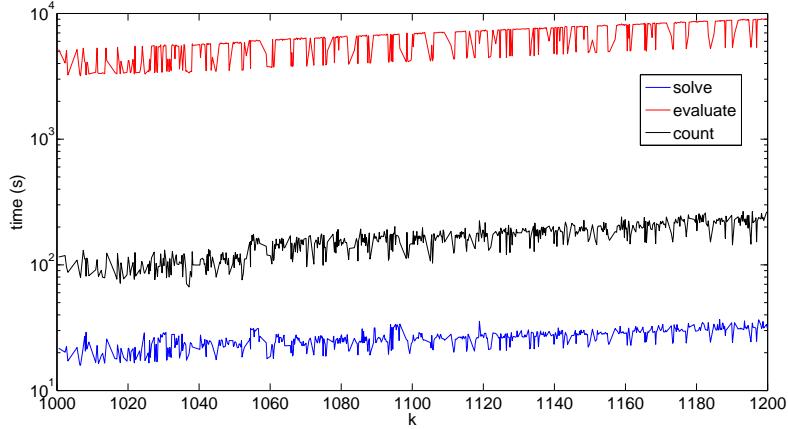


Figure 3.1: Comparison of run times at $\alpha = 0.5$ of solving for basis coefficients, evaluating eigenfunctions, and counting nodal domains. Evaluating eigenfunctions dominates runtime, taking approximately 97% of computation time.

Nodal counts in the stadium billiard approached scaled mean of $0.0535 \pm 3.991e-5$, which is 14.36% below the conjectured value and a deviation of 225σ .

We find that the mean for random plane waves also disagrees with the conjectured value and in fact converges $.0589 \pm 1.42e-4$, which very nearly matches the asymptotic mean of the Sinai billiard. This indicates that the percolation model is not an accurate model of quantum chaos.

The mean number of the nodal domains of the percolation model agrees well with the conjectured value. These were computed on a unit grid as described in 1.4.

In all models, we find a relatively slow convergence toward the asymptotic mean. Bogomolny and Schmit [5] predict a convergence of the form $A + B/k$, which we fit to data to predict asymptotic values. We find fitted values of B between 3 and 5 for the various billiards and models considered, indicating that it is necessary to approach $k 10^3$ to obtain counts within 5% of the asymptotic value.

3.3 Variance of number of nodal domains

The variance of number of nodal domains in quantum chaotic eigenfunctions does not agree with conjecture 2 for either billiard shape (fig. 3.4). The quarter generalized rectangular Sinai billiard has a variance of scaled nodal domains counts of 0.107, which is 2.13 times the value of conjecture 2. The quarter stadium has a variance of scaled nodal domain counts of 2.70, which is over 50 times the conjectured value.

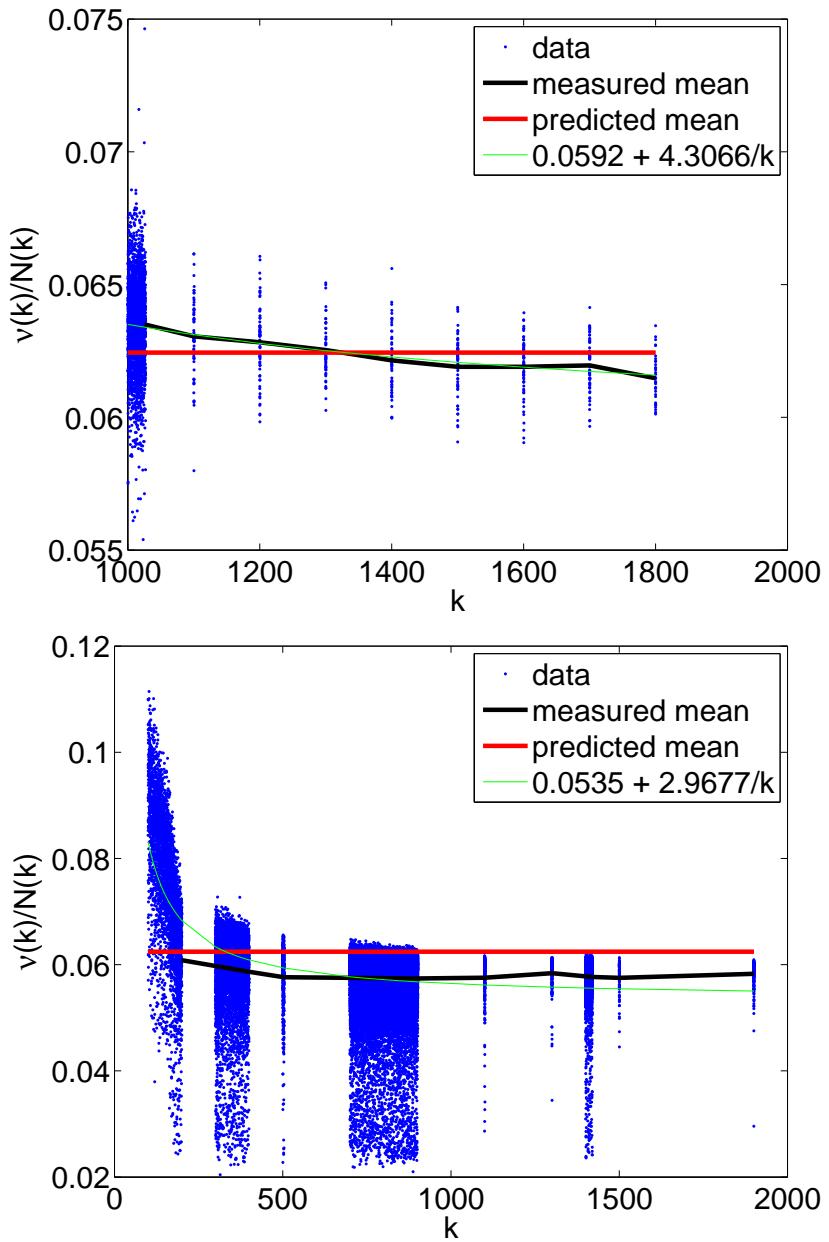


Figure 3.2: Mean number of nodal domains. Above: Sinai; below: stadium. The least-squares best fit of the form $A + B/k$ is shown in green.

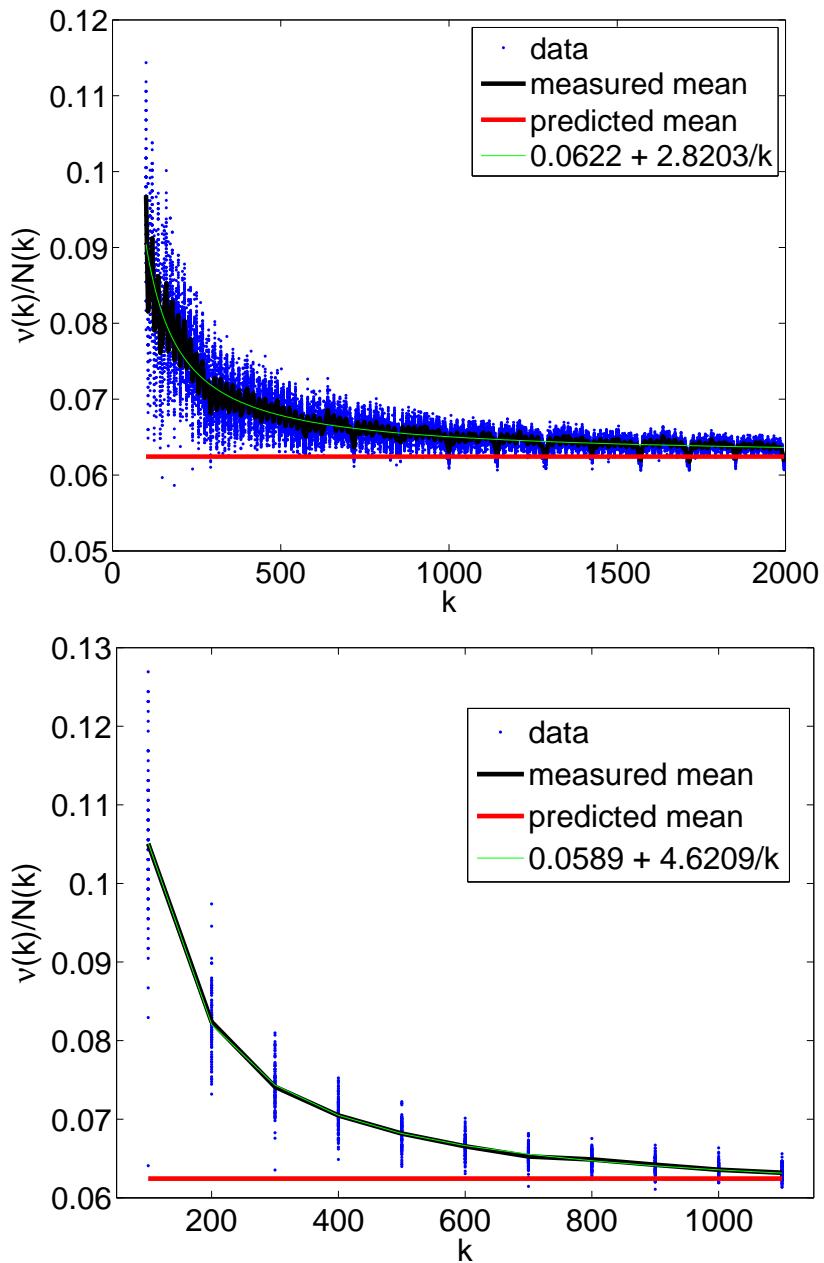


Figure 3.3: Mean number of nodal domains. Above: percolation with $k \in \{100, 106, 112, \dots, 1998\}$; below: random plane waves with $k \in \{100, 200, \dots, 1100\}$ with 100 repetitions at each k .

The random plane wave model has a variance of scaled nodal domain counts of 0.0883, which is again comparable to the Sinai value. The percolation model has a variance of 0.0501, which agrees well with the predicted value, having a relative error of $2.40e-3$.

3.4 Distribution of number of nodal domains

The number of nodal domains for the Sinai billiard is nearly normal with a slight left skew (fig. 3.5).

There is a strong left skew in the distribution of nodal domain counts in the stadium billiard (fig. 3.5). This is due to bouncing ball modes (fig. 1.4), which have much fewer nodal domains than general chaotic eigenfunctions (fig. 3.6). Bouncing ball modes have a high concentration of probability mass in the central rectangular region and are largely responsible for the significantly lowered mean and large variance of nodal domain counts in the quarter stadium billiard. We can therefore identify bouncing ball modes by low probability mass in the quarter circle region. Specifically, we use the “wing tip mass”

$$w = \int_W u^2(\mathbf{r}) d\mathbf{r}$$

where $W = \{(x, y) \in \Omega \mid x \geq 1.1\}$.

3.5 Areas of nodal domains

The distribution of area of nodal domains all follow a power law. Nodal domain areas are scaled by the area of the smallest possible nodal domain $s_{min} = \pi(j_1/k)^2$ where $j_1 \approx 2.4048$ is the first zero of the Bessel function J_0 . The best fit exponent for the Sinai billiard is $2.0567 \pm 8.5524e-4$, which has a relative error of $8.65e-4$ and a difference of 2.079σ from the conjectured value. For the stadium billiard, the best fit exponent is $2.0578 \pm 1.2e-3$, a 0.14% error and a difference of 2.3σ . These deviations of order 2σ are not statistically significant; thus we accept conjecture 3 for both billiard shapes.

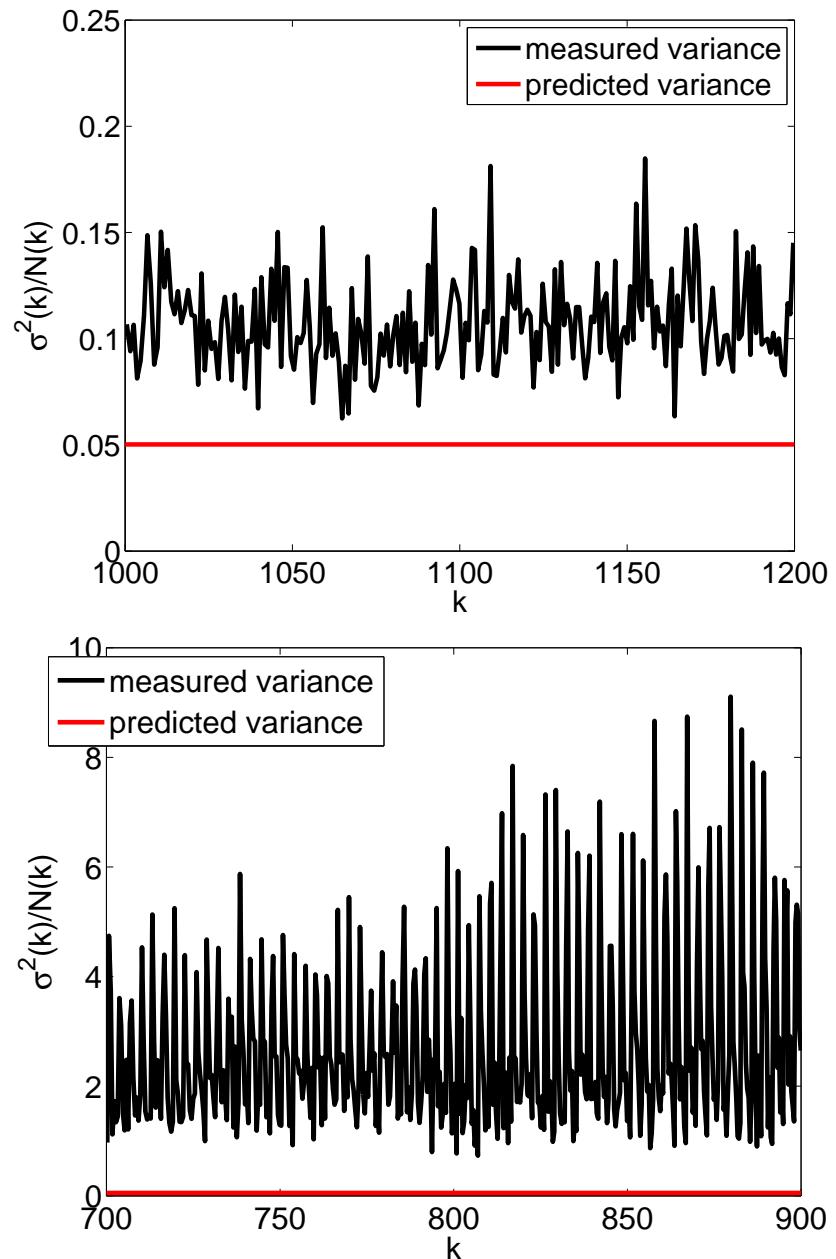
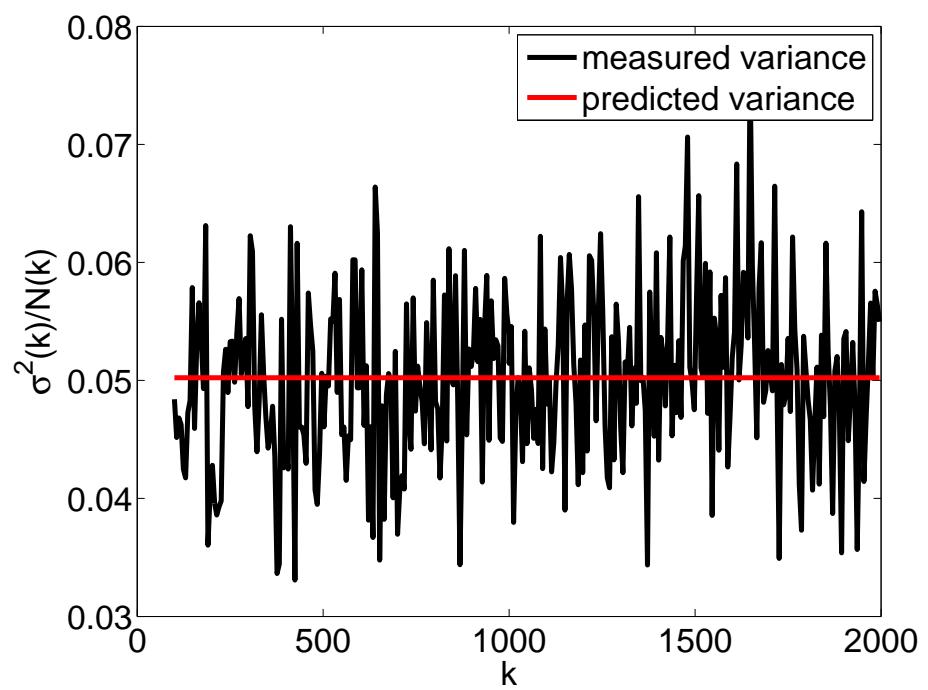


Figure 3.4: Variance of number of nodal domains. Top: Sinai; middle: stadium; bottom: percolation



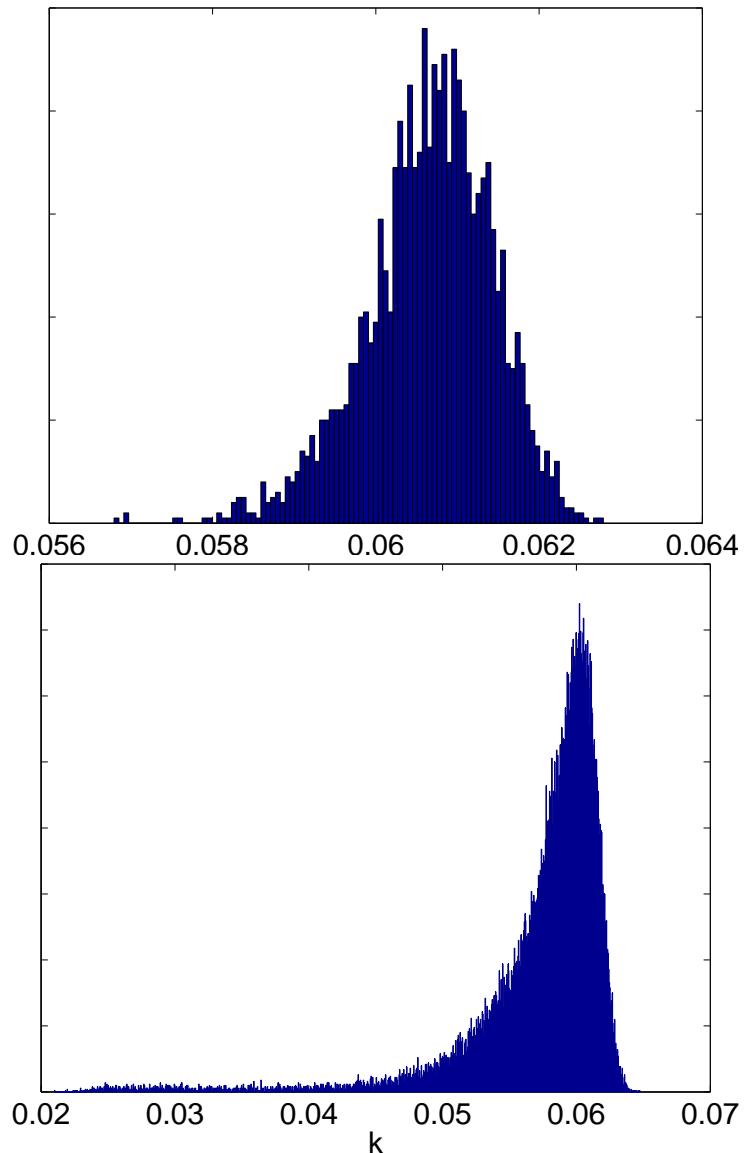


Figure 3.5: Histogram of scaled nodal domain counts. Above: Sinai billiard with $k \in [2000, 2020]$; below: stadium billiard with $k \in [700, 900]$.

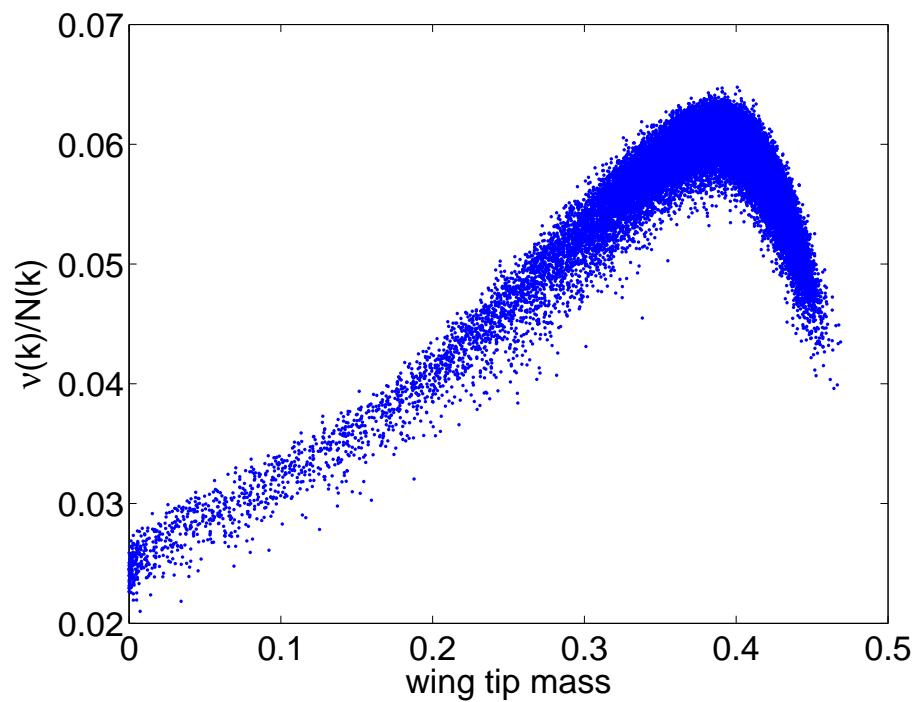


Figure 3.6: Wing tip masses of eigenfunctions of the stadium billiard with $k \in [700, 900]$. The low wing tip mass regions have low nodal counts and skew the overall distribution while lowering the mean.

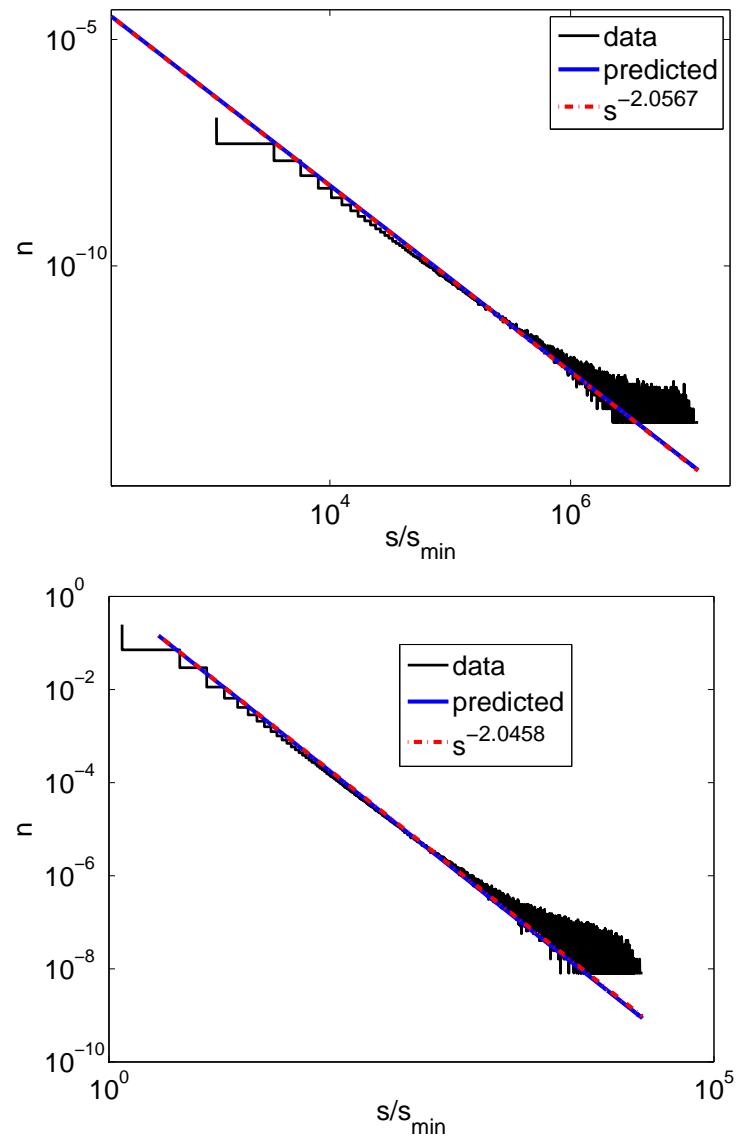


Figure 3.7: Relative frequencies of normalized nodal domain areas. The dashed red line shows the least-squares best fit of the form $s^{-\tau}$. Above: Sinai; below: stadium

Chapter 4

Conclusion

By combining the scaling method of Vergini and Saraceno [12] with adaptive interpolation, we have obtained nodal domain data at energies 400 times higher than previously explored for over 100 times as many eigenfunctions as previously examined [6]. Comparing this data to the predictions of Bogomolny and Schmit [5] from the percolation model, we find a that the distribution of sizes of nodal domains matches, while the mean and variance of number of nodal domains differ. We are the first to collect data from the generalized Sinai billiard, which displays hard chaos and has a mean number of nodal domains nearly in agreement with that determined experimentally for random superpositions of plane waves.

The source code for tools developed here has been released under an open source license [1] and is available for the community to modify and distribute [9].

Appendix A

Definition of Ergodicity

Informally, a mapping T is said to be ergodic if it has homogenous dynamics across its domain, i.e., there are no regions which behave “differently” under the mapping T . Formally, we consider T as a mapping on a probability space. A probability space requires a measure, which is defined over a σ -algebra. Thus we present the following definitions:

Definition 1 (σ -algebra). $\Sigma \subset 2^\Omega$ is a σ -algebra over Ω if:

1. $\emptyset, \Omega \in \Sigma$
2. $\Omega \setminus A \in \Sigma \forall A \in \Sigma$
3.
$$\bigcup_{i=1}^{\infty} A_i \in \Sigma \quad \forall \{A_i\}_{i=1}^{\infty} \text{ such that } A_i \in \Sigma \forall i$$

Definition 2 (Measure). $\mu: \Sigma \rightarrow \mathbb{R}$ is a measure on Σ if:

1. $\mu(A) \geq 0 \forall A \in \Sigma$
2. $\mu(\emptyset) = 0$
3.
$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i)$$

Definition 3 (Measure space). A measure space is a triple (Ω, Σ, μ) where Σ is a σ -algebra over Ω and μ is a measure on Σ .

Definition 4 (Probability space). A probability space is a measure space (Ω, Σ, μ) where $\mu(\Omega) = 1$.

These definitions allow us to formally define ergodicity as:

Definition 5 (Ergodicity). *Let (Ω, Σ, μ) be a probability space. A mapping $T: \Sigma \rightarrow \Sigma$ is ergodic if $T(E) = E \implies \mu(E) = 0$ or $\mu(E) = 1$*

Thus, under an ergodic mapping T , any $E \in \Sigma$ that maps to itself must have measure zero (in which case dynamics on this set are inconsequential) or measure one (in which case the set is almost the entire domain Ω). Furthermore, all $E \in \Sigma$ such that $0 < \mu(E) < 1$ must contain points that map to points outside of E under T . In this sense, T “mixes” points in the domain Ω .

Appendix B

General Solution of the Helmholtz Equation

Here we show that the functions $J_n(kr) \sin(n\theta)$ and $J_n(kr) \cos(n\theta)$ form a complete basis of solutions of (1.1).

In polar coordinates,

$$\Delta = \frac{1}{r} \partial_r(r \partial_r) + \frac{1}{r^2} \partial_{\theta\theta}$$

Thus, (1.1) can be expressed as

$$u_{rr}(r, \theta) + \frac{1}{r} u_r(r, \theta) + \frac{1}{r^2} u_{\theta\theta}(r, \theta) + k^2 u(r, \theta) = 0$$

Using separation of variables we attempt solutions of the form

$$u(r, \theta) = R(r)\Theta(\theta)$$

where $\Theta(\theta)$ is periodic with period 2π . This gives

$$\Theta''(\theta) + n^2\Theta(\theta) = 0$$

and

$$r^2 R''(r) + rR'(r) + r^2 k^2 R(r) - n^2 R(r) = 0$$

The periodicity of $\Theta(\theta)$ requires that

$$\Theta(\theta) = \alpha \sin(n\theta) + \beta \cos(n\theta)$$

where $n \in \mathbb{Z}$. The radial differential equation is known as Bessel's equation and has solutions

$$R(r) = J_n(kr)$$

where J_n is a regular Bessel function and $k \in \mathbb{R}$ is allowed to take discrete values determined by boundary conditions.

Thus, the general solution of (1.1) can be expressed as a sum of the form

$$b_0 J_0(kr) + \sum_{n=1}^{\infty} a_n J_n(kr) \sin n\theta + b_n J_n kr \cos n\theta \quad (\text{B.1})$$

Bibliography

- [1] Gnu general public license. <http://www.gnu.org/licenses/gpl.html>.
- [2] M. Abramowitz and I. A. Stegun, editors. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Dover Publications, New York, 1970.
- [3] A. Barnett. Asymptotic rate of quantum ergodicity in chaotic euclidean billiards. *Communications in Pure and Applied Mathematics*, 59(10):1457–1488, 2006.
- [4] A. Barnett and A. Hassell. Fast computation of high frequency dirichlet eigenmodes via the spectral flow of the interior neumann-to-dirichlet map. 2012.
- [5] E. Bogomolny and C. Schmit. Percolation model for nodal domains of chaotic wavefunctions. *Physical Review Letters*, 88(11), 2002.
- [6] E Bogomolny and C Schmit. Random wavefunctions and percolation. *Journal of Physics A: Mathematical and Theoretical*, 40(47):14033, 2007.
- [7] P. R. Garabedian. *Partial differential equations*. John Wiley & Sons Inc., New York, 1964.
- [8] J. P. Keating, J. Marklof, and I. G. Williams. Nodal domain statistics for quantum maps, percolation, and stochastic loewner evolution. *Phys. Rev. Lett.*, 97:034101, Jul 2006.
- [9] K. Konrad. qc. <https://github.com/ktkonrad/qc>.
- [10] K. Mischaikow and T. Wanner. Probabalistic validation of homology computation for nodal domains. *The Annals of Applied Probability*, 17(3):980–1018, 2007.
- [11] P. O’Connor, J. Gehlen, and E. J. Heller. Properties of random superpositions of plane waves. *Phys. Rev. Lett.*, 58:1296–1299, Mar 1987.
- [12] E. Vergini. Calculation by scaling of highly excited states of billiards. *Physical Review E*, (3):2204–2207, 1995.

- [13] I. Wigman. On the nodal line of random and deterministic laplace eigenfunctions. 2011.