Universidade de São Paulo

Instituto de Física

Estudo experimental de caos quântico com ressonadores acústicos

Thiago Nascimento Nogueira

Orientador: Prof. Dr. José Carlos Sartorelli

Tese de doutorado apresentada ao
Instituto de Física para a obtenção
do título de Doutor em Ciências.

Banca examinadora:

Prof. Dr. José Carlos Sartorelli (IFUSP)

Profa. Dra. Maria Carolina Nemes (UFMG)

Prof. Dr. Alfredo Gontijo de Oliveira (UFMG)

Prof. Dr. Paulo Batista Gonçalves (PUC/RJ)

Prof. Dr. Antônio Domingues dos Santos (IFUSP)

SBI-IFUSP

305M810T4665

São Paulo
2006

Prof. Celso Luiz Lima
Presidente da Comissão de Pós-Graduação

clg. 14/12/06

530.13
N7782
D
ex.1

FICHA CATALOGRÁFICA

**Preparada pelo Serviço de Biblioteca e Informação
do Instituto de Física da Universidade de São Paulo**

Nogueira, Thiago Nascimento

Estudo experimental de caos quântico com
ressonadores acústicos. São Paulo, 2006.

Tese (Doutorado) - Universidade de São Paulo.
Instituto de Física-Deptº Física Geral.

Orientador: Prof. Dr. José Carlos Sartorelli

Área de Concentração: Física

Unitermos: 1. Mecânica estatística; 2. Caos quântico;
3. Ressonadores acústicos; 4. Caos ondulatório;
5. Bilhares.

USP/IF/SBI-002/2007

Resumo

Construímos um arranjo experimental para o estudo de ressonadores acústicos, que têm sido considerados como análogos clássicos de bilhares quânticos. O equipamento mantém estabilidade durante vários dias, o que é uma condição necessária para a obtenção de espectros de autofrequências com a resolução requerida para a caracterização precisa destes sistemas.

Caracterizamos 7 amostras, que são placas de alumínio com espessura < 2 mm e que possuem as seguintes geometrias: dois estádios de Sinai, com e sem dessimetrização planar; três triângulos sendo um equilátero, um retângulo e outro escaleno, este com todos os ângulos agudos e irracionais em unidades de π ; além de duas amostras circulares, com e sem dessimetrização planar.

Observamos que três amostras apresentam estatísticas GOE, uma 2GOE, uma semi-Poisson, uma Poisson com perda de níveis, e outra aparentemente intermediária entre a GOE e a 2GOE, que não foi possível classificar.

A qualidade dos dados também permitiu a obtenção das energias dos espectros, onde obtivemos resultados coerentes com a classificação acima.

Abstract

We have built an experimental apparatus to study acoustic resonators which have been considered as classical analogs of quantum billiards. The equipment was able to keep the stability during several days, which is a requirement to the precise eigenfrequency measurements allowing a characterization of the systems.

We have characterized 7 samples made of aluminum plates with thickness smaller than 2 mm having the following geometries: two Sinai's stadiums (with and without planar symmetry), an equilateral triangle, a rectangle triangle, and a scalene triangle with three acute and irrational angles, and two circular shaped samples, with and without planar symmetry.

We observed that three of the samples followed the GOE statistics (the asymmetrical Sinai stadium, the rectangle triangle and the scalene one). The asymmetrical Sinai stadium was described by 2GOE statistics, the equilateral triangle by the semi-Poisson, the symmetrical circle by a Poisson with missing levels and the asymmetrical circle has statistics apparently between 1GOE and 2GOE which was not possible to classify.

The high quality of data allowed us to calculate the spectra energies and we found these results compatible with the previous one.

Agradecimentos

Ao Prof. Sartorelli, por sua orientação dedicada e pelo ensinamento de técnicas experimentais de medidas e análises extremamente valiosas. Agradeço também à sua enorme disposição nessa reta final.

Ao Prof. Maurício Porto Pato, que teve importância fundamental no desenvolvimento deste trabalho. Apesar de não ser formalmente um co-orientador, quando for utilizada a primeira pessoa do plural, a sua pessoa está incluída.

À Dra. Adriana Tufaile, com quem tive a oportunidade de trabalhar junto em todas as etapas do doutorado.

Ao Fernando Horita, excelente técnico de laboratório, cuja ajuda foi de extrema importância na fase inicial da montagem do arranjo experimental.

Ao Dr. Odair Gimenez, pelo empréstimo do controlador de temperatura e pela paciência demonstrada ao esperar “só mais um pouquinho”.

Ao Prof Reynaldo, por sua eterna disposição em ajudar e por manter o astral do grupo lá em cima.

Ao Prof. Alberto Tufaile, pelos auxílios desde a época da iniciação científica.

Ao Marcelo, amigo, pelas discussões proporcionadas desde os tempos da graduação. Esses papos, sem dúvida, enriqueceram muito meu universo científico.

Aos outros colegas do grupo, Viviane, Pedro, Tiaguinho, Boris, Ludmila, Caroline, Rogério, Dariel, Desirée e Daniel, pelas ajudas em geral pela convivência tão agradável.

Ao Eloy, a quem devo atribuir o crédito pelas figuras tridimensionais desta tese.

À Renata, pela imensa ajuda nas correções, e pela paciência e sensibilidade no exercício desta função.

À Regina, por compreender minhas decisões e compartilhá-las com tamanha proximidade; e pelas referências, é claro.

À Fapesp, pelo suporte financeiro.

Enfim, e principalmente, aos meus pais, pela postura de vida e por prezarem tanto o desenvolvimento intelectual. Espero conseguir passar estes ensinamentos ao meu filho, Acauã.

À Regina e ao Acauã,
que está a caminho...

*"Das conjunções improváveis
de órbitas instáveis
é que eu me mantenho.*

*E venho arrimado nuns versos
tropeçando universos
pra achar-te no fim"*

Paulo Vanzolini, em Samba Abstrato.

Conteúdo

1. Introdução.....	1
2. Aspectos teóricos.....	3
2.1 Sistemas hamiltonianos clássicos.....	3
2.1.1 Sistemas integráveis	3
2.1.2 Sistemas não-integráveis	4
2.1.3 Bilhares bidimensionais clássicos	5
2.2 Sistemas quânticos	8
2.2.1 Limite semi-clássico de um sistema integrável.....	8
2.2.2 Limite semi-clássico de um sistema ergódico	10
2.2.3 Espectro de autovalores.....	11
2.2.4 Degenerescência dos autovalores	12
2.3 Teoria de Matrizes Aleatórias (TMA).....	14
2.3.1 <i>Ensemble</i> Gaussiano Ortogonal (E_{1G})	14
2.3.2 <i>Ensemble</i> Gaussiano Simplético (E_{4G})	15
2.3.3 <i>Ensemble</i> Gaussiano Unitário (E_{2G})	16
2.3.4 <i>Ensembles</i> circulares	16
2.3.5 Sistemas integráveis	17
2.3.6 Distribuição de primeiros vizinhos e rigidez espectral	17
2.3.7 Grandezas termodinâmicas.....	21
2.4 Elastodinâmica e ressonadores acústicos	23
2.4.1 Modelo tridimensional infinito.....	24
2.4.2 Modelo de Placas.....	25
3. Construção do laboratório	31
3.1 O aparato experimental	31
3.1.1 Controle de temperatura	35
3.1.2 Sistemas de excitação e detecção	40
3.1.3 O analisador de espectros	42
3.2 Confecção das amostras	42
3.2.2 Programas de monitoramento e de aquisição de dados	46

4. Aquisição e tratamento dos dados	53
4.1 Aquisição de dados.....	53
4.2 Tratamento dos dados.....	57
4.2.1 Detecção inicial dos picos de ressonância.....	57
4.2.2 Ajuste dos picos e obtenção do espectro	60
4.2.3 Desdobramento do espectro	63
4.2.4 O método de corte (<i>cut-off</i>)	65
5. Resultados	68
5.1 Curvatura de níveis.....	68
5.2 Estádio de Sinai	71
5.2.1 Estádio de Sinai com Dessimetrização Planar	75
5.3 Termodinâmica de espectros gaussianos finitos.	78
5.3.1 Simulações com matrizes aleatórias.....	80
5.3.2 Termodinâmica dos estádios de Sinai.	82
5.4 Ressonadores Triangulares.....	86
5.4.1 Triângulo T_1 (equilátero).....	87
5.4.2 Triângulo T_2 (retângulo).....	89
5.4.3 Triângulo T_3 (agudo)	91
5.5 Sistema circulares.....	93
5.5.1 Ressonador circular	96
5.5.2 Ressonador circular com dessimetrização planar.....	100
Conclusões.....	103
Apêndice A - Listagem dos Programas desenvolvidos.....	105
Apêndice B - Cópia do artigo publicado	106
Bibliografia.....	107

Capítulo 1

Introdução

Os avanços provenientes do estudo de sistemas determinísticos não-lineares permitiram a observação de que uma grande quantidade de sistemas físicos pode apresentar divergência exponencial de suas órbitas no espaço de fases, o que caracteriza um sistema caótico. Nestes casos, a previsão de uma órbita particular só é válida para um curto intervalo de tempo, já que as incertezas são ampliadas exponencialmente com o tempo. Criaram-se então abordagens alternativas que buscam a caracterização global do sistema ao invés da descrição individual de suas órbitas.

Além disso, com o nascimento da física quântica, o mundo passa a ser tratado como intrinsecamente probabilístico e o espaço de fases concebido classicamente perde o seu significado, já que em qualquer sistema quântico uma condição inicial pode gerar diferentes evoluções. Assim, a definição usual de caos não pode ser aplicada diretamente para sistemas quânticos e se faz necessária a construção de uma nova abordagem a fim de identificar características que permitam a diferenciação dos tipos de movimento neste novo mundo.

Uma das alternativas propostas é o estudo dos sistemas no limite semi-clássico, onde a ação é muito grande quando comparada com $\hbar$. Essa abordagem tem se mostrado adequada para a compreensão de sistemas integráveis e quase-integráveis, mas é muito difícil de ser aplicada para a grande maioria dos sistemas não-integráveis.

E. P. Wigner, ao estudar o comportamento de núcleos complexos, propôs uma abordagem em que a hamiltoniana é substituída por uma matriz cujos elementos são aleatórios e dependem apenas das simetrias do sistema. Conhecida hoje como Teoria de Matrizes Aleatórias (TMA), ela tem se mostrado eficiente para a caracterização de sistemas físicos não-integráveis.

Os bilhares são sistemas particularmente interessantes nesse contexto de conexão entre a física clássica e a quântica e têm sido objeto exaustivo

de estudos nesses últimos anos. Em especial, o estudo de bilhares ondulatórios bidimensionais pode trazer grandes contribuições para o entendimento de bilhares quânticos, já que neste caso a equação de Helmholtz é semelhante à de Schödinger.

O primeiro arranjo experimental desenvolvido para medir os espectros de emissão acústica de ressonadores foi construído por R. L. Weaver, no final da década de 80 do século XX [1]. Nesta montagem, a amostra é excitada através de uma força impulsiva e as autofrequências são detectadas a partir da transformada de Fourier das medidas da oscilação da amostra ao longo do tempo. Com este arranjo experimental, Weaver obteve espectros com centenas de pontos, onde cada ponto é uma autofrequência do ressonador.

Em meados dos anos 90, M. Oxborrow e C. Ellegaard construíram um aparato semelhante ao de Weaver [2]. Os constantes aperfeiçoamentos fizeram com que esta montagem perdesse quase todos os aspectos da montagem original e este refinamento permitiu a obtenção de espectros com milhares de pontos [3]. A modificação mais significativa foi a introdução de um equipamento que permitiu a medida da resposta do ressonador quando excitado controladamente por uma onda senoidal em função de uma frequência da excitação. Este processo, entretanto, aumentou consideravelmente o tempo de aquisição de dados, gerando a necessidade de um controle bastante rigoroso de alguns parâmetros que podem influenciar no espectro como, por exemplo, a temperatura da amostra.

Em colaboração com o Prof. Maurício Porto Pato construímos um aparato experimental para o estudo de ressonadores acústicos, que é o sistema experimental mais barato entre os “análogos clássicos” de bilhares quânticos. Esses sistemas apresentam espectros com grande número de autofrequências que permitem boas análises estatísticas de sistemas integráveis e não integráveis com aplicação da Teoria de Matrizes aleatórias.

No Capítulo 2 apresentamos os aspectos teóricos, no Capítulo 3 descrevemos detalhadamente a montagem do arranjo experimental e sua evolução para obtenção de dados de boa qualidade. A aquisição e tratamento dos dados são apresentados no Capítulo 4, e a discussão das análises dos dados no Capítulo 5. Por fim, apresentamos as conclusões no capítulo 6.

Capítulo 2

Aspectos teóricos

Apresentaremos alguns aspectos da teoria clássica de sistemas hamiltonianos para desenvolvermos, nas seções seguintes, as discussões sobre duas abordagens sugeridas para a análise das manifestações de caos em sistemas quânticos. Posteriormente, discutimos alguns aspectos da teoria de matrizes aleatórias. Finalmente, apresentaremos alguns aspectos da teoria elastodinâmica.

2.1 Sistemas hamiltonianos clássicos

Num sistema hamiltoniano clássico com N graus de liberdade, um determinado estado é representado por um ponto X num espaço de fases $2N$ -dimensional, através de suas coordenadas e momentos generalizados.

$$X = (\mathbf{q}, \mathbf{p}) = (q_1, \dots, q_N; p_1, \dots, p_N) \quad (2.1)$$

A evolução deste estado se dará de acordo com as restrições impostas ao sistema pela hamiltoniana H da seguinte maneira:

$$\dot{q} = \frac{\partial H}{\partial p} \text{ e } \dot{p} = -\frac{\partial H}{\partial q} \quad (2.2)$$

Integrando as equações (2.2) podemos obter a evolução temporal do sistema a partir de determinada condição inicial. Entretanto, a obtenção de uma expressão analítica é possível apenas para casos muito particulares.

2.1.1 Sistemas integráveis

O teorema de Arnold [4] postula que sistemas com N graus de liberdade e que possuam N constantes de movimento C_i independentes (isto é, o parêntesis de Poisson entre os pares de constantes é nulo), são integráveis se as C_i forem “suficientemente suaves”. Além disso, nestas

circunstâncias, a trajetória no espaço de fases será confinada a uma superfície Σ N-toroidal.

Um sistema integrável poderá sempre ser escrito num sistema de coordenadas especiais obtidas através de uma transformação canônica: as coordenadas de ângulo-ação. Com esta transformação $(q, p) \rightarrow (\theta, I)$, a hamiltoniana passa a depender explicitamente apenas das variáveis de ação I , que são combinações das constantes C_i . A partir das equações (2.2) é fácil notar que as equações horárias serão escritas da forma:

$$\theta(t) = \theta_0 + \omega(I)t \quad (2.3)$$

A maneira com que a superfície Σ é preenchida por uma trajetória no espaço de fases depende da relação entre as frequências ω_i . No caso geral, estas frequências serão incomensuráveis e uma órbita irá preencher densamente o N-toro Σ , gerando um movimento quase periódico. No entanto, se for possível escrever as relações $\omega_i/\omega_j = m/n$ (para qualquer $i < j$) a órbita no toro será fechada e, portanto, periódica.

2.1.2 Sistemas não-integráveis

Para a vasta maioria dos sistemas físicos, a condição da existência de N constantes de movimento não é válida e, portanto, não é possível integrá-los analiticamente. Neste caso, a superfície Σ em que se dá o movimento não é necessariamente um toro. Os sistemas não-integráveis são divididos em classes diferentes, de acordo com suas características topológicas.

Sistemas quase-integráveis

As hamiltonianas de sistemas quase-integráveis podem ser escritas como a soma de dois termos, sendo que um deles é integrável (H_0) e o outro é não-integrável (H_1):

$$H(q, p) = H_0(q, p) + \varepsilon H_1(q, p) \quad (2.4)$$

Para $\varepsilon = 0$, H é integrável e o movimento se dará num toro. À medida que ε é aumentado, as superfícies toroidais vão sendo destruídas em algumas regiões do espaço de fases, fazendo com que haja o convívio de órbitas regulares e caóticas. O teorema KAM [5, 6, 7] mostra a forma pela qual estes toros são quebrados e quais o serão. De maneira geral, a introdução da perturbação leva à quebra dos toros racionais, criando nestas regiões ilhas de ressonância onde os movimentos são caóticos.

Sistemas ergódicos

Um sistema ergódico pode ser definido como aquele em que as médias temporais e espaciais são iguais. Isto quer dizer que uma trajetória típica do sistema irá explorar toda a superfície de energia no espaço de fases. Para sistemas físicos, a ergodicidade é um forte indício de movimento caótico.

Sistemas pseudo-integráveis

Os sistemas pseudo-integráveis, assim como os integráveis, possuem N constantes de movimento. Entretanto, ao contrário do que se poderia imaginar, seu movimento se dará numa superfície Σ N -dimensional que não possui as características topológicas de um N -toro. Nestes sistemas, os campos vetoriais gerados pelas constantes de movimento possuem singularidades, violando um dos requerimentos para a validade do teorema de Arnold.

2.1.3 Bilhares bidimensionais clássicos

Um bilhar pode ser entendido como um sistema constituído por uma partícula livre, confinada em uma região do espaço delimitada pelo contorno B . A hamiltoniana deste sistema pode ser escrita de maneira geral da seguinte forma:

$$H(x, y, p_x, p_y) = \begin{cases} \frac{p_x^2 + p_y^2}{2}, & \text{dentro do contorno } B \\ \infty, & \text{fora do contorno } B \end{cases} \quad (2.5)$$

O movimento desta partícula consiste em segmentos de retas combinados com reflexões em B e suas características são extremamente sensíveis à geometria do contorno. Casos como o do bilhar quadrado e circular são exemplos típicos de sistemas integráveis. Outro bilhar que pode ser integrável é o triângulo, dependendo dos ângulos de seus vértices. Um triângulo equilátero, por exemplo, é um sistema integrável. Reprodiremos a seguir um argumento geométrico simples apresentado por Berry [8] para corroborar este fato. O argumento consiste em mostrar que é possível construir um toro bidimensional que caracteriza as órbitas deste bilhar.

Inicialmente, devemos observar que uma reflexão em B pode ser

substituída por uma transposição em que não se altere a trajetória da partícula, como ilustrado na Figura 2.1. Para o caso de polígonos cujos ângulos sejam racionais de π , é sempre possível substituir o estudo do bilhar pelo de uma partícula livre numa superfície fechada formada por células com a geometria de **B**. Para um triângulo eqüilátero, esta superfície pode ser construída a partir de um hexágono regular, como pode ser observado na Figura 2.2. Juntando-se os caminhos (identificados por a, b e c) chegamos à conclusão de que a superfície de movimento de um triângulo eqüilátero é topologicamente igual a um toro e, conseqüentemente, integrável.

Figura 2.1: Ilustração exemplo de como substituir uma reflexão numa parede por uma transposição.

Figura 2.2: Passos para a construção da superfície toroidal na qual se dá o movimento de uma partícula num bilhar com formato de triângulo eqüilátero.

Em 1970, Sinai provou que um bilhar quadrado com um obstáculo central circular de raio R é um sistema ergódico, para qualquer valor de $R > 0$ [9]. Este bilhar ficou conhecido como bilhar de Sinai e um desenho esquemático é apresentado na Figura 2.3 (a). A ergodicidade deste sistema se deve à desfocalização das órbitas ao se chocarem com o círculo central. O bilhar de Sinai, além de ergódico, é um sistema caótico.

Outro exemplo de bilhar ergódico é o estádio de Bunimovich¹ [10], que consiste em dois semicírculos de raio R , conectados por linhas retas de comprimento L , conforme a Figura 2.3 (b). Este sistema é ergódico para qualquer que seja o valor de $L > 0$. Apesar de o choque com os semicírculos levar à convergência das órbitas, a posterior divergência entre elas antes do próximo choque gera a ergodicidade que, neste caso, também vem acompanhada de movimentos caóticos.

Figura 2.3: Dois exemplos de bilhares caóticos: (a) Bilhar de Sinai, (b) Estádio de Bunimovich.

Bilhares poligonais pseudo-integráveis foram estudados por Richens e Berry [11] que verificaram que alguns polígonos que tem vértices com ângulos racionais em π podem ter duas constantes de movimento sem, entretanto, serem integráveis. Um exemplo deste sistema é o “bilhar quadrado toroidal” que consiste num bilhar quadrado, com um obstáculo central também quadrado. Ao contrário dos bilhares de Sinai e de Bunimovich, este sistema não é ergódico e possui 2 constantes de movimento (P_x^2, P_y^2). Entretanto, este sistema é não integrável, já que não é possível a construção de uma superfície toroidal que represente o espaço de fases como no caso do triângulo equilátero. A justificativa para este comportamento é de que vetores construídos a partir da constante de movimento são singulares nos vértices do quadrado interno, e por isso não satisfazem as condições necessárias para a validade do teorema de Arnold.

¹ Este estádio muitas vezes aparece na literatura apenas como nome de “estádio”.

2.2 Sistemas quânticos

Como dito anteriormente, a definição clássica de espaço de fases não se aplica para sistemas quânticos. Em 1932, Wigner [12] introduziu a função de distribuição no espaço de fases $W(\mathbf{q}, \mathbf{p})$, relacionada a um determinado estado quântico $\Psi(\mathbf{q}, \mathbf{p})$. A função é dada por

$$W(\mathbf{q}, \mathbf{p}) = \frac{1}{(2\pi\hbar)^N} \int \dots \int d^N X e^{i\frac{\mathbf{p} \cdot \mathbf{X}}{\hbar}} \Psi^*(\mathbf{q} - \mathbf{X}/2) \Psi(\mathbf{q} + \mathbf{X}/2) \quad (2.6)$$

e é a transformada de Fourier do produto Ψ^* com Ψ nas posições separadas por $\mathbf{X}$. W é a generalização quântica da densidade de pontos no espaço de fases clássico. Esta função possui toda a informação sobre o estado quântico. Em particular, a densidade de probabilidade na posição e momento podem ser obtidas a partir de

$$\begin{aligned} |\Psi(\mathbf{q})|^2 &= \int \dots \int d^N p W(\mathbf{q}, \mathbf{p}) \\ |\overline{\Psi}(\mathbf{p})|^2 &= \int \dots \int d^N q W(\mathbf{q}, \mathbf{p}) \end{aligned} \quad (2.7)$$

2.2.1 Limite semi-clássico de um sistema integrável

Para entendermos o comportamento de W no caso de um sistema classicamente integrável, devemos tentar associar as funções de onda do sistema quântico com as superfícies toroidais do sistema clássico. Para isso, imaginemos uma função de onda $\Psi(\mathbf{q})$ e uma superfície Σ , N dimensional, embebida num espaço de fases $2N$ -dimensional [8]. Σ pode ser escrita localmente como uma função $\mathbf{p}(\mathbf{q})$ e corresponde a um conjunto de pontos no espaço de fases clássico. Podemos agora fazer uma transformação canônica $(\mathbf{q}, \mathbf{p}) \rightarrow (\mathbf{Q}, \mathbf{P})$, através da função geratriz $S(\mathbf{q}, \mathbf{P})$, para que a densidade de pontos na superfície Σ seja uniforme:

$$\begin{aligned} S(\mathbf{q}, \mathbf{P}) : (\mathbf{q}, \mathbf{p}) &\rightarrow (\mathbf{Q}, \mathbf{P}) \\ \mathbf{p} &= \nabla_{\mathbf{q}} S \\ \mathbf{Q} &= \nabla_{\mathbf{p}} S \end{aligned} \quad (2.8)$$

Queremos agora associar Σ a uma função do tipo

$$\Psi(\mathbf{q}) = a(\mathbf{q}) e^{i b(\mathbf{q})}, \quad (2.9)$$

de maneira que $|a|^2$ seja proporcional à densidade de pontos no espaço de

fases $(\mathbf{q}, \mathbf{p})$. Portanto:

$$a^2(q) = K \left| \frac{dQ}{dq} \right| = K \det \left| \frac{\partial^2 S}{\partial q_i \partial p_j} \right|. \quad (2.10)$$

Para a determinação de $b(\mathbf{q})$, utilizamos a relação de de Broglie:

$$b(\mathbf{q} + \delta \mathbf{q}) - b(\mathbf{q}) = \nabla b \cdot \delta \mathbf{q} = \mathbf{k} \cdot \delta \mathbf{q} = \frac{\mathbf{p} \cdot \delta \mathbf{q}}{\hbar} \quad (2.11)$$

Portanto,

$$\nabla b(\mathbf{q}) = \frac{\mathbf{p}(\mathbf{q})}{\hbar} \quad (2.12)$$

$$b(\mathbf{q}) = \frac{1}{\hbar} \int_{\mathbf{q}_0}^{\mathbf{q}} \mathbf{p}(\mathbf{q}') d\mathbf{q}' = \frac{S(\mathbf{q}; \mathbf{P})}{\hbar} \quad (2.13)$$

Podemos construir a função de onda plana local substituindo as equações (2.12) e (2.13) em (2.11):

$$\Psi(\mathbf{q}) = K \left| \det \frac{\partial^2 S(\mathbf{q}; \mathbf{P})}{\partial q_i \partial q_j} \right|^{1/2} \exp \left(\frac{i}{\hbar} S(\mathbf{q}; \mathbf{P}) \right) \quad (2.14)$$

Esta função de onda é associada à Σ através de $\mathbf{P}$ e possui grande utilidade no estudo de sistemas semi-clássicos, pois, nesses casos, é comprovado que a evolução temporal de Ψ para um estado Ψ' através da equação de Schrödinger é consistente com a evolução de Σ para Σ' pela ação da hamiltoniana clássica.

Podemos finalmente estudar o comportamento da função de Wigner no limite semi-clássico pela substituição de (2.14) em (2.6) e estudando o limite $\hbar \rightarrow 0$. Com esta substituição, chegamos a

$$W(\mathbf{q}, \mathbf{p}) = \frac{K^2}{(2\pi\hbar)^N} \int \dots \int d^N X \left| \det \left(\frac{\partial^2 S(\mathbf{q} + \mathbf{X}/2; \mathbf{P})}{\partial q_i \partial p_j} \right) \det \left(\frac{\partial^2 S(\mathbf{q} - \mathbf{X}/2; \mathbf{P})}{\partial q_i \partial p_j} \right) \right|^{1/2} \times \quad (2.15)$$

$$\exp \left(\frac{i}{\hbar} \left(\int_{\mathbf{q}-\mathbf{X}/2}^{\mathbf{q}+\mathbf{X}/2} \mathbf{p}(\mathbf{q}', \mathbf{P}) d\mathbf{q}' - \mathbf{p} \cdot \mathbf{X} \right) \right).$$

À medida que $\hbar \rightarrow 0$, o integrando passa a oscilar rapidamente e é dominado pela região $\mathbf{X}=0$. Fazemos então a seguinte aproximação:

$$\int_{\mathbf{q}-\mathbf{X}/2}^{\mathbf{q}+\mathbf{X}/2} \mathbf{p}(\mathbf{q}', \mathbf{P}) d\mathbf{q}' \approx \mathbf{X} \cdot \mathbf{p}(\mathbf{q}). \quad (2.16)$$

Aproximando agora o determinante para $X=0$, a integral pode ser resolvida:

$$W(\mathbf{q}, \mathbf{p}) \approx K^2 \left| \det \left(\frac{\partial^2 S(\mathbf{q}; \mathbf{P})}{\partial \mathbf{q}_i \partial \mathbf{P}_j} \right) \right| \delta(\mathbf{p} - \mathbf{p}(\mathbf{q}; \mathbf{P})). \quad (2.17)$$

Vimos na seção 2.1.1 que, para um sistema integrável clássico, é possível fazer uma transformação de variáveis para as de ângulo-ação. No caso de um sistema quântico, a ação deve satisfazer a seguinte condição para que $\Psi(\mathbf{q})$ seja univocamente determinada em $\mathbf{q}$:

$$\mathbf{I}_m = \left(\mathbf{m} + \frac{\boldsymbol{\alpha}}{4} \right) \hbar, \quad (2.18)$$

onde $\mathbf{m}$ é um conjunto de números quânticos inteiros positivos e $\boldsymbol{\alpha}$ são os índices de Maslov.

Com uma alteração de variáveis em (2.17) e utilizando (2.18), podemos finalmente escrever:

$$W(\mathbf{q}, \mathbf{p}) \approx \frac{1}{(2\pi)^N} \delta(\mathbf{I}(\mathbf{q}, \mathbf{p}) - \mathbf{I}_m). \quad (2.19)$$

Esta equação nos mostra que, no limite semi-clássico de um sistema integrável, a função de Wigner se concentra na região do toro da solução clássica.

2.2.2 Limite semi-clássico de um sistema ergódico

Para um sistema não-integrável, em que o movimento não está confinado em um toro, devemos elaborar a seguinte hipótese [8]:

“Cada autoestado semi-clássico possui uma função de Wigner concentrada na região explorada por uma órbita típica num tempo infinito”.

No caso de um sistema ergódico, uma órbita típica deverá preencher toda a superfície da energia no espaço de fases e cada estado quântico corresponderá a uma superfície de energia. Com a hipótese acima postulada chegamos à seguinte expressão, já normalizada, para a função de Wigner:

$$W(\mathbf{q}, \mathbf{p}) \approx \frac{\delta(E - H(\mathbf{q}, \mathbf{p}))}{\int \dots \int d^N q d^N p \delta(E - H(\mathbf{q}, \mathbf{p}))}. \quad (2.20)$$

A função delta é agora unidimensional e reflete o fato de que a função de Wigner W está espalhada em uma vasta região do espaço de fases, ao contrário do caso regular em que ela está concentrada nas regiões dos toros.

Com esta análise pudemos estabelecer um critério de diferenciação entre sistemas integráveis e ergódicos no limite semi-clássico. A próxima abordagem a ser apresentada busca a diferenciação destes movimentos a partir de seus espectros de autovalores.

2.2.3 Espectro de autovalores

Para sistemas regulares, podemos utilizar a quantização semi-clássica do toro a fim de obtermos o espectro de energia no limite de $\hbar \rightarrow 0$. Se escrevermos a hamiltoniana em função das ações I_m , as energias E_m podem ser escritas como

$$E_m = H(I_m) \quad (2.21)$$

Para um sistema quase-integrável, o teorema KAM garante a existência de toros em algumas regiões do espaço de fases e, se a região de movimentos irregulares é pequena se comparado com $\hbar^N$, a equação (2.21), associada com a teoria perturbativa, pode trazer informações satisfatórias sobre E_m .

Para um sistema geral, os autovalores podem ser obtidos resolvendo-se a equação

$$H|\Psi\rangle = E|\Psi\rangle. \quad (2.22)$$

Mesmo não sabendo a expressão dos autoestados $|\Psi\rangle$, é possível expandi-los em termos de um conjunto completo $|\varphi_m\rangle$. Dessa forma, devemos resolver o determinante secular

$$\det_{m,n} \{ \langle \varphi_m | H | \varphi_n \rangle - E \delta_{m,n} \} = 0. \quad (2.23)$$

A equação acima, apesar de exata, não é apropriada para abordar o comportamento de sistemas caóticos no limite semi-clássico, pois costuma ser de difícil resolução. Em alguns casos específicos, ela pode ser utilizada como ponto de partida para algumas aproximações, como no caso de

bilhares com $N=2$.

2.2.4 Degenerescência dos autovalores

Mesmo não conseguindo obter uma expressão útil para o espectro de energia de sistema ergódicos, é possível chegarmos a algumas conclusões importantes sobre sua estrutura. Um aspecto interessante de se analisar refere-se à tendência de cruzamento ou não dos níveis de energia quando parâmetros do sistema são alterados. Para este caso, estudaremos sistemas dessimetrizados, dado que a presença de simetrias produz degenerescências que não são interessantes ao estudo proposto. O bilhar de Sinai, por exemplo, possui simetrias referentes às retas $x=0$ e $y=0$. Uma dessimetrização possível consiste na análise apenas dos modos ímpares, sendo equivalente ao estudo do bilhar apresentado na Figura 2.4.

Figura 2.4: Bilhar de Sinai dessimetrizado.

Para estudarmos a dinâmica de níveis, vamos imaginar uma hamiltoniana H próxima a outra H^* , que possui um autovalor degenerado:

$$\begin{cases} H^* |u\rangle = E^* |u\rangle \\ H^* |v\rangle = E^* |v\rangle \\ \langle u | v \rangle = 0 \end{cases} \quad (2.24)$$

O sistema

$$\begin{aligned} H &= H^* + \Delta H \text{ e} \\ E &= E^* + \Delta E, \end{aligned} \quad (2.25)$$

pode ser estudado a partir da teoria de perturbação de sistemas degenerados, e os autoestados $|\Psi\rangle$ de H serão expandidos em torno de $|u\rangle$ e $|v\rangle$ da seguinte forma:

$$|\Psi\rangle = \cos \chi |u\rangle + \sin \chi |v\rangle \quad (2.26)$$

Resolvendo a equação de autovalores de H , com o auxílio das equações (2.24) e (2.25), chega-se à condição:

$$\Delta E_{\pm} = \langle u | \Delta H | u \rangle + \langle v | \Delta H | v \rangle \pm \left[\left(\langle u | \Delta H | u \rangle - \langle v | \Delta H | v \rangle \right)^2 + 4 \left(\langle u | \Delta H | v \rangle \right)^2 \right]^{1/2}. \quad (2.27)$$

Para que $\Delta E_+ = \Delta E_-$, o discriminante deve ser igual a zero. Por ser uma soma de dois quadrados, as duas condições deverão ser satisfeitas e, tipicamente, é necessária a variação de dois parâmetros [8].

Imaginando que $E = \Delta E_{\pm}(A, B)$, onde A e B são os dois parâmetros em análise, a conexão das superfícies no espaço E, A, B terá tipicamente a forma de um cone duplo cujo vértice (E^* , A^* , B^*) representa o ponto onde ocorre a degenerescência (ponto diabólico). Para um maior esclarecimento, apresentamos na Figura 2.5 (a) a representação deste cone com o ponto diabólico. Se apenas um parâmetro é variado, os pontos diabólicos dificilmente serão atingidos e o que se observará é uma repulsão dos níveis similar ao observado na Figura 2.5 (b).

Figura 2.5: (a) Ilustração dos cones por onde transitam os autovalores de um sistema hamiltoniano perto de um "ponto diabólico", em que é possível a observação do cruzamento de níveis. (b) Repulsão de níveis observada quando se varia apenas um parâmetro do sistema.

A regra acima apresentada não é válida para sistemas integráveis [13]. Neste caso, a variação de apenas um parâmetro já será suficiente para que haja cruzamento de níveis no espaço E, A (como mostra a Figura 2.6).

Figura 2.6: Dinâmica de níveis característica de um sistema integrável. Nestes sistemas não há repulsão de níveis e é possível observar o cruzamento de níveis quando apenas um parâmetro é alterado.

Vemos, portanto, que a existência de cruzamentos de níveis com alteração de apenas um parâmetro é um poderoso critério para a distinção de sistemas que são classicamente caóticos ou integráveis no limite semi-clássico.

2.3 Teoria de Matrizes Aleatórias (TMA)

A idéia da utilização de *ensembles* de matrizes aleatórias para a descrição de sistemas físicos foi primeiramente proposta por Wigner no estudo de núcleos excitados. Em seu postulado está a idéia de que, nestes casos, o grau de complexidade da hamiltoniana é tão grande que os elementos das matrizes aparentemente não apresentam nenhum tipo de correlação e seus elementos podem ser substituídos por números aleatórios. Nesta abordagem, as particularidades do sistema dão lugar ao estudo de estatísticas que devem seguir comportamentos universais, ditados apenas pelas simetrias do sistema.

Nas seções seguintes, apresentaremos os *ensembles* gaussianos e, posteriormente, os *ensembles* circulares, propostos por F. J. Dyson.

2.3.1 Ensemble Gaussiano Ortogonal (E_{1G})

Qualquer sistema que apresente momento angular total inteiro e

invariância em relação à reversão temporal e também aqueles com momento angular semi-inteiro, que sejam invariantes a reversão temporal e por rotação, podem ser representados por hamiltonianas reais e simétricas. Além disso, por serem operadores hermiteanos, os elementos dessas matrizes serão reais.

Dentro deste contexto é proposto o *ensemble* ortogonal gaussiano (E_{1G} ou GOE), em que se espera que a estatística de níveis de energia seja a mesma dos sistemas acima mencionados e que seus comportamentos clássicos sejam caóticos.

O E_{1G} é formalmente definido no espaço T_{1G} das matrizes reais simétricas devendo satisfazer duas condições [14, 15]:

1. O *ensemble* é invariante sob toda a transformação

$$H \rightarrow W^T H W \quad (2.28)$$

de T_{1G} nele mesmo, onde W é qualquer matriz real ortogonal.

2. Os elementos H_{jk} , $k \geq j$ são estatisticamente independentes.

O primeiro requerimento se deve ao fato de que o *ensemble* deve ser independente da base escolhida para representar o sistema e, desta forma, a probabilidade de que um sistema de E_{1G} se encontre num elemento de volume dH é invariante sob transformações reais ortogonais.

Da segunda condição, temos que a função densidade de probabilidade pode ser escrita como um produto de funções em que cada uma delas depende de apenas um elemento da matriz:

$$P(H) = \prod_{k \leq j} f_{kj}(H_{kj}) \quad (2.29)$$

2.3.2 Ensemble Gaussiano Simplético (E_{4G})

Queremos agora considerar os sistemas que possuem momento angular total semi-inteiro e são invariantes por reversão temporal, mas não possuem simetrias de rotação. Para estes sistemas, tomamos o *ensemble* E_{4G} , que é definido no espaço T_{4G} , das matrizes hermiteanas auto-duais.

A matriz auto-dual tem como característica apenas a invariância por reversão temporal, de maneira que o espaço T_{1G} está contido no espaço T_{4G} . As duas propriedades formais que definem o E_{4G} (também chamado de GSE) são [14, 15]:

1. O *ensemble* é invariante sob qualquer transformação

$$H \rightarrow S^{-1}HS \quad (2.30)$$

de T_{4G} nele mesmo, onde S é qualquer matriz unitária simplética.

Uma matriz para ser unitária e simplética deve satisfazer às seguintes condições:

$$\begin{aligned} SS^\dagger &= 1 \text{ e} \\ S^\dagger JS &= \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \end{aligned} \quad (2.31)$$

onde I é a matriz identidade.

2. Os vários elementos linearmente independentes de H são estatisticamente independentes.

2.3.3 Ensemble Gaussiano Unitário (E_{2G})

Os sistemas que não possuem simetria em relação à reversão temporal são caracterizados pelo *ensemble* gaussiano unitário. A única restrição às matrizes neste caso é que elas sejam hermiteanas. As matrizes do E_{2G} (ou GUE) devem ter as seguintes propriedades [14]:

1. A probabilidade $P(H)dH$ deste *ensemble* irá pertencer ao elemento de volume

$$dH = \prod_{k \leq j} dH_{kj}^{(0)} \prod_{k < j} dH_{kj}^{(1)}, \quad (2.32)$$

onde $dH_{kj}^{(0)}$ e $dH_{kj}^{(1)}$ são, respectivamente, as partes reais e imaginárias de dH_{kj} . Este elemento de volume deve ser invariante sob uma transformação

$$H \rightarrow U^{-1}HU \quad (2.33)$$

de T_{2G} nele mesmo, onde U é qualquer matriz unitária.

2. Os vários elementos linearmente independentes de H são estatisticamente independentes.

2.3.4 Ensembles circulares

Ao introduzir um novo tipo de *ensemble*, F. J. Dyson afirma que as

dificuldades em se calcular as médias dos *ensembles* gaussianos limitam o progresso dos trabalhos baseados nessa teoria. Ele também aponta para o fato de que os elementos das matrizes H não têm o mesmo peso e, conseqüentemente, as interações não são igualmente prováveis. A fim de corrigir estes problemas inerentes aos *ensembles* gaussianos, Dyson introduz os *ensembles* circulares [14, 16].

Os *ensembles* circulares são representados por matrizes unitárias S . Seus autovalores são números complexos da forma $e^{(i\theta_j)}$, com $j = 1, 2, \dots, N$, distribuídos em torno do círculo unitário. Apesar de não haver uma conexão precisa entre H e S , assume-se que S é uma função de H e, assim, os ângulos θ_i são funções dos níveis de energia ε_i . Esta função pode ser imaginada, por exemplo, como

$$S = e^{iH\tau} \quad \text{ou} \quad S = \frac{1 - iH\tau}{1 + iH\tau}. \quad (2.34)$$

Os *ensembles* E_{1C} , E_{2C} e E_{4C} serão construídos, respectivamente, a partir dos mesmos argumentos de simetria de E_{1G} , E_{2G} e E_{4G} . Além disso, de maneira similar aos *ensembles* Gaussianos, o E_{1C} é definido no espaço das matrizes simétricas unitárias, E_{2C} das matrizes unitárias auto-duais e E_{4C} no espaço das matrizes unitárias em geral.

2.3.5 Sistemas integráveis

É importante ressaltar novamente que devem seguir as estatísticas dos *ensembles* acima descritos os sistemas classicamente caóticos. Para sistemas integráveis, espera-se que os níveis de energia sejam completamente descorrelacionados uns dos outros. Desta forma, um sistema classicamente integrável e com dimensão $N \geq 2$ pode ser tratado estatisticamente como um processo de Poisson, sendo representado por uma matriz cujos elementos diagonais são aleatórios e os elementos fora da diagonal são iguais a zero.

2.3.6 Distribuição de primeiros vizinhos e rigidez espectral

Duas grandezas bastante utilizadas para a caracterização de sistemas físicos no limite semi-clássico são a distribuição de primeiros vizinhos (DPV) e a rigidez espectral (Δ_3). Discutiremos a seguir as suas propriedades para cada um dos *ensembles* definidos.

Distribuição de Primeiros Vizinhos (DPV)

A distribuição de primeiros vizinhos é uma função que mede a correlação entre dois níveis vizinhos. Define-se $P(S)dS$ como sendo a probabilidade de se encontrar dois níveis vizinhos a uma distância S e a função $P(S)$ é a função de distribuição de primeiros vizinhos. Para um processo Poissônico, é simples demonstrar que

$$P(S) = e^{-S}. \quad (2.35)$$

Porém, o cálculo desta função para os *ensembles* gaussianos é uma tarefa extremamente difícil. Wigner propôs que a função de distribuição de primeiros vizinhos seja dada por

$$P(S) = \frac{\pi}{2} S e^{-\frac{\pi}{4} S^2}. \quad (2.36)$$

Este resultado pode ser alcançado através de dois argumentos simples [14]. Dado um nível numa posição E , a probabilidade de se encontrar um nível entre E e $E+S$, para S pequeno, é diretamente proporcional a S . O segundo argumento consiste em extrapolar o primeiro argumento para todos os S 's e assumir que a probabilidade nos vários intervalos de tamanho S/m são mutuamente independentes. Com isto temos que

$$P(S) = \lim_{m \rightarrow \infty} \prod_{r=0}^{m-1} \left(1 - \frac{Sr}{m} \frac{1}{m} a \right) aS = aS e^{-\frac{1}{2}aS^2}. \quad (2.37)$$

A partir da condição de normalização de (2.37) chega-se a (2.36). A expressão proposta por Wigner é muito próxima ao resultado exato do E_{1G} , que pode ser escrita como uma série infinita:

$$P(S) = \frac{\pi^2}{6} S - \frac{\pi^4}{60} S^3 + \frac{\pi^4}{270} S^4 + \frac{\pi^6}{1680} S^5 + \dots \quad (2.38)$$

A comparação entre os dois resultados é apresentada na Figura 2.7. É importante ressaltar que a hipótese de repulsão linear entre os níveis para $S \rightarrow 0$ é fundamental para a semelhança dos resultados. Na verdade, a função proposta por Wigner pode ser obtida considerando um E_{1G} com dimensão $N=2$.

Figura 2.7: Comparação de $P(S)$ entre a distribuição de Wigner e de um ensemble E_{IG} [17].

Apresentamos a seguir os resultados obtidos para $N=2$ nos ensembles E_{2G} e E_{4G} [17]:

E_{2G} ($N=2$):

$$P(S) = \frac{32}{\pi^2} S^2 e^{-\frac{4}{\pi} S^2} \quad (2.39)$$

E_{4G} ($N=2$):

$$P(S) = \frac{2^{18}}{3^6 \pi^3} S^4 e^{-\frac{64}{9\pi} S^2} \quad (2.40)$$

A Figura 2.8 mostra $P(S)$ para os quatro casos discutidos acima. Vale notar que, para um sistema integrável, os níveis tendem a se agrupar e $P(S) \rightarrow 1$ quando $S \rightarrow 0$. Já para os sistemas caóticos, os ensembles $E_{\beta G}$ possuem repulsão de níveis de ordem $\beta=1,2,4$ quando $S \rightarrow 0$.

Figura 2.8: Distribuição de primeiros vizinhos $P(S)$ de uma distribuição de Poisson e dos ensembles gaussianos para $N=2$.

Rigidez Espectral (Δ_3)

A rigidez espectral $\Delta_3(L)$ é a média sobre o espectro do desvio quadrático médio da reta que melhor se ajusta a cada intervalo $[\alpha, \alpha+L]$. Esta grandeza foi introduzida por Dyson e Metha como uma maneira de se estudar a correlação dos níveis em diferentes escalas do espectro e pode ser escrita como:

$$\Delta_3(L) = \frac{1}{L} \left\langle \min_{A,B} \int_{\alpha}^{\alpha+L} (N(v) - A v - B)^2 dv \right\rangle_{\alpha} \quad (2.41)$$

Para um sistema classicamente integrável, $\Delta_3(L)$ cresce linearmente com L e tem a expressão dada por

$$\Delta_3(L) = \frac{L}{15}. \quad (2.42)$$

No caso dos *ensembles* gaussianos, é possível obter expressões analíticas assintóticas para L grande, que são mostradas a seguir [18]:

$$\begin{aligned}
\beta = 1: \quad \Delta_3(L) &= \frac{1}{\pi^2} \left[\ln(2\pi L) + \gamma - \frac{5}{4} - \frac{\pi^2}{8} \right] + O(L^{-1}) \\
\beta = 2: \quad \Delta_3(L) &= \frac{1}{2\pi^2} \left[\ln(2\pi L) + \gamma - \frac{5}{4} \right] + O(L^{-1}) \\
\beta = 4: \quad \Delta_3(L) &= \frac{1}{4\pi^2} \left[\ln(4\pi L) + \gamma - \frac{5}{4} + \frac{\pi^2}{8} \right] + O(L^{-1})
\end{aligned} \quad (2.43)$$

onde γ é a constante de Euler.

A utilização destes resultados não costuma ser aconselhada na comparação com resultados experimentais, dada a finitude dos espectros obtidos. Para estes casos, convém comparar os resultados obtidos com simulações numéricas.

Convém também introduzir a expressão obtida para a sobreposição de m espectros independentes. Neste caso a rigidez espectral é dada por [17]:

$$\Delta_3(L) = \sum_{i=1}^m \Delta_{3i}(g_i L), \quad (2.44)$$

onde a somatória é feita sobre os valores de Δ_3 obtidas para cada espectro i . Os valores g_i são as densidades relativas de cada espectro, de maneira que $\sum_{i=1}^m g_i = 1$.

2.3.7 Grandezas termodinâmicas

A utilização de grandezas termodinâmicas foi sugerida por Dyson como uma boa maneira de confrontar teoria com dados experimentais [16]. Para calcular as grandezas termodinâmicas de um $E_{\beta C}$, utilizaremos como exemplo um gás coulombiano confinado num anel de raio unitário [14]. Imaginando este sistema num universo bidimensional, podemos escrever sua energia como:

$$W = \frac{1}{2} \sum \dot{\theta}^2 - \sum_{j < k} \ln |e^{\theta_j} - e^{\theta_k}|. \quad (2.45)$$

Observando que a energia potencial é independente de $\dot{\theta}$, a parte cinética da função de partição pode ser resolvida de maneira trivial e a parte relativa à energia potencial é dada por

$$\Phi_N(\beta) = (2\pi)^{-N} \int_0^{2\pi} \dots \int_0^{2\pi} e^{-\beta(W-W_0)} d\theta_1 \dots d\theta_N, \quad \beta = \frac{1}{K_B T} \quad (2.46)$$

onde a energia

$$W' = W - W_0, \quad (W_0 = -\frac{1}{2} N \ln N) \quad (2.47)$$

foi renormalizada para que se tenha energia potencial nula quando a temperatura tende a infinito ($\beta \rightarrow 0$). É possível provar que, para qualquer N positivo inteiro e β real ou complexo, a função de partição pode ser escrita da seguinte maneira:

$$\Phi_N(\beta) = \Gamma\left(1 + \frac{1}{2}\beta N\right) \left[\Gamma\left(1 + \frac{1}{2}\beta N\right) \right]^{-N} \quad (2.48)$$

A partir desta função de partição, é possível calcular exatamente a energia livre do sistema por partícula:

$$f_N(\beta) = -\frac{1}{N} \frac{1}{\beta} \ln \Phi_N(\beta) = \frac{1}{2} \ln N - \frac{1}{N} \frac{1}{\beta} \ln \left(1 + \frac{1}{2}\beta N\right) + \frac{1}{\beta} \ln \left[\Gamma\left(1 + \frac{1}{2}\beta\right) \right] \quad (2.49)$$

Observando agora o sistema no limite termodinâmico $N \rightarrow \infty$:

$$f(\beta) = \frac{1}{\beta} L\left(\frac{1}{2}\beta\right) + \frac{1}{2} \left[1 - \ln\left(\frac{1}{2}\beta\right) \right], \text{ onde} \quad (2.50)$$

$$L(z) = \ln \Gamma(1+z) \quad (2.51)$$

As grandezas energia por partícula $u(\beta)$, entropia por partícula $s(\beta)$ e calor específico por partícula podem agora ser facilmente calculados:

$$\begin{aligned} u(\beta) &= -\frac{1}{N} \frac{\partial}{\partial \beta} \ln \Phi = f(\beta) - \beta \frac{\partial f(\beta)}{\partial \beta} = \frac{1}{2} \left[L\left(\frac{1}{2}\beta\right) - \ln\left(\frac{1}{2}\beta\right) \right] \\ s(\beta) &= \beta^2 \frac{\partial f(\beta)}{\partial \beta} = \frac{1}{2} \beta \left[L\left(\frac{1}{2}\beta\right) - 1 \right] - L\left(\frac{1}{2}\beta\right) \\ c(\beta) &= -\beta^2 \frac{\partial u(\beta)}{\partial \beta} = -\frac{1}{4} \beta^2 L''\left(\frac{1}{2}\beta\right) + \frac{1}{2} \beta \end{aligned} \quad (2.52)$$

Calculando os valores de $L(z)$ e suas derivadas para valores de β fisicamente relevantes, pode-se construir a Tabela 2.1[14]:

	$\beta \rightarrow 0$	$\beta = 1$	$B = 2$	$\beta = 4$	$\beta \rightarrow \infty$
U	$\frac{1}{2}[1-\gamma-\ln(\frac{1}{2}\beta)]$ + $(\pi^2/24)\beta + \dots$	$1 - \frac{1}{2}(\gamma+\ln 2)$ ≈ 0.365	$\frac{1}{2}(1-\gamma)$ ≈ 0.211	$\frac{3}{4} - \frac{1}{2}(\gamma+\ln 2)$ ≈ 0.115	$\frac{1}{2\beta} - \frac{1}{6\beta^2} + \dots$
S	$-\frac{1}{2}\beta + (\pi^2/48)\beta^2$ + $\dots$	$\frac{1}{2}(1-\gamma-\ln \pi)$ ≈ -0.361	$-\gamma$ ≈ -0.577	$1 - 2\gamma - \ln 2$ ≈ -0.848	$\frac{1}{2}[1-\ln(\pi\beta)]$ - $\frac{1}{3\beta} + \dots$
C	$\frac{1}{2}\beta - (\pi^2/24)\beta^2 +$...	$\frac{3}{2} - \pi^2/8$ ≈ 0.266	$2 - \pi/6$ ≈ 0.355	$7 - \frac{3}{2}\pi^2$ ≈ 0.420	$\frac{1}{2} - \frac{1}{3\beta} + \dots$

Tabela 2.1: Valores das grandezas termodinâmicas u , s e c para diferentes valores de β .

Podemos enunciar os resultados obtidos sob a forma de um teorema [14]:

“Sejam $z_1, z_2, \dots, z_N$ autovalores de uma matriz unitária tirada de um *ensemble* $E_{\beta C}$. A estatística

$$W - W_0 = \frac{1}{2} N \ln N - \sum_{i < j} \ln |z_i - z_j| \quad (2.53)$$

possui valor médio NU e desvio quadrático médio $\beta^{-1}(NC)^{\frac{1}{2}}$, com os valores de U e C listados na Tabela 2.1.”

2.4 Elastodinâmica e ressonadores acústicos

Apesar de muitas vezes comparados com bilhares, os ressonadores acústicos são sistemas mecânicos ondulatórios, cujos comportamentos são estudados pela elastodinâmica. As equações básicas de movimento da elastodinâmica foram originalmente escritas em torno de 1820 por Cauchy e Poisson. Entretanto, a alta complexidade com que as órbitas se acoplam nas bordas dos ressonadores faz com que soluções analíticas, mesmo que aproximadas, possam ser encontradas apenas em sistemas com geometrias bastante simples, com alto grau de simetria.

Apresentaremos, a seguir, alguns aspectos da teoria elastodinâmica, tratando inicialmente das soluções das equações de onda se propagando num sólido infinito. Posteriormente apresentaremos modelos que buscam a descrição da propagação de ondas em placas ressonadoras, que são o

objeto de estudo desta tese.

2.4.1 Modelo tridimensional infinito

Este é um dos sistemas mais simples de ser tratado pela elastomecânica, já que não apresenta problemas relacionados às condições de contorno. Para um sólido infinito, isotrópico, homogêneo e livre de forças externas a equação de movimento pode ser escrita como [19, 20]:

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} \quad (2.54)$$

onde ρ é a densidade do material, $\mathbf{u}$ é o vetor de deslocamento e λ e μ as constantes de Lamé.

Considerando uma onda plana

$$\mathbf{u} = \mathbf{u}_0 e^{i(\mathbf{k} \cdot \mathbf{r} - \omega t)}, \quad (2.55)$$

existem duas classes de ondas que se propagarão no sólido de maneira independente. No primeiro caso, assumimos que o deslocamento é paralelo ao vetor de onda, $\mathbf{k} \times \mathbf{u} = 0$, e as ondas são chamadas de longitudinais. Adicionando a restrição à equação (2.54), a equação diferencial será escrita como

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}), \quad (2.56)$$

de onde se pode obter facilmente o valor da velocidade das ondas longitudinais:

$$c_l = \sqrt{\frac{\lambda + 2\mu}{\rho}}. \quad (2.57)$$

A segunda classe, das ondas transversais, é caracterizada por possuir o vetor de onda ortogonal ao deslocamento, $\mathbf{k} \cdot \mathbf{u} = 0$. Neste caso, teremos

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u}, \quad (2.58)$$

e a velocidade será dada por

$$c_t = \sqrt{\frac{\mu}{\rho}}. \quad (2.59)$$

Considerando que λ e μ são sempre positivas, podemos concluir que $c_l > c_t$.

Em geral, as velocidades são escritas em função do coeficiente de Poisson (ν) e do módulo de Young (E), que se relacionam com os coeficientes de Lamé através das expressões

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \quad e \quad \nu = \frac{\lambda}{2(\lambda + \mu)}, \quad (2.60)$$

de maneira que velocidades longitudinais e transversais podem ser escritas da seguinte maneira:

$$c_l = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}, \quad c_t = \sqrt{\frac{E}{2\rho(1+\nu)}}. \quad (2.61)$$

As relações de dispersão para os dois tipos de onda são dadas por

$$k_l = \frac{\omega}{c_l} \quad (2.62)$$

$$k_t = \frac{\omega}{c_t}$$

Na literatura, ondas longitudinais são também chamadas de primárias (P) e as ondas transversais de secundárias (S).

2.4.2 Modelo de Placas

A facilidade no tratamento analítico do movimento ondulatório num meio infinito e isotrópico vem de que, neste caso, as ondas longitudinais e transversais se propagam independentemente. De maneira geral, a presença de contornos no sistema aumentará muito a sua complexidade, pois uma onda puramente longitudinal (ou transversal) incidente num contorno livre pode gerar uma onda refletida que é a superposição de movimentos ondulatórios longitudinais e transversais. Esta forma de acoplamento dos modos torna muito difícil o tratamento analítico de sistemas com contornos.

Para estudarmos ressonadores planares, convém introduzir um modelo de placas infinitas. Inicialmente, vamos considerar o movimento de ondas harmônicas em um material homogêneo de espessura h , e assumiremos que plano-xy coincide com o plano do centro da placa. Neste

sistema, teremos propagação de ondas transversais horizontais (SH), transversais verticais (SV) e primárias (P). A restrição para o modelo de placas é feito de maneira a impedir que haja propagação de ondas na direção da espessura.

Dados um vetor de onda $\mathbf{k}$ e um vetor $\mathbf{n}$ perpendicular à superfície da placa, conforme a Figura 2.9, as ondas SH não caracterizadas por terem deslocamento perpendicular ao plano- $\mathbf{kn}$, enquanto as ondas SV e P possuem deslocamento na direção deste plano.

Figura 2.9: Definição do sistema de coordenadas e do plano- $\mathbf{kn}$ utilizados nas equações elastodinâmicas do modelo de uma placa infinita.

Uma onda SH com deformação na direção y (ou seja, $\mathbf{k}$ está na direção x) pode ser escrita como $\mathbf{u} = (0, u_y, 0)$ e será governada pela equação [21]:

$$\nabla^2 u_y = \frac{1}{c_t^2} \frac{\partial^2 u_y}{\partial t^2}. \quad (2.63)$$

Considerando o conjunto de soluções na forma de uma onda harmônica $u_y = g(z)e^{i(kx - \omega t)}$ teremos, a partir de (2.63), que

$$\frac{d^2 g}{dz^2} + \beta^2 g = 0, \quad (2.64)$$

onde $\beta^2 = \frac{\omega^2}{c_t^2} - k^2$. As soluções de (2.64) serão

$$g(z) = A_1 \sin(\beta z) + A_2 \cos(\beta z). \quad (2.65)$$

Para o caso de contornos livres, a condição de contorno é dada por

$$\frac{\partial u_y}{\partial z} = \frac{\partial g}{\partial z} = 0 \quad (2.66)$$

em $z = \pm \frac{h}{2}$, e será satisfeita de duas maneiras:

$$A_1 = 0 \quad \text{e} \quad \text{sen}\left(\beta \frac{h}{2}\right) = 0 \quad (2.67)$$

ou

$$A_2 = 0 \quad \text{e} \quad \cos\left(\beta \frac{h}{2}\right) = 0. \quad (2.68)$$

Para um determinado número de onda k , as equações (2.67) e (2.68) terão um número infinito de soluções para a frequência ω , regidos pela condição

$$\beta h = n\pi, \quad (2.69)$$

sendo que os modos simétricos ocorrerão para $n=0,2,4,\dots$, satisfazendo a equação (2.67) e os modos anti-simétricos, que satisfazem a equação (2.68), ocorrem com $n=1,3,5,\dots$

Para o modelo de placas, devemos impor a restrição que não haja transmissão de ondas na direção da espessura, o que equivale a dizer que

$$\lambda > 2h, \quad (2.70)$$

onde λ é o comprimento de onda. É fácil notar que esta condição será satisfeita apenas se $n=0$, e a propagação das ondas se dará inteiramente no plano da placa, sem que haja deslocamento na direção z . Para este caso, a classe SH é também chamada de T, ou das ondas transversais,

Se escrevermos relação de dispersão (2.69) em termos da frequência adimensional $\Omega = \frac{\omega h}{c_t}$ e do número de onda $K = kh$, teremos para o modelo de placas que

$$K_T^2 = \Omega^2. \quad (2.71)$$

Seguindo um procedimento análogo ao apresentado para as ondas SH, é possível chegar a seguinte expressão para a relação de dispersão das ondas P e SV [20]:

$$\frac{\tan\left(\frac{\alpha}{2}\right)}{\tan\left(\frac{\beta}{2}\right)} + \left\{ \frac{4\alpha\beta K^2}{(2K^2 - \Omega^2)^2} \right\}^\varepsilon = 0, \quad (2.72)$$

onde $\alpha^2 = (\Omega^2 - K^2)$, $\beta^2 = \left(\frac{\Omega^2}{\kappa^2} - K^2 \right)$, $\kappa = \frac{c_l}{c_t}$ e $\varepsilon = \pm 1$. Para $\varepsilon = +1$, teremos os modos simétricos e, no caso de $\varepsilon = -1$, os modos anti-simétricos.

Assim como no caso dos modos SH, é possível estabelecermos uma frequência de corte Ω_c até a qual não há propagação na direção da espessura. Abaixo desta frequência as ondas P e SV formarão duas classes de modos que se propagam de maneira independente. A partir de Ω_c , as reflexões dessas ondas no contorno gerarão acoplamentos, aumentando significativamente a complexidade do sistema.

As classes de ondas para o caso de $\Omega < \Omega_c$ podem ser caracterizadas em relação à simetria. A classe RL-S¹ é formada por ondas simétricas ($\varepsilon = 1$). Estas ondas são longitudinais e seus deslocamentos se dão principalmente no plano, mas com uma componente fora do plano. Já a classe RL-A é formada pelas ondas anti-simétricas ($\varepsilon = -1$), que possuem deslocamento principalmente fora do plano, mas com uma componente no plano. Estas ondas são também chamadas de flexurais. Na Figura 2.10 é apresentada uma ilustração do deslocamento dos modos transversais (T), longitudinais (RL-S) e flexurais (RL-A).

Figura 2.10: Ilustração do deslocamento das três classes de ondas encontradas em uma placa infinita: transversais (T), longitudinais (RL-S) e flexurais (RL-A).

¹ A nomenclatura utilizada faz referência ao fato de que a expressão do modo de dispersão (2.72) foi originalmente calculada por Rayleigh e Lamb, em 1889. Utilizamos então suas iniciais para designar estes modos.

Para o modelo de placas, a relação de dispersão (2.72) pode ser expandida, chegando-se às seguintes expressões aproximadas [22]:

$$K_{RL-S} \approx \frac{1}{\kappa'} \Omega \left[1 + a_{S,1} \Omega^2 + a_{S,2} \Omega^4 + c_S \Omega^6 \right] e \quad (2.73)$$

$$K_{RL-A} \approx \frac{12^{1/4}}{\sqrt{\kappa'}} \sqrt{\Omega} \left[1 + a_{A,1} \Omega + a_{A,2} \Omega^2 + c_A \Omega^3 \right], \quad (2.74)$$

onde $\kappa' = \sqrt{2/(1-\nu)}$, os coeficientes a 's são funções analíticas da razão de Poisson ν e os coeficientes c 's são obtidos através de ajustes numéricos [22]. Os valores dos coeficientes ν , a 's e c 's são apresentados na Tabela 2.2 para o caso do alumínio:

Material	ν	$a_{S,1}$	$a_{S,2}$	c_S	$a_{A,1}$	$a_{A,2}$	c_A
Alumínio	0,33	$1,83 \times 10^{-1}$	$-7,97 \times 10^{-3}$	$8,68 \times 10^{-4}$	$3,34 \times 10^{-3}$	$1,22 \times 10^{-4}$	$6,53 \times 10^{-4}$

Tabela 2.2: Razão de Poisson ν de coeficientes das relações de dispersão (2.73) e (2.74) para o caso do alumínio.

Como podemos ver, no modelo de placas infinitas teremos três classes de ondas que se propagam independentemente no material. Ao tratarmos ressonadores planares finitos, teremos um sistema extremamente complexo devido ao acoplamento dos modos nas suas bordas. De maneira geral, sabemos que os acoplamentos se darão, sobretudo, entre os modos que se deslocam principalmente no plano, a saber, os modos transversais e longitudinais. Os modos flexurais se propagarão de maneira praticamente independente, de forma que o espectro de uma placa finita deve ser a soma de dois espectros independentes; um formado pelos chamados "*modos no plano*", que são os modos simétricos, e outro pelos modos "*fora do plano*" (flexurais), que são anti-simétricos [23].

Estimativa do número de níveis

Utilizando o teorema de Weyl (1911), podemos estimar o número de médio de níveis existentes até uma frequência f . Para um sistema bidimensional, é possível mostrar que este valor será dado por [24]:

$$N(f) \approx \frac{S}{4\pi h^2} \sum_m K_m^2(\Omega), \quad (2.75)$$

onde S é a área do ressonador e $K_m(\Omega)$ é a relação de dispersão da classe do modo m . Esta equação, conhecida como fórmula de Weyl, foi escrita originalmente para sistemas quânticos, mas pode ser aplicada também a sistemas mecânicos ondulatórios em que os contornos satisfaçam condições de contorno suficientemente regulares [25, 24]. Utilizando as relações de dispersão (2.71), (2.73) e (2.74), e lembrando que os modos simétricos se propagam de maneira independente dos modos anti-simétricos, podemos escrever a fórmula de Weyl como uma soma de dois termos que gerarão espectros desacoplados:

$$N(f) = N_s(f) + N_A(f) = \left[\frac{S}{4\pi h^2} (K_T^2 + K_{RL-S}^2) \right] + \left[\frac{S}{4\pi h^2} K_{RL-A}^2 \right], \quad (2.76)$$

onde a contribuição dos modos simétricos $N_s(f)$ é dada por

$$N_s(f) \approx \frac{S}{4\pi h^2} \left[\Omega^2 + \frac{1}{\kappa^2} \Omega^2 (1 + a_{s,1} \Omega^2 + a_{s,2} \Omega^4 + c_s \Omega^6)^2 \right] \quad (2.77)$$

e a contribuição dos modos anti-simétricos $N_A(f)$ será

$$N_A(f) \approx \frac{S}{4\pi h^2} \left[\frac{\sqrt{12}}{\kappa} \Omega (1 + a_{A,1} \Omega + a_{A,2} \Omega^2 + c_A \Omega^3)^2 \right]. \quad (2.78)$$

A fórmula de Weyl apresentada na equação (2.75) possui validade no limite de altas frequências, onde as dimensões do ressonador são grandes quando comparadas ao comprimento de onda λ . Se quisermos uma expressão que valha também para baixas frequências, devemos considerar os termos de correção N_C , proporcionais ao perímetro L [25]:

$$N_C = \beta_s \frac{L}{4\pi h} K_{LR-A} + \beta_A \frac{L}{4\pi h} (K_T + K_{RL-S}), \quad (2.79)$$

onde β_s e β_A são funções de ν . Para o caso do alumínio, teremos $\beta_s = 2,0858$ e $\beta_A = 1,7126$.

Capítulo 3

Construção do laboratório

Montamos um laboratório para a obtenção de espectros de autofreqüências de ressonadores acústicos, que são considerados análogos clássicos de bilhares quânticos. Dentre os sistemas análogos, ponderamos inicialmente entre o estudo de ressonadores acústicos e o de cavidade de microondas. Concluimos que os ressonadores acústicos possuem diversas vantagens. Além de um custo significativamente menor, e da facilidade na confecção das amostras, é possível obter espectros com maior número de autofreqüências, permitindo assim maior riqueza estatística durante a análise de dados.

Descreveremos a seguir a construção do nosso aparato experimental, o qual, no início, contou com a colaboração do Prof. C. Ellegaard e de K. Schaadt, do Niels Bohr Institute, Dinamarca.

3.1 O aparato experimental

O diagrama do aparato experimental no estágio atual é mostrado na Figura 3.1(a).

O sistema consiste de uma mesa de ferro com tampo de granito e, sobre esta, uma câmara de vácuo com os fornos porta-amostra. Este conjunto está montado sobre uma plataforma de concreto, independente da estrutura do prédio, que é eficiente para evitar que ruídos mecânicos interfiram nas medidas.

A câmara de vácuo é composta por uma campânula de vidro apoiada sobre um disco de aço inox, mostrada na fotografia da Figura 3.1(b). O vácuo é produzido por uma bomba mecânica da Edwards modelo RV8 através de uma tubulação flexível. Foram feitos cinco furos no disco de inox para a instalação de **passadores** (*feed-thrus*) para as conexões elétricas entre a parte externa e interna da câmara. Cada um dos passadores permite oito conexões elétricas. Para isolar as vibrações produzidas pela bomba mecânica, foi feita uma ancoragem mecânica da tubulação flexível com uma caixa de concreto.

Colocamos um sistema de talha motorizada para levantar e abaixar a campânula, facilitando assim o acesso à parte interna da câmara. Este sistema utiliza um motor elétrico e um conjunto de roldanas para que o movimento da cúpula se dê em baixa velocidade.

Para a obtenção de dados utilizamos um analisador de espectros, um amplificador de baixo ruído, dois controladores de temperatura e o conjunto todo foi interfaciado a um computador. Como os sistemas de medição, de aquisição de dados e de controle da temperatura são os mais sensíveis, eles serão descritos separadamente.

Considerando que a estabilização da temperatura e tomada de dados pode durar semanas, todos os equipamentos foram ligados a um "no-break" de 5.5kVA, conectado em paralelo a um gerador a gasolina que entra em funcionamento automaticamente se houver queda da energia elétrica.

(b)

Figura 3.1: (a) Desenho esquemático do aparato experimental no estágio atual. (b) Uma fotografia, com a campânula levantada, mostrando a base de inox, o forno de alumínio e conexões elétricas.

As amostras devem ter poucos e pequenos pontos de contato para termos apenas os modos determinados pela geometria da amostra, por isso esta fica apoiada sobre três esferas de rubi de 1mm de diâmetro, que fazem parte dos transdutores piezelétricos utilizados para excitar (emissor) e de obter a resposta (receptor), que é proporcional à amplitude de vibração. Para a obtenção de um traço, o analisador de espectros (ou apenas **analisador**) envia um sinal elétrico senoidal com frequência (f) e amplitude (A_0) bem determinadas ao transdutor emissor, e o receptor (um dos outros transdutores) converte a vibração do ressonador em sinal elétrico. Esta resposta é enviada ao analisador, que filtra o sinal e nos dá a respectiva amplitude A . A frequência do analisador de espectros é incrementada linearmente com o tempo, e assim obtemos a resposta $A(f)$.

Para melhorar a qualidade dos dados, utilizamos um pré-amplificador de baixo ruído entre o receptor e o analisador de espectros. Por possuir dois canais de entrada, a utilização deste equipamento permite também que dois dos apoios da amostra funcionem como receptores e que, desta maneira, seja possível a aquisição de dois traços simultâneos.

A temperatura da amostra é um parâmetro extremamente crítico, pois pequenas modificações podem gerar alterações significativas nas posições dos picos de ressonância. Para garantir a estabilidade do espectro, desenvolvemos um sistema de controle de temperatura composto por dois fornos, cada um com um controlador de temperatura independente. Desta forma, a amostra é posicionada dentro do forno menor que é, por sua vez, circundado por um segundo forno. A opção pela utilização de dois fornos será explicada na seção 3.1.1, onde o sistema de controle de temperatura será apresentado com mais detalhes.

O sistema de vácuo foi utilizado para evitar o amortecimento das vibrações da amostra causado pela pressão ambiente. Apesar de não alterar a posição dos picos de ressonância, este amortecimento faz com que os picos fiquem mais largos, dificultando a distinção entre picos próximos na fase de análise de dados. O sistema de vácuo funciona também como um auxiliar no controle de temperatura, já que aumenta o isolamento entre os fornos e o ambiente externo à câmara.

Como a aquisição de um traço pode durar mais de uma semana, desenvolvemos um programa, com interface GPIB, para controlar todos os equipamentos necessários e realizar a aquisição dos dados, como veremos em detalhes na seção 3.2.2.

3.1.1 Controle de temperatura

A obtenção experimental de um espectro de autofrequências é feita a partir de um **traço**, que é o registro da resposta de uma amostra a uma excitação senoidal em função da frequência. As autofrequências correspondem às ressonâncias. Na Figura 3.2 é mostrado um trecho de um traço obtido experimentalmente, assim como o espectro extraído (linhas vermelhas verticais). Na seção 4.2, descreveremos com detalhes o processo de tratamento dos dados para que obtenhamos o espectro a partir dos traços medidos.

Figura 3.2: Trecho de um traço obtido experimentalmente e o espectro (linhas verticais vermelhas) obtido a partir deste.

Como dito anteriormente, os espectros são muito sensíveis às variações de temperatura e a boa qualidade dos dados depende de um bom controle de temperatura da amostra. Entretanto, não podemos medir a temperatura diretamente na amostra, pois o contato com o sensor pode alterar o espectro de ressonância.

Desenvolvemos um sistema de controle indireto de temperatura colocando a amostra dentro de um forno metálico com temperatura controlada, permitindo assim que a amostra entre em equilíbrio térmico com o forno através de troca de calor por irradiação. Vale lembrar que a troca de calor por condução foi minimizada ao colocarmos o forno em vácuo.

A temperatura do forno é controlada utilizando um controlador de temperatura PID, Gênesis PC-2001. Este equipamento permite o monitoramento simultâneo da temperatura de até quatro sensores, mas a temperatura pode ser controlada em apenas um canal. A foto do painel do controlador de temperatura é apresentada na Figura 3.3. Para garantir a precisão de leitura máxima do equipamento (1mK), utilizamos um sensor de diodo de silício da LakeShore, modelo DT-670.

A alteração da temperatura do forno é feita com resistores de potência fixados em suas paredes. Por não termos um sistema de resfriamento, a temperatura de estabilização deve ser configurada sempre acima da ambiente.

Figura 3.3: Foto do painel do controlador de temperatura.

Os primeiros testes efetuados com esta montagem mostraram que é possível controlar a temperatura do forno com boa precisão $T = (320,001 \pm 0,007)K$ por longo período de tempo.

Na primeira montagem deste sistema, utilizamos um forno retangular de alumínio medindo 11x11x13 cm, cujas paredes podem ser removidas independentemente, de modo a facilitar a manipulação dos objetos em seu interior. Inicialmente, foram utilizados como fonte de calor 4 resistores de potência de 15 Ω , fixados em duas das paredes laterais. Apresentamos, na Figura 3.4, a foto da montagem atual do forno que, como descreveremos adiante, possui um número maior de resistores.

O fato da temperatura do forno estar estabilizada, vide *Figura 3.5(a)*, não significa que a amostra já entrou em equilíbrio térmico com o forno. Este equilíbrio é verificado monitorando a posição de uma autofreqüência, conforme mostrado na *Figura 3.5(b)*.

Figura 3.4: Imagem do forno de alumínio mostrando os três transdutores e resistores de potência em amarelo.

Figura 3.5 (a) Variação da temperatura com um forno e 4 aquecedores. (b) Posição do pico monitorado. Nota-se uma variação máxima de 14Hz.

Podemos observar que, com um dia e meio de monitoramento, houve uma variação de 14Hz na posição do pico, o que é uma variação acima da esperada. Mesmo após alguns dias não houve o equilíbrio térmico da amostra com o forno. Depois de efetuarmos outras séries de medidas, em condições ambientais diversas, pudemos perceber que a amplitude de variação do pico sofria influência de variáveis ambientais, como temperatura e insolação. Passamos então a trabalhar em busca de um melhor isolamento térmico do forno.

Para evitar a irradiação direta de calor sobre o forno, revestimos a campânula de vidro com folhas de alumínio, utilizadas para isolamento de telhado, e adicionamos uma placa de teflon sobre a base de aço inox da mesa. As janelas de vidro do laboratório foram revestidas com folhas de alumínio para diminuir a amplitude térmica do laboratório. Além disso, foram instalados três exaustores na sala.

Aproveitamos esta fase para também fazer modificações no forno, aumentando para 16 do número de aquecedores. Assim, a potência total dissipada tem melhor distribuição espacial e é diminuída em cada aquecedor. O monitoramento do pico é apresentado na Figura 3.6(a) e, como podemos ver, a variação máxima caiu para 6 Hz. Entretanto, ainda persistia uma componente periódica relacionada às variações diárias de temperatura.

Figura 3.6: (a) Variação da posição de um pico de monitoramento de um forno com 16 aquecedores. Variação máxima = 6Hz (b) Variação do pico com 2 fornos. Variação máxima = 0.8Hz.

A solução definitiva veio quando colocamos este forno dentro de outro forno, que blindou adequadamente a troca de calor com o ambiente. A temperatura foi controlada nas mesmas condições do forno de alumínio, por um segundo controlador (Neocera, LTC-21).

O segundo forno é feito de cobre em formato cilíndrico, com diâmetro de 20.5cm, 30cm de altura e parede com espessura de 4mm. A opção pelo cobre deve-se à alta condutibilidade térmica deste material, o que diminui a duração de gradientes de temperatura. Para o aquecimento do forno foram utilizados 32 resistores de potência de 10Ω . A foto deste forno é apresentada na Figura 3.7.

O resultado dos testes da nova configuração é apresentado na Figura 3.6(b), de onde se pode notar que conseguimos uma variação máxima de 0,8Hz, mostrando que a grande capacidade térmica deste forno consegue blindar as variações de temperatura ambiental.

Figura 3.7: Forno externo de cobre.

3.1.2 Sistemas de excitação e detecção

Os dispositivos transdutores são caracterizados pela propriedade de conversão entre diferentes tipos de energia. Em nosso caso, utilizamos transdutores piezoelétricos tanto para transformar os sinais elétricos enviados pelo analisador de espectros em energia mecânica quanto para converter novamente a energia de vibração da amostra em energia elétrica.

O piezelétrico utilizado para construção dos transdutores é o VP-A50 da Valpey-Fisher, escolhido por sua alta sensibilidade às alterações mecânicas e elétricas e por ter boa resposta na faixa de frequência desejada. Este piezelétrico tem o formato de disco com dimensões de 3mm de diâmetro e 0,5mm de altura.

A face inferior do piezelétrico é colada com cola condutora elétrica sobre um cilindro de ouro para aumentar a inércia do sistema. Neste cilindro é feito um furo no qual é conectado um dos terminais elétricos. Sobre o piezoelétrico é colada uma esfera de rubi na qual é apoiada a amostra. A utilização de uma esfera de pequeno diâmetro (1mm) para apoio da amostra se dá para que o contato entre apoio e amostra seja

pequeno. A opção pelo rubi deve-se à alta dureza do material, evitando perda de energia transmitida entre transdutor e amostra. Na face superior do piezoelétrico é conectado o outro terminal elétrico. Apresentamos na Figura 3.8 o desenho esquemático (a) e a foto (b) de um transdutor construído no laboratório.

A fim de reduzir as interferências eletromagnéticas, todo o conjunto acima descrito é blindado com uma casca cilíndrica de latão. O isolamento elétrico entre a parte interior do transdutor e a casca é feito através de uma casca cilíndrica de teflon. Os transdutores são ainda envolvidos em outra capa de teflon para que haja isolamento elétrico e mecânico.

Figura 3.8 (a) Desenho esquemático e (b) foto do transdutor piezoelétrico. Estes dispositivos foram construídos para excitar a amostra (emissor) e medir sua amplitude de vibração (receptor).

3.1.3 O analisador de espectros

Utilizamos um analisador de espectros da Agilent, modelo 4395A, que trabalha na faixa de 10Hz-500MHz com resolução de 1mHz e acurácia de 0,13 ppm. As medidas foram efetuadas com o aparelho no modo analisador de rede (*Network analyser*). Neste modo, o equipamento envia ao canal de saída um sinal senoidal com amplitude fixa e frequência que aumenta linearmente com o tempo. Simultaneamente, o sinal recebido no canal de entrada é analisado, e o que se mede é a amplitude do sinal na frequência emitida. As medidas são apresentadas em tempo real na tela do analisador como um gráfico da amplitude em função da frequência. A Figura 3.9 mostra uma foto do painel do analisador, na qual se podem ver os resultados obtidos depois de uma varredura.

Figura 3.9: Foto do painel do analisador de espectros. As medidas efetuadas são apresentadas em tempo real na tela.

Para garantir a qualidade dos dados adquiridos, é fundamental que o analisador de espectros esteja corretamente configurado. Os parâmetros mais importantes para que seja feita uma boa aquisição são a Potência do sinal emitido (*PowerDb*) e a largura do filtro do receptor (*IF Bandwidth*). A configuração destes parâmetros será explicada em detalhes no Capítulo 4.

3.2 Confeção das amostras

As amostras utilizadas nos experimentos são construídas por uma

fresa 3D controlada por um computador tipo PC. O desenho das amostras é feito no programa Autocad® e depois exportado para o programa VirtualMill®, que é responsável por controlar a fresa durante a confecção.

As amostras estudadas foram confeccionadas em alumínio devido à facilidade de usinagem deste material e ao seu alto fator de qualidade (Q). Esta medida está relacionada com a capacidade de se distinguir dois picos próximos e será explicada em detalhes adiante.

O primeiro passo para a confecção das amostras consiste em desbastar uma chapa, para que fique com a espessura desejada e com as duas faces paralelas. Depois disso, suas bordas são desbastadas gradativamente, até que a amostra se separe da chapa.

As amostras devem ser tratadas termicamente a 300°C para eliminar as tensões internas. A quebra destas tensões internas facilita a estabilização da temperatura da amostra, e diminui de 15 para 2 dias o tempo necessário para o que o espectro atinja o equilíbrio térmico.

Fator de qualidade (Q)

A capacidade de se resolver dois picos de ressonância vizinhos depende da distância entre eles e de suas larguras. Em geral, dizemos que os picos estão resolvidos se a distância (s) entre eles é maior do que suas larguras à meia altura (Δf).

O fator de qualidade é definido como

$$Q = \frac{f}{\Delta f} \quad (3.1)$$

e serve como uma medida da facilidade em se distinguir dois picos a determinada frequência f . Desta maneira, ao desenvolvermos um arranjo para o estudo de ressonadores, sempre procuramos aumentar o valor de Q .

Dentre os diversos fatores que podem alterar o valor de Q , o aumento da pressão sobre a amostra faz com que a oscilação seja amortecida, diminuindo assim o fator de qualidade. Neste sentido, o sistema de vácuo é eficiente para o aumento de Q , como pode ser visto na Figura 3.10, em que apresentamos o formato de um pico adquirido em vácuo e em pressão ambiente. No primeiro caso, obtivemos $Q = 1 \times 10^4$ e, no segundo, $Q = 6 \times 10^3$.

Figura 3.10: Formato de um pico adquirido (a) em vácuo (10^{-2} Torr) e (b) em pressão ambiente.

Quanto maior a massa da amostra, maior será o valor de Q , entretanto, veremos adiante que existem limitações quanto às suas dimensões, além das impostas pelo tamanho do forno.

O terceiro aspecto de grande influência no fator de qualidade é o material que constitui o ressonador. K. Schaadt comparou os fatores de qualidade de diferentes materiais, concluindo que os melhores materiais a serem utilizados são o quartzo ($Q \approx 10^5$), o silício monocristalino ($Q \approx 10^6$) e o alumínio ($Q \approx 10^4$). Levando em consideração que tanto o quartzo quanto o silício são de difícil usinagem optamos por utilizar o alumínio para garantir precisão na confecção das amostras.

Dimensões da amostra

Para se obter o maior número de ressonâncias em determinada faixa de frequências e com alto valor de Q , procuramos fazer as amostras no maior tamanho possível, mas com a condição de que sejam facilmente manipuladas no interior do forno. Outro fator que se deve levar em conta

na escolha é a espessura (h), que delimitará a frequência de corte f_c na qual o ressonador pode ser considerado uma placa. A espessura da amostra deve tal que

$$h < \frac{V}{2f_c}, \quad (3.2)$$

onde V é a velocidade do som na amostra.

Removendo simetrias

Como mencionado na seção 2.2.4, a presença de simetrias triviais em bilhares geram degenerescências triviais no sistema. Em sistemas ressonadores, a presença destas simetrias tende a gerar sobreposições de espectros independentes que dificultam a análise e compreensão dos resultados obtidos. Devemos então sempre construir amostras que não tenham simetrias. A Figura 3.11 mostra um bilhar circular com e sem simetrias.

Figura 3.11: Bilhar circular com e sem simetrias..

Dessimetrização planar

Apesar de estarmos trabalhando em uma faixa de frequência em que os ressonadores podem ser considerados planares, devemos lembrar que estes sistemas são tridimensionais e ainda apresentam uma simetria em relação ao plano da espessura. A dessimetrização planar¹ é um

¹ É importante ressaltar que o termo usualmente utilizado para designar este procedimento é apenas de *dessimetrização*. Acreditamos, entretanto que esta utilização pode ter sentido dubio, podendo ser confundida com a retirada das outras simetrias. Desta maneira, optamos pela utilização do termo *dessimetrização planar*.

procedimento que visa retirar esta simetria através de cortes na amostra com altura de aproximadamente metade da espessura. Um exemplo de um estádio de Sinai, com e sem a dessimetrização planar é apresentado na Figura 3.12.

Figura 3.12: Desenho esquemático de dois estádios de Sinai, em que se pode ver o efeito do processo de dessimetrização planar.

Com a remoção desta simetria, espera-se que o sistema não apresente mais sobreposição de espectros independentes. Nós, particularmente, preferimos a interpretação física de que a dessimetrização planar acopla as ondas flexurais com as simétricas. Como vimos na seção 2.4.2, estas ondas se propagam de maneira independente em uma placa de espessura constante.

3.2.2 Programas de monitoramento e de aquisição de dados

Considerando que todos os equipamentos utilizados no experimento possuem interface que permite a comunicação com um computador, desenvolvemos um programa de monitoramento de dados para estabelecer as condições de início da aquisição de uma maneira simples.

O programa de monitoramento se comunica com os dois controladores de temperatura e com o analisador de espectro através de uma interface GPIB, largamente utilizada para controle de equipamentos.

Com a configuração adequada, todos os comandos acessíveis pelo painel destes instrumentos também podem ser acessados via computador. A comunicação com o pré-amplificador é feita via interface RS-232 de maneira unidirecional, sendo possível apenas o envio de comandos ao aparelho. Esta limitação, entretanto, não gerou nenhum problema que comprometesse a qualidade dos programas.

O programa de aquisição de dados foi construído de maneira independente do programa de monitoramento, mas utilizamos alguns elementos que permitiram o acompanhamento das variáveis relacionadas à estabilidade espectral durante a aquisição dos dados. O monitoramento

durante a aquisição é um procedimento original no estudo de ressonadores acústicos, que julgamos ser muito importante para que se ateste a qualidade dos dados.

Os programas foram desenvolvidos no ambiente LabView® por este permitir que a configuração inicial da comunicação com os equipamentos seja relativamente simples. Neste ambiente, também é possível se configurar uma interface amigável com o usuário, permitindo inclusive a construção de gráficos em tempo real. Unindo-se a isto a possibilidade de exportação da tela do programa via internet, os programas nos permitem que, de maneira simples, sejam feitos monitoramentos e configurações a qualquer momento, dispensando a exigência de presença física no laboratório para efetuá-los.

Programa de monitoramento da temperatura e estabilização do pico de referência (pré-aquisição)

Com este programa, cuja tela é apresentada na Figura 3.13, podemos acompanhar em tempo real os gráficos da temperatura dos dois fornos, da potência emitida por cada um dos controladores de temperatura e da posição do pico monitorado.

Os valores de temperatura e potência podem ser lidos diretamente dos controles de temperatura. Entretanto, para a obtenção da posição do pico é necessário que o analisador faça uma varredura completa em torno do pico selecionado para que seja feita a detecção automática de sua posição. Desta maneira, devemos ter o cuidado de garantir que o intervalo entre medidas consecutivas seja maior do que o tempo de uma varredura.

Entre duas leituras, o programa permite que as escalas sejam alteradas para que os gráficos sejam analisados em maior detalhe; mas após cada atualização o gráfico é reescalado para que todos os pontos apareçam na tela. A possibilidade de manipulação dos gráficos em tempo real permite que se defina com facilidade a condição de equilíbrio para o início da aquisição.

Quando iniciado, o programa consulta os parâmetros de todos os equipamentos e atualiza os valores na tela. A alteração dos parâmetros é feita diretamente nos campos em que aparecem seus valores atuais. Quando um campo é alterado, o programa envia um comando de modificação seguido de um comando de leitura da grandeza, a fim de verificar se o parâmetro foi modificado com êxito. Esta verificação é extremamente importante, já que a interface GPIB é passível de falhas em

algumas circunstâncias.

Figura 3.13: Tela do programa de monitoramento de temperatura dos fornos e do pico de referência.

O passo usual a ser tomado depois de iniciarmos o programa é definir um intervalo de frequência no analisador de espectros que contenha o pico a ser monitorado. Depois disso, os controladores de temperatura são colocados no modo ativo e passam a atuar de maneira a garantir que a temperatura estabilize no valor programado. Lembrando que mudanças de temperatura levam à movimentação dos picos de ressonância, o intervalo de frequência é automaticamente atualizado a cada varredura para que o pico permaneça no centro no intervalo monitorado pelo analisador. O diagrama de sequência do programa de monitoramento é apresentado na Figura 3.14.

Figura 3.14: Diagrama de sequência do programa de monitoramento.

Programa de Aquisição de dados

O programa de aquisição é iniciado depois de constatada a estabilidade térmica espectral com o programa de monitoramento. A tela inicial deste programa, apresentada na Figura 3.15, permite que uma série de parâmetros seja modificada antes de se iniciar a coleta de dados. Nessa etapa, geralmente são feitos os ajustes finais no intervalo de frequência de monitoramento do pico e são configurados os parâmetros do analisador relacionados à qualidade dos dados e ao tempo de aquisição.

Antes de a aquisição ser iniciada, devemos escolher um caminho para salvar os dados, as frequências inicial (f_{IniAq}) e final (f_{FinAq}) de aquisição e o “Span” (Δf_{Varr}), intervalo de frequência de uma varredura. Este último valor, em conjunto com o número de pontos por varredura ($\# points$), definirá a resolução do traço. Outra opção a ser definida é a aquisição de um ou dois traços ($\# Canais$).

Figura 3.15: Tela inicial do programa de aquisição, utilizada para os ajustes dos parâmetros da aquisição de dados.

Ao pressionarmos o botão “ADQUIRIR”, o programa faz uma verificação para garantir que todos os parâmetros apresentados na tela coincidam com aqueles dos equipamentos, e apresenta uma nova tela ao usuário. Nessa tela, conforme a Figura 3.16, é possível monitorar da posição do pico e das temperaturas dos fornos, o que permite constatar a

presença de anomalias durante a aquisição. Além disso, é possível visualizar o gráfico com os dados obtidos na última varredura e, com isto, detectar uma eventual falha no funcionamento dos transdutores. Uma barra horizontal no inferior da tela indica a fração de dados adquiridos até o momento. O diagrama de sequência da aquisição de dados é apresentado na Figura 3.17.

Figura 3.16: Tela de monitoramento da aquisição de dados.

Capítulo 4

Aquisição e tratamento dos dados

Neste capítulo discutiremos os cuidados experimentais que devem ser tomados para que tenhamos espectros de boa qualidade. Esses cuidados vão desde a aquisição de dados, que será explicada na seção 4.1, até o tratamento de dados, explicado na seção 4.2, que consiste na obtenção de um espectro desdobrado a partir dos traços medidos.

4.1 Aquisição de dados

A faixa típica de frequência utilizada na medida de um traço é 0-1MHz, valores estes que dependem intrinsecamente das propriedades físicas e das dimensões das amostras. Para a obtenção de um traço completo devemos variar a frequência em intervalos Δf , chamados de varredura, e para cada varredura devemos determinar o número N de pontos a serem obtidos, 801 pontos no máximo. Os traços normalmente são adquiridos com resolução de 1Hz, sendo $\Delta f=800\text{Hz}$ e $N=801$ ¹ os valores mais utilizados.

A taxa de aquisição e o tempo de varredura são determinados pela largura do filtro passa faixa do analisador (*IF Bandwidth*), que determina a capacidade de resolver sinais. Quanto menor o valor de *IF Bandwidth*, maior a capacidade de resolução, porém, maior será o tempo necessário para que se faça uma medida. A *Tabela 4.1* mostra o tempo necessário para uma varredura de 801 pontos, na qual foram utilizados os menores valores de *IF Bandwidth* que o analisador pode aceitar. Também é possível observar a duração estimada da aquisição de um traço com resolução de 1 Hz e intervalo de medida de 1MHz. Podemos notar que utilização de valores muito baixos pode tornar muito longa a aquisição de dados, o que não é desejado principalmente devido à estabilidade térmica

¹ Devemos notar que o programa de aquisição exclui o último dado lido de cada traço.

do espectro.

<i>IF Bandwidth (Hz)</i>	<i>Tempo de varredura (s)</i>	<i>Tempo de aquisição (horas)</i>
2	750	260
10	188	65
30	47	16
100	23	8

Tabela 4.1: Tempo necessário para que se complete uma varredura de 801 pontos. Os valores de "IF Bandwidth" são os menores que o analisador pode aceitar. Para os tempos de aquisição consideramos um traço com resolução de 1Hz, no intervalo de 1MHz.

Para obtermos um espectro completo, geralmente são necessárias aquisições de diversos traços. Isto porque pode ocorrer de um emissor ou receptor estarem posicionados em um nó da onda, fazendo com que o modo não seja excitado ou medido. A Figura 4.1 mostra uma região de três traços obtidos experimentalmente, de onde se pode notar que o primeiro nível não seria detectado se apenas o traço (c) fosse medido.

Figura 4.1: Detalhe de três traços obtidos, nos quais foram variadas as posições do emissor e receptor. Pode-se observar que a aquisição de apenas um traço pode levar à perda de níveis.

Levando em consideração que o tempo de aquisição pode variar significativamente com a escolha de *IF Bandwidth*, geralmente adquirimos alguns traços com valores menores dos parâmetros, e outros com valores maiores, apenas para verificação de que não há perda de níveis.

A potência emitida pelo analisador pode variar na faixa de -50 dBm a +15dBm e, ao contrário do que se possa imaginar, nem sempre a maior potência é a melhor para excitar a amostra. Como mostrado na Figura 4.2, a utilização de uma potência muito alta pode fazer com que as medidas de alguns picos de ressonância apresentem uma quebra, principalmente para as amostras de menor massa, impossibilitando a correta determinação de sua posição. Este fenômeno foi relatado anteriormente por K. Schaadt [3], que o atribuiu a uma trepidação da amostra sobre os apoios quando submetida a vibrações de maior intensidade.

Para ajustar o valor da potência utilizado nas aquisições, fazemos algumas varreduras rápidas sobre toda a faixa de frequência em que faremos medidas, apenas com o intuito de identificarmos os picos mais altos. Estes picos funcionarão como referência para que escolhamos o valor da potência, sempre deixando certa margem de segurança, mas tomando o cuidado de evitar que um sinal muito baixo torne os dados ruidosos.

Figura 4.2: Detalhe da medida de um pico de ressonância para dois

valores de amplitude de excitação. Devido à pequena espessura da placa, há uma falha quando a potência de emissão é alta, provavelmente ocasionada pela trepidação da amostra.

A estabilização térmica do espectro pode durar muitos dias e, com isso, estender muito o tempo de obtenção de todos os traços necessários. Neste sentido, a utilização de 2 receptores para a medida de espectros simultâneos faz com que tenhamos que abrir o forno menos vezes e diminui o tempo global de aquisição.

Um traço típico adquirido é mostrado na Figura 4.3.

Figura 4.3: Curvas em vermelho: traço entre 0.4 kHz a 400kHz. Curvas

em preto: o mesmo traço com escalas ampliadas para melhor visualização de picos de baixa intensidade

4.2 Tratamento dos dados

Depois de medidos os traços, fazemos o tratamento de dados para obtermos os espectros de ressonância. Os picos de ressonância devem ser identificados um a um para que seja feito um ajuste numérico da curva de ressonância em torno de cada um deles. Posteriormente, efetuamos o desdobramento do espectro, processo no qual são removidas todas as características dependentes das particularidades do sistema, o que é fundamental para uma análise baseada na TMA.

A dificuldade em se observar diretamente se determinada ressonância é proveniente de uma órbita periódica faz com que sejam desenvolvidas ferramentas que buscam removê-las a partir de seus efeitos no espectro. Um método bastante utilizado para o caso de ressonadores acústicos é o método de corte (*cut-off*), baseado na transformada de Fourier do espectro.

A seguir, apresentaremos em detalhes os passos necessários para que, a partir dos traços medidos, obtenhamos o espectro desdobrado, que será utilizado nas análises e obtenção de resultados.

4.2.1 Detecção inicial dos picos de ressonância

O processo de obtenção do espectro de ressonância a partir dos traços medidos é dividido em duas etapas para que possamos, em menor tempo, obter resultados menos suscetíveis a erros. A primeira etapa consiste na construção de uma lista com a posição aproximada de todos os picos detectados nos traços. Posteriormente são feitos os ajustes numéricos em torno desses pontos para garantir a correta determinação de suas posições.

Para construirmos a lista inicial da posição dos picos, desenvolvemos um programa em C++, cuja listagem é apresentada no Apêndice A-I, no qual os traços são desenhados na tela (*amplitude x frequência*) para que os picos de ressonância sejam identificados. Na tela de configuração, apresentada na Figura 4.4, é possível carregar os traços e configurar parâmetros como frequência inicial e resolução das medidas. Outra característica importante é a possibilidade do carregamento de uma lista preexistente.

Mostra	Nome	Fator Deslocamento	Freq. Inicial	Resolucao	Num Pontos	Carregar
0	Arquivo de Dados E Analises\Circulo Desimetrizado\Spec1.dat	0	1000	1.0	1000	☒
1	Arquivo de Dados E Analises\Circulo Desimetrizado\Spec2.dat	0	1000	1.0	1000	☒
2	Arquivo de Dados E Analises\Circulo Desimetrizado\Spec3.dat	0	1000	1.0	1000	☒
3	Arquivo de Dados E Analises\Circulo Desimetrizado\Spec4.dat	0	1000	1.0	1000	☒
4		0	1000	1.0		☐
5		0	1000	1.0		☐
6		0	1000	1.0		☐

☐ Arquivo de Referencia ☒ Valores Unicos 1000 1.0

PICOS

Mostra	Nome	Num. Pontos	Carregar	Hz/kHz
☐		1000	☐	Hz
☐		1000	☐	Hz

☐ PloteTipoPontos

Figura 4.4: Tela de configuração do programa utilizado para determinação aproximada das posições dos picos.

Na tela principal do programa, apresentada na Figura 4.5, podemos visualizar os traços e as posições dos picos da lista (linhas verticais vermelhas). O intervalo do espectro que será mostrado na tela é configurado através das variáveis Início (frequência inicial) e Intervalo.

Para se adicionar um ponto à lista, devemos clicar com o botão esquerdo do mouse na posição desejada, no interior do retângulo preto. O ponto adicionado será instantaneamente apresentado na tela. Ao se clicar com o botão direito, é removido o ponto da lista mais próximo da posição do clique.

Nos primeiros traços analisados, a pré-deteção foi feita de maneira totalmente manual, tornando o processo demorado e sujeito a falhas, já que a distinção visual dos picos pode ser difícil em alguns casos. A proximidade de picos com amplitudes muito diferentes, como mostrado na Figura 4.6, é a situação mais comum para que este problema ocorra.

Figura 4.5: Tela principal do programa utilizado para determinação aproximada das posições dos picos.

Figura 4.6: Situação típica na qual a diferença de amplitude pode comprometer a detecção de um pico.

Com a finalidade de automatizar a tarefa de pré-deteccção dos picos, escrevemos um programa em MatLab, cujo funcionamento é descrito a seguir:

1. É feita uma suavização do traço para evitar que flutuações causadas por ruídos sejam interpretadas como picos. Para isso, o valor de cada ponto é recalculado através de uma média ponderada (gaussiana) considerando-se o intervalo até as segundas vizinhanças.

2. Calcula-se a primeira derivada do traço suavizado e aplica-se o processo de suavização a este resultado. Calcula-se então a segunda derivada do traço.

3. Para encontrar os valores aproximados das autofreqüências devemos localizar os mínimos (vales) da série de dados da segunda derivada. Um parâmetro de corte é utilizado para definir o limiar em que os vales serão considerados. Podemos notar que a qualidade da detecccção é extremamente sensível a esta variável. No caso da escolha de um limiar muito baixo, muitos picos espúrios são identificados, enquanto a escolha de um alto limiar poderá causar a perda excessiva de picos.

Nos testes efetuados com este programa, concluímos que um ajuste adequado dos parâmetros permite detectar automaticamente cerca de 90% dos picos, com uma quantidade muito pequena de falsas detecccões.

4.2.2 Ajuste dos picos e obtenção do espectro

O espectro de ressonância é obtido através do ajuste numérico em torno dos picos detectados na fase inicial de análise. Para isto, utilizamos o método do gradiente [26] para ajustar a curva "aproximação da lorentziana assimétrica", utilizada originalmente para ajustar picos de cavidades de microondas [27], mas que se mostrou eficiente para o caso dos ressonadores acústicos [25]. Ele consiste em uma aproximação da fórmula de Breit-Wigner [28, 29], na qual a análise é feita em uma região com um pequeno número de picos e a contribuição externa é aproximada por um polinômio. No caso de um polinômio de 1º grau, a expressão da sobreposição de n lorentzianas assimétricas será dada pela expressão:

$$A(f) = \sum_{n=1}^N \frac{C_n + D_n(f - f_0)}{(f - f_n)^2 + \frac{1}{4}\Gamma_n^2} + B_0 + B_1(f - f_0), \quad (4.1)$$

onde f_n é a freqüência de ressonância. A é a amplitude da curva, N o número de picos e f_0 é o centro do intervalo. Os parâmetros C_n , D_n e Γ_n

estão relacionados, respectivamente, com a amplitude, assimetria e largura do n -ésimo pico e os termos B_0 e B_1 são os coeficientes do polinômio.

Antes de iniciarmos os ajustes, devemos escolher o traço que servirá como base para a obtenção do espectro, o qual chamamos de **traço principal**. A escolha é geralmente feita na fase da detecção inicial, com base na qualidade do sinal, no parâmetro de aquisição *If-Bandwidth*, e na resolução das medidas. Os outros traços, chamados de **traços auxiliares**, são utilizados para a determinação dos pontos onde não é possível fazer um ajuste adequado com o traço principal.

Os ajustes dos picos de ressonância foram feitos em um programa escrito em Matlab e a listagem é apresentada no Apêndice A-II. Na primeira etapa, a análise é feita a partir da lista construída na fase inicial e no traço principal. Inicialmente, o primeiro pico é apresentado na tela e, através de cliques na figura, é possível selecionar a região a ser ajustada. Feito o ajuste, o programa apresenta o gráfico com a curva ajustada e os dados experimentais, conforme mostrado na Figura 4.7, para fazermos uma verificação visual da qualidade do ajuste. Se o ajuste for satisfatório, os parâmetros do ajuste serão gravados em um arquivo e o próximo pico da lista é mostrado na tela. Caso contrário, podemos optar por reiniciar o processo e escolher outro intervalo para o ajuste, ou gravar um indicativo no arquivo de parâmetros de que o pico deve ser ajustado com outro traço.

A segunda fase de obtenção do espectro é feita com a lista de parâmetros ajustados e com os traços auxiliares. O processo descrito anteriormente é repetido para os picos que não foram bem ajustados, de maneira a permitir a substituição dos parâmetros no caso de melhora.

Figura 4.7: Ajustes obtidos em três situações diferentes. Podemos notar que em (a) o ajuste é adequado. Em (b), a baixa amplitude do sinal faz com que se tenha um número menor de pontos para o ajuste. Esta é uma situação na qual é feita uma nova tentativa de ajuste, utilizando um traço auxiliar. A proximidade de picos(c) pode fazer com que seja mais eficiente o ajuste de mais de um pico simultaneamente.

4.2.3 Desdobramento do espectro

Para que possamos comparar os resultados obtidos experimentalmente com as previsões da TMA, devemos "normalizar" o espectro de maneira que nele permaneçam apenas as características universais. Este processo é conhecido como desdobramento do espectro e para efetuá-lo é conveniente definirmos a **função escada**:

$$n(f) = \sum_{f_i \leq f} 1. \quad (4.2)$$

Dado um espectro $\{f_1, f_2, \dots, f_i, \dots\}$, esta função conta o número de níveis do espectro com valores menor ou igual a f .

A função escada pode ser reescrita como a soma de dois termos: um termo médio $n_{\text{médio}}(f)$, que é a integral sobre a densidade média de níveis, e um termo da flutuação $n_{\text{flut}}(f)$:

$$n(f) = n_{\text{médio}}(f) + n_{\text{flut}}(f). \quad (4.3)$$

O termo $n_{\text{médio}}(f)$ carrega todas as informações dependentes do sistema, enquanto n_{flut} trás as características universais do espectro. Para remover as contribuições dependentes do sistema devemos efetuar uma transformação $f_i \rightarrow \nu_i$ tal que $n_{\text{médio}}(f) = \nu$ e assim:

$$n(\nu) = \nu + n_{\text{flut}}(\nu). \quad (4.4)$$

Com isto, obtemos um espectro com densidade média igual a um, cujas propriedades são determinadas apenas pelas flutuações do espectro original em torno de seu comportamento médio.

Desdobrando os espectros experimentais

Para desdobrarmos os espectros experimentais é necessário que saibamos estimar adequadamente a parte média da função escada. Na seção 2.4.2 foi apresentado um método que busca esta estimativa a partir da lei de Weyl e de um modelo elastodinâmico de ressonadores acústicos planares. Estes resultados, entretanto, não se mostraram adequados para o desdobramento dos espectros. Acreditamos que esta discrepância está relacionada ao fato de que os coeficientes apresentados na Tabela 2.2 são muito dependentes das características particulares do material utilizado na ocasião. Tais discrepâncias já haviam sido observadas anteriormente e a

solução encontrada foi a utilização de um polinômio de 3º grau [30].

Na Figura 4.8, apresentamos a função escada de um espectro experimental e a curva ajustada. Podemos ver no detalhe as flutuações do espectro em torno do comportamento médio.

Figura 4.8: Curva ajustada à função escada de um espectro experimental. No detalhe é possível observar as flutuações em torno do comportamento médio.

Depois de efetuado o ajuste, chegamos à expressão

$$n_{\text{médio}}(f) = a_0 + a_1 f + a_2 f^2 + a_3 f^3 \quad (4.5)$$

para a parte média da função escada. O nível v_i do espectro desdobrado $\{v_1, v_2, \dots, v_i, \dots, v_N\}$ será dado por:

$$v_i = a_0 + a_1 f_i + a_2 f_i^2 + a_3 f_i^3, \quad (4.6)$$

Vale lembrar que os coeficientes a 's possuem dimensão inversa da grandeza que eles multiplicam e, assim, o espectro desdobrado é adimensional.

4.2.4 O método de corte (*cut-off*)

Durante o tratamento de dados, devemos considerar a presença das órbitas periódicas do tipo "*bouncing balls*", pois elas causarão desvios nas estatísticas dos espectros. Em um estudo sobre vibrações flexurais de placas, E. Bogolmony e E. Hugues [31] derivaram uma fórmula aproximada para o número dessas órbitas em um estádio:

$$N^{bb}(k) = \frac{1}{(2\pi)^{3/2}} \frac{a}{r} \sqrt{2kr} \sum_{m=1}^{\infty} m^{3/2} \cos\left(2mkr - \frac{3\pi}{4}\right) \quad (4.7)$$

onde k é o número de onda, r é o raio do estádio e a é a sua largura. Estes resultados foram verificados experimentalmente por H. D. Gräf [32] em uma cavidade de microondas, como pode ser visto na Figura 4.9.

Figura 4.9: Flutuações do espectro de uma cavidade de microondas no formato de um estádio de Bunimovich [32]. As previsões para N^{bb} são representadas pela linha mais fina.

As flutuações de espectros experimentais de ressonadores acústicos se mostraram diferentes das previsões [25, 23, 3], de maneira que é necessário desenvolver outro método.

Uma das soluções possíveis seria a de se fazer uma varredura das amplitudes de oscilação ao longo da amostra nas frequências de ressonância e, a partir da análise dos padrões espaciais, concluir se a órbita é periódica [23]. Entretanto, a utilização deste método para espectros inteiros torna-se inviável devido à longa duração de uma

varredura na amostra.

Observando novamente a Figura 4.9, podemos notar que o efeito das órbitas "bouncing balls" nas flutuações é a de adicionar uma componente de baixa frequência às suas flutuações espectrais. Com base nesta conclusão, utilizamos o método de corte, ou *cut-off*, que consiste em filtrar as componentes de baixa frequência das flutuações do espectro para depois reconstruir o espectro. Este método foi originalmente utilizado por K. Schaadt [3].

Inicialmente, aplicamos o algoritmo FFT [33, 34] (*Fast Fourier Transform*) à série das flutuações e construímos o gráfico da potência, que é apresentado na Figura 4.10. As flutuações a serem removidas aparecerão como picos na região de x pequeno. Desta maneira, estabelecemos um valor de corte (x_c), acima do qual todos os valores do FFT são iguais a zero, e calculamos o FFT inverso (iFFT). Como resultado, obtemos as flutuações que serão removidas da série original, conforme a Figura 4.11(b). As flutuações do novo espectro, mostradas na Figura 4.11(c), são obtidas subtraindo este resultado das flutuações originais. Este programa foi escrito no MatLab e a listagem é apresentada no Apêndice A-III.

Figura 4.10: Gráfico de potência do FFT das flutuações do espectro de um ressonador acústico

Figura 4.11: (a) Flutuações do espectro experimental de um ressonador acústico; (b) flutuações removidas e (c) flutuações obtidas após a aplicação do método de corte.

Capítulo 5

Resultados

Efetuamos um estudo paramétrico com espectros de ressonadores acústicos obtidos no laboratório do grupo CATS, por C. Ellegaard e K. Schaadt. Os resultados deste trabalho serão apresentadas na seção 5.1.

Os primeiros resultados obtidos com nosso arranjo experimental foram para o estádio de Sinai, com e sem dessimetrização planar, com o qual pudemos verificar a qualidade de nossa montagem experimental. Esses resultados serão apresentados na seção 5.2.

Adaptamos um modelo da termodinâmica de espectros de *ensembles* circulares, desenvolvido por F. J. Dyson e M. L. Metha, para o caso de espectros gaussianos finitos, de onde pudemos derivar expressões para o cálculo da energia e do calor específico. Apresentamos este estudo na seção 5.3 e, nas subseções posteriores, aplicamos nosso modelo em simulações numéricas e nos espectros experimentais dos estádios de Sinai.

Estudamos três ressonadores triangulares, cujos bilhares análogos pertencem a diferentes classes em relação à integrabilidade. Os resultados serão mostrados na seção 5.4.

Em um estudo de sistemas circulares, apresentado na seção 5.5, fizemos análises do espectro formado pelos zeros da função de Bessel (J_m), que são as soluções de um bilhar circular, e podem ser considerados como uma primeira aproximação de ressonadores circulares. Estudamos também os espectros gerados após a separação entre m =ímpar e m =par, lembrando que o segundo caso equivale ao espectro de um bilhar com formato de 1/4 de círculo. Comparamos estes resultados com os espectros experimentais de ressonadores de mesma geometria, com e sem dessimetrização planar.

5.1 Curvatura de níveis

Em um estudo sobre dinâmica de níveis, M. Wilkinson [35] construiu um modelo baseado na TMA para investigar as propriedades estatísticas

da velocidade e curvatura (primeira e segunda derivada) de um espectro GOE, na medida em que um parâmetro é variado. Foi encontrado que as velocidades devem seguir uma distribuição gaussiana e que a distribuição das curvaturas de níveis (k) será:

$$P(k) = \frac{1}{2(1+k^2)^{3/2}}, \quad (5.1)$$

onde as curvaturas foram renormalizadas de maneira que a média de seu módulo seja igual a um, ou seja, $\langle |k| \rangle = 1$.

Este resultado foi posteriormente provado e generalizado por vonOppen [36] para todos os *ensembles* gaussianos, onde se mostrou que a curvatura pode ser escrita como:

$$P(k) = \frac{1}{2(1+k^2)^{(\beta+2)/2}}, \quad (5.2)$$

com $\beta=1$ (GOE), $\beta=2$ (GUE) ou $\beta=4$ (GSE).

Em um estudo efetuado com ressonadores acústicos, em que a temperatura (T) foi utilizada como parâmetro, P. Bertelsen *et al.* [37] chegaram a distribuições de curvaturas diferentes daquelas previstas pela teoria, encontrando valores mais baixos para valores de k pequeno. Em trabalho recente, M. Hussein *et al.* [38] propuseram que esta divergência pode estar ligada à falta de uma normalização adequada das curvaturas.

Num trabalho em colaboração com o grupo CATS, de Copenhague, utilizamos as distribuições de curvaturas de diferentes ressonadores acústicos que obedecem às estatísticas do tipo GOE [25], aos quais ajustamos à equação normalizada:

$$P(k) = \frac{\gamma^2}{2(\gamma^2 + k^2)^{3/2}} \quad (5.3)$$

com $\gamma = \langle |k| \rangle$ sendo extraído dos dados experimentais. Na Figura 5.1, apresentamos os resultados desses ajustes, onde, do lado esquerdo temos as curvaturas não reescaladas e, na coluna direita, as curvaturas reescaladas.

Figura 5.1: Distribuição de curvaturas (col. esquerda) sem e (col. direita) com a aplicação do método de reescalonamento. A aplicação do método remove a discrepância encontrada nas regiões de alta curvatura. (a) Estádio de Sinai (alumínio), (b) modos flexurais do estádio de Sinai, (c) bloco de Quartzo com retirada de massa, (d) bloco de Quartzo com variação de temperatura, (e) bilhar de Robnick (P. Leboeuf e M. Sieber [39]).

Os valores de γ e do χ^2 , apresentamos na Tabela 5.1, corroboram a qualidade obtida nos ajustes. Em relação aos valores de γ , não pudemos encontrar nenhuma relação com o material utilizado na confecção das amostras ou com o parâmetro variado.

Sistema	γ	χ^2
Estádio de Sinai, (a)	1.23	2.1×10^{-5}
Estádio de Sinai - flexurais, (b)	0.95	6.9×10^{-5}
Bilhar de Sinai 3D - massa, (c)	1.18	1.5×10^{-5}
Bilhar de Sinai 3D - temperatura, (d)	1.30	9.3×10^{-6}
Bilhar de Robnik, (d)	1.42	5.3×10^{-4}

Tabela 5.1: Valores de γ e χ^2 para cada um dos sistemas estudados.

Estes resultados foram publicados na revista *Physical Review E*, 65, 057203. A cópia do artigo é apresentada no Apêndice B.

5.2 Estádio de Sinai

O estádio de Sinai é um bilhar construído a partir do estádio 'real' em que, para se evitar a presença de uma classe de órbitas periódicas entre os lados paralelos, se retira um círculo de seu centro. O desenho esquemático deste bilhar, e a foto do ressonador construído são apresentados na Figura 5.2. A espessura da amostra é de 1,6 mm e, de acordo com o valor de c , apresentado na seção 3.2, teremos uma frequência de corte $f_c = 625 \text{ kHz}$.

Figura 5.2: (a) Desenho esquemático do estádio de Sinai, com as medidas utilizadas. (b) Foto da amostra confeccionada no laboratório.

O estádio de Sinai é um sistema caótico que já foi estudado experimentalmente por P. Bertelsen [25], sendo assim um bom sistema

para testarmos a qualidade de nossa montagem experimental.

De acordo com a Teoria de Matrizes Aleatórias, as estatísticas do espectro do estádio de Sinai devem seguir as de um espectro do tipo GOE. Devemos ter em mente que para o caso de placas ressonadoras, os modos simétricos se propagam independentemente dos modos assimétricos e, assim, devemos esperar que o espectro medido tenha as características da sobreposição de dois espectros GOE independentes.

Foram medidos cinco traços na faixa de frequências entre 400 Hz e 500 kHz, com resolução de 1 Hz. Depois da análise dos traços, obtivemos um espectro com 1024 níveis. A função escada do espectro obtido é mostrada na Figura 5.3, onde podemos notar dois comportamentos distintos. Na primeira parte, que possui 110 níveis, há um crescimento aparentemente linear da função escada, o que demonstra que a densidade de níveis é praticamente constante nesta região. Em $f=27$ kHz, há uma mudança brusca de comportamento, a partir do qual a derivada da função passa a aumentar com f .

Figura 5.3: Função escada do espectro experimental do ressonador no formato de do estádio de Sinai. No detalhe, é possível verificar a mudança de comportamento na região de 27kHz.

De acordo com os modelos elastodinâmicos discutidos na seção 2.4.2, esperamos que a densidade de níveis cresça com a frequência, fato este que não foi verificado para a primeira parte do espectro. A falta de

uma expressão apropriada para o comportamento médio da função escada gera a impossibilidade do desdobramento do espectro, fazendo com que optemos por analisar apenas a segunda parte. Desta maneira, foram descartados primeiros 114 pontos e a análise apresentada a seguir será feita com os 910 pontos restantes. O desdobramento do espectro foi efetuado com o ajuste de um polinômio de 3º grau, conforme discutido na seção 4.2.3.

Apresentamos, na Figura 5.4(a), os resultados obtidos para $P(S)$. Podemos observar que estes resultados são compatíveis com a sobreposição de dois espectros do tipo GOE independentes (indicado na figura como 2GOE), o que corrobora o resultado esperado.

Figura 5.4: Distribuição de primeiros vizinhos do estádio de Sinai obtida (a) em nosso laboratório e (b) por P. Bertelsen [25].

Em comparação com os resultados obtidos por P. Bertelsen (Figura 5.4(b)) podemos observar que nossos resultados apresentam maior coerência com a previsão para 2GOE, principalmente nos valores de S pequeno.

Na Figura 5.5(a), apresentamos os resultados obtidos para a rigidez espectral e, como se pode ver, os resultados obtidos são novamente compatíveis com a sobreposição de dois espectros GOE. Observamos também que, à medida em que o valor de L cresce, o valor obtido experimentalmente tende a desviar da curva esperada. Esta tendência de distanciamento também foi observada por P. Bertelsen [25], como pode ser visto na Figura 5.5(b), onde é apontado que este efeito pode estar relacionado com a presença de órbitas periódicas que não tenham sido removidas pela aplicação do método *cut-off*.

Figura 5.5: Rigidez espectral do estádio de Sinai obtida (a) em nosso laboratório e (b) por P. Bertelsen [25].

Andersen *et. al.* [40] separaram os modos anti-simétricos dos simétricos do espectro de um ressonador caótico e encontraram que, para o primeiro caso, os valores de Δ_3 são compatíveis com os valores de um espectro GOE. Entretanto, os valores obtidos para os modos no plano (simétricos) ficaram acima do esperado. Os autores argumentam que este resultado pode indicar que o acoplamento dos modos no plano pode não ser total, restando ainda alguns modos puros no espectro.

Considerando esta proposta, comparamos nossos resultados com uma curva obtida para a sobreposição de 3 espectros GOE independentes, onde supomos que 2% dos níveis simétricos são puros, e que eles formam um espectro independente. Os valores obtidos são apresentados na Figura 5.6, onde podemos observar maior coerência com os dados experimentais, atestando assim a plausibilidade da hipótese levantada por Andersen.

Figura 5.6: Rigidez espectral do estádio de Sinai obtida experimentalmente, comparada com as curvas para 2GOE e para a sobreposição de três espectros GOE independentes, com densidades relativas $g_1 = 0,50$, $g_1 = 0,48$ e $g_1 = 0,02$.

5.2.1 Estádio de Sinai com Dessimetrização Planar

Procedemos com a dessimetrização planar do estádio de Sinai, que consiste na remoção de seus da simetria em relação às suas faces

paralelas, com o intuito de que os acoplamentos dos modos flexurais e no-plano resultem na medida de um único espectro do tipo GOE. As medidas dos traços foram feitas na faixa de 400 Hz a 512 KHz, de onde obtivemos um espectro com 990 autofrequências. A foto e o desenho esquemático do estádio de Sinai com dessimetrização planar são apresentados na Figura 5.7.

Figura 5.7: (a) Foto e (b) Desenho esquemático do estádio de Sinai após o procedimento de dessimetrização planar.

Depois de efetuarmos o desdobramento do espectro, construímos as figuras de $P(S)$ e $\Delta_3(L)$, que são apresentados na Figura 5.8. Podemos notar que as estatísticas de $P(S)$ estão de acordo com um espectro tipo GOE. No caso da rigidez espectral, notamos novamente um desvio do resultado esperado com o aumento L . É importante ressaltar que outros trabalhos com este formato de ressonador apresentaram desvios semelhantes aos nossos [25].

Figura 5.8: (a) $P(s)$ e (b) $\Delta_3(L)$ obtidos com o estádio de Sinai com dessimetriação planar.

5.3 Termodinâmica de espectros gaussianos finitos.

Na seção 2.3.7 foi apresentado um modelo termodinâmico desenvolvido por Dyson e Metha para calcular grandezas termodinâmicas de espectros de *ensembles* circulares infinitos, onde a motivação física é a de que os níveis de energia podem ser considerados como as posições angulares de cargas coulombianas confinadas em um anel de raio unitário [14].

Apresentaremos a seguir um modelo que desenvolvemos para derivar as expressões da energia (U) e do calor específico (C) de espectros finitos de *ensembles* gaussianos. Neste caso, podemos considerar que os níveis do espectro são representados por cargas coulombianas dispostas em uma linha reta. O potencial eletrostático de uma determinada configuração será dado por:

$$W = - \sum_{j>i} \ln |x_j - x_i| \quad (5.4)$$

onde x_i , $i=1,2,3,\dots,N$, são as posições das cargas. Para garantir que a densidade ρ do gás seja estável, devemos considerar um potencial de confinamento $V(x)$. A conexão entre ρ e $V(x)$ é dada pela equação de campo médio de Dyson [14]:

$$P \int \frac{\rho(t)dt}{x-t} = V'(x), \quad (5.5)$$

onde P é o valor principal.

Para uma sequência de níveis no intervalo $[-L, L]$, queremos descobrir qual o potencial de confinamento produzirá um espectro desdobrado, ou seja, $\rho(t)=1$. Utilizando a equação (5.5), chegamos à expressão

$$V(x) = - \left\{ (L-x) \left[\frac{1}{2} - \ln \left(\frac{L-x}{2L} \right) \right] + (L+x) \left[\frac{1}{2} - \ln \left(\frac{L+x}{2L} \right) \right] \right\}, \quad (5.6)$$

na qual o termo constante é adicionado de forma a garantir que $\langle V(x) \rangle = 0$.

Devemos agora adicionar a condição de que a energia deve ser nula quando a temperatura do sistema vai a zero. Para isto, utilizamos um

argumento físico de que, na medida em que a temperatura diminui, os níveis de energia flutuarão cada vez menos, até chegar à situação em que todos estão equidistantes. Este caso corresponde ao espectro do tipo cerca $\xi = (\xi_1, \xi_2, \dots, \xi_i, \dots, \xi_{2L})$, que tem potencial eletrostático W_0 dado por:

$$W_0 = - \sum_{j>i} \ln |j-i|. \quad (5.7)$$

Podemos finalmente escrever a energia de um espectro finito gaussiano $x = \{x_1, x_2, \dots, x_i, \dots, x_N\}$ como:

$$W(L) = - \sum_{j>i} \ln \left| \frac{x_j - x_i}{j-i} \right| + \sum_{i=1}^N [V(x_i) - V(\xi_i)], \quad (5.8)$$

onde $V(\xi_i)$ é o potencial de confinamento do i -ésimo nível do espectro tipo cerca e $N = 2L$.

A energia interna média $U(L) = \langle W(L) \rangle$ pode ser escrita analiticamente a partir de (5.8) em termos da “função agrupamento” (*cluster function*) Y_2 como:

$$U(L) = - \int_0^{2L} dr (2L-r) Y_2(r) \ln(r) + \sum_{n=1}^N [(2L-n) \ln(n) + V(\xi_n)]. \quad (5.9)$$

O calor específico será escrito em função da variância da energia $\sigma_U^2 = \langle U^2(L) \rangle - \langle U(L) \rangle^2$:

$$C = \beta^2 \frac{\sigma_U^2}{N}, \quad (5.10)$$

ou, se definirmos a energia por partícula $u = \frac{U}{N}$, o calor específico será dado por:

$$C = N \beta^2 \sigma_u^2, \quad (5.11)$$

onde $\beta = 1, 2, 4$ (GOE, GUE, GSE) faz o papel do inverso da temperatura.

Apesar de não termos obtido uma expressão analítica para o calor específico, é fácil notar que sua expressão dependerá também das funções agrupamento Y_3 e Y_4 , que medem a correlação de três e quatro níveis, respectivamente.

A partir de (5.9), analisamos a dependência de u com N para um espectro GOE, conforme apresentado na Figura 5.9. Adicionamos também o valor do *ensemble* ortogonal circular (COE) no limite de $N \rightarrow \infty$, $u(\beta=1) = 0.365$. Podemos notar que a energia do espectro aumenta com L e tende ao valor obtido por Dyson e Metha. A convergência destes valores é, de fato, esperada no limite termodinâmico de $N \rightarrow \infty$ [16].

Figura 5.9: Energia por partícula obtida analiticamente para um espectro do tipo GOE finito. Os valores tendem para o resultado previsto para ensembles circulares gaussianos a medida em que o número de níveis aumenta.

5.3.1 Simulações com matrizes aleatórias

A primeira aplicação dos resultados obtidos para as grandezas termodinâmicas foi feita com espectros gerados numericamente, onde desenvolvemos um programa para gerar espectros desdobrados do tipo GOE e calcular sua energia. A construção do espectro de tamanho se inicia através da criação de uma matriz $M \times M$ simétrica, cujos elementos são aleatórios e possuem distribuição normal com média zero e desvio padrão igual a um. Os elementos da diagonal são corrigidos para que tenham desvio padrão igual a $\sqrt{2}$. O espectro $y = \{y_1, \dots, y_N\}$ é obtido a partir da diagonalização da matriz, e o espectro desdobrado $x = \{x_1, \dots, x_N\}$

é obtido a partir da expressão integral da função do semicírculo [41]:

$$x_i = \frac{1}{\pi} \left\{ N \arcsen \left(\frac{y_i}{\sqrt{2N}} \right) + \frac{y_i}{2} \sqrt{2N - y_i^2} \right\}. \quad (5.12)$$

Para evitar os efeitos causados pela finitude do espectro, geralmente seus níveis iniciais e finais são descartados.

De posse do espectro desdobrado, calculamos a energia u a partir da equação (5.8). Este é um programa de fácil construção, mas devemos levar em conta que o número de níveis dentro de um intervalo L poderá variar, não valendo a condição de que $N = 2L$. Temos assim a opção de fixar nos cálculos o número de níveis ou o tamanho do intervalo. Acreditamos ser mais conveniente fixar o tamanho do intervalo, já que este não depende das características locais do espectro. O calor específico é obtido através da estatística dos valores de u obtidos com o sorteio de diversas matrizes.

Calculamos o valor da energia dos espectros de 11.000 matrizes GOE geradas numericamente para o intervalo de $L=450$, o que nos dá uma média de 900 níveis por espectro. Os espectros foram construídos, inicialmente, com 3000 pontos e foram aproveitados os pontos que, depois de desdobrados, se encontravam no intervalo $[-L, L]$.

Construímos um histograma dos valores obtidos para a energia que é apresentado na Figura 5.10. Observamos que a distribuição de energia obtida é uma gaussiana com valor médio $\langle u \rangle = 0,362$, valor muito próximo do obtido com a expressão analítica (5.9), onde obtivemos $u(L=450) = 0,361$. O calor específico foi obtido a partir da variância da curva ajustada, $C = 0,285$. Como dissemos anteriormente, não temos a expressão analítica de C em função de L , de forma que devemos comparar o nosso resultado com aquele obtido para um espectro COE infinito, $C = 0,266$. A diferença relativa entre os dois resultados é de 7%, que acreditamos estar relacionada aos efeitos da finitude dos espectros.

Figura 5.10: Histograma dos valores de energia obtido com 11000 matrizes, no intervalo determinado por $L=450$. A partir do histograma obtivemos os valores $u=0,362$ e $C=0,285$, valores estes próximos dos previstos pela teoria.

5.3.2 Termodinâmica dos estádios de Sinai.

Para obtermos o calor específico de um espectro, devemos ter certo número de valores de energia para podermos calcular sua variância. Para obtermos mais de um valor de energia de um espectro experimental, utilizamos um artifício no qual particionamos o espectro em intervalos (I_n), de tamanho w , e calculamos a energia ε_i de cada intervalo. Os cálculos são feitos sorteando um ponto inicial de varredura P_{ini} a partir do qual as energias serão calculadas em passos de w . O sorteio de ponto P_{ini} é feito em um intervalo inicial (I_0). A Figura 5.11 exemplifica como é feita a varredura.

Figura 5.11: Ilustração de como o espectro é varrido. Calculamos a energia para cada T_i , que possui tamanho w .

Terminada a varredura do espectro, teremos um conjunto de valores de energia ε_i , a partir do qual podemos obter os valores de u e de C através das expressões:

$$u = \langle \varepsilon \rangle = \frac{1}{N} \sum_i \varepsilon_i, \text{ e} \quad (5.13)$$

$$C' = N \left[\frac{1}{N} \sum_i \varepsilon_i^2 - \left(\frac{1}{N} \sum_i \varepsilon_i \right)^2 \right], \quad (5.14)$$

onde redefinimos o calor específico como $C' = \frac{C}{\beta^2}$, por ser uma grandeza que podemos medir diretamente de um espectro, sem que façamos suposições a priori sobre o seu tipo de estatística.

Os resultados são apresentados na forma de gráficos dessas grandezas em função de w . Para estimarmos a incerteza do método, u e C' são calculados diversas vezes para diferentes valores de P_{ini} , mantendo-se fixo o valor de w . O programa desenvolvido para estes cálculos foi escrito em MatLab e sua listagem é apresentada no Apêndice A-IV.

Estádio de Sinai com dessimetrização planar

O primeiro espectro no qual testamos nosso modelo termodinâmico foi o estádio de Sinai com dessimetrização planar que, por ser tipo GOE, pode ter seus valores de u e C' comparados com valores esperados. Os resultados obtidos para a energia e calor específico são apresentados na Figura 5.12.

Figura 5.12: Energia e calor específico calculados para o espectro experimental de um estádio de Sinai com dessimetrização planar.

A primeira coisa que podemos notar, é que os valores da energia são estáveis em relação ao tamanho da janela. Isto mostra que o método de particionamento do espectro em janelas é robusto e pode ser utilizado com segurança para o cálculo de u . O valor obtido utilizando o espectro inteiro foi $u_{SD} = 0.41$. Se compararmos este valor com aqueles obtidos pelo cálculo analítico e pelo valor obtido numericamente, notaremos que este valor está acima do esperado. Observando o histograma da Figura 5.10, podemos verificar que a probabilidade de encontrarmos uma energia $u_{SD} = 0.41$ é muito baixa.

A diferença dos valores da energia é coerente com os resultados apresentados anteriormente, se considerarmos que tanto Δ_3 quanto u são grandezas relacionadas com as correlações entre dois níveis do espectro. Assim, o distanciamento encontrado para a rigidez espectral deverá aparecer no valor da energia. Além disso, devemos notar que um desvio para cima na curva da rigidez espectral indica um comportamento mais próximo de um espectro do tipo Poisson, o que também é coerente com a obtenção de uma energia maior.

Passamos agora a analisar o resultado obtido para o calor específico. Neste caso, o resultado se apresentou estável para valores de w entre 100 e 150, o que demonstra maior sensibilidade ao tamanho da janela. Tirando uma média dos valores de C' , obtemos $C'_{SD} = 0,58$, valor significativamente acima do obtido numericamente, $C' = 0,285$.

Estádio de Sinai

O gráfico de u e C' do espectro do estádio de Sinai sem dessimetrização planar é apresentado na Figura 5.13. Para este espectro obtivemos os valores $u_s = 0,67$ e $C'_s = 0,87$, ambos maiores do que os obtidos para o caso com dessimetrização planar.

Com relação à energia, podemos novamente afirmar que o aumento observado é esperado já que a sobreposição de dois espectros GOE terá estatísticas Δ_j maiores do que às de um espectro GOE.

Figura 5.13: Valores de u e C' obtidos para o espectro experimental do estádio de Sinai.

Os resultados apresentados até o momento mostram que pudemos diferenciar os dois espectros experimentais a partir do cálculo das grandezas termodinâmicas. Nas seções seguintes, ao apresentarmos os resultados obtidos com outros sistemas, passaremos a incorporar estes cálculos na análise.

5.4 Ressonadores Triangulares

Desde que o estudo de bilhares veio à tona, uma série de trabalhos procura estabelecer critérios para classificação de sistemas triangulares [8, 11, 42, 43, 44, 45]. Os bilhares triangulares podem ser integráveis, pseudo-integráveis ou não integráveis, a depender de seus ângulos.

No caso em que estes são frações irracionais de π o bilhar é não integrável e espera-se que tenha estatísticas do tipo E_{IG} . Os triângulos que têm seus ângulos iguais a $\left(\frac{\pi}{3}, \frac{\pi}{3}, \frac{\pi}{3}\right)$, $\left(\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{4}\right)$ e $\left(\frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{6}\right)$ (número de genus $g=1$) são integráveis e as estatísticas de seus espectros serão do tipo Poisson. Já os outros triângulos com ângulos que são frações racionais de π ($g>1$) são pseudo-integráveis e, acredita-se que devem seguir uma estatística do tipo semi-Poisson [46].

A. G. Miltenburg e Th. W. Ruijgrok [44] construíram um espaço de parâmetros a partir de simulações numéricas, no qual o ângulo $\frac{\pi}{2}$ foi identificado como um divisor de comportamentos. Baseados nas estatísticas de $\Delta_3(L)$, eles sugerem que “em geral, os triângulos agudos são mais integráveis do que os obtusos”, na medida em que tendem a apresentar valores maiores de Δ_3 para um determinado valor de L .

Em nossos estudos, efetuamos medidas dos espectros de três ressonadores triangulares com dessimetrização planar, cujos ângulos são apresentados na Tabela 5.2. O primeiro sistema é um triângulo equilátero T_1 , e os triângulos T_2 e T_3 possuem ângulos irracionais em unidades de π e são classificados como não-integráveis. O que diferencia estes dois triângulos é o fato de que o primeiro possui um ângulo reto e o segundo tem os três ângulos menores do que $\frac{\pi}{2}$.

As amostras, cujas fotos são apresentadas na Figura 5.14, possuem espessura de 0,8 mm, de maneira que $f_c = 1,25 MHz$. As medidas foram

feitas na faixa de 400Hz a 1MHz .

Ângulos dos ressonadores triangulares estudados			
Triângulo	$\alpha(\pi rad)$	$\beta(\pi rad)$	$\gamma(\pi rad)$
T1	$\frac{\pi}{3}$	$\frac{\pi}{3}$	$\frac{\pi}{3}$
T2	$\frac{\pi}{2}$	$\frac{\sqrt{5}-1}{4}$	$\left(\frac{1}{2} - \frac{\sqrt{5}-1}{4}\right)$
T3	$\frac{\sqrt{2}-1}{2}$	$\frac{\sqrt{5}-1}{4}$	$\left(1 - \frac{\sqrt{2}-1}{2} - \frac{\sqrt{5}-1}{4}\right)$

Tabela 5.2: Ângulos utilizados para construção dos triângulos.

Figura 5.14: Fotos dos ressonadores triangulares.

5.4.1 Triângulo T₁ (equilátero)

As medidas dos traços do ressonador T₁ forneceram um espectro com 965 pontos. O gráfico da DPV e de Δ_3 é apresentado na Figura 5.15. As estatísticas de P(S) não concordam nem com estatísticas do tipo Poisson nem com espectros GOE, o que fica ainda mais evidente ao observarmos a rigidez espectral. Se por um lado o sistema parece apresentar repulsão de níveis, os valores da rigidez espectral indicam fraca correlação entre níveis distantes.

Bogomolny *et. al.* propuseram que alguns sistemas pseudo-integráveis possuem estatísticas espectrais semelhante a sistemas de Anderson no ponto de transição [46], e introduziram uma estatística chamada semi-Poisson que, vale lembrar, não possui caráter universal

como os *ensembles* gaussianos ou a estatística de Poisson [47]. Nesta estatística, é assumido que os níveis do espectro possuem correlação apenas com os primeiros vizinhos. A expressão da DPV de um espectro do tipo semi-Poisson é dada por:

$$P(S) = 2Se^{-2S}, \quad (5.15)$$

de onde se pode notar que este espectro prevê a repulsão de níveis. Esta curva foi adicionada à Figura 5.15(a) e podemos observar que os resultados experimentais são mais bem descritos por esta distribuição. O fato da correlação entre níveis ocorrer apenas para a primeira vizinhança nos dá outro indício de que o espectro de T_1 seja do tipo semi-Poisson, dada a rapidez com que os valores da rigidez espectral aumentam.

Figura 5.15: (a) $P(S)$ e (b) rigidez espectral obtidos com o triângulo T_1 (equilátero).

O gráfico das grandezas termodinâmicas de T_1 é apresentado na Figura 5.16. Podemos notar que o valor $u_{T_1} = 0,64$ é próximo do valor obtido para o estádio de Sinai $u_s = 0,67$. Entretanto, o valor calculado de $C'_{T_1} = 2,95$ é bastante diferente de $C'_s = 0,87$, mostrando que a grandeza C' também é importante para uma boa caracterização do espectro. O alto valor de C' é coerente com um espectro em que a correlação entre os níveis se dá apenas na primeira vizinhança já que o calor específico irá depender das correlações entre até quatro níveis.

Figura 5.16: Energia e calor específico do espectro do triângulo T_1 (equilátero).

5.4.2 Triângulo T_2 (retângulo)

Para o triângulo T_2 , obtivemos um espectro com 1265 pontos. As estatísticas de $P(S)$ e a rigidez espectral são apresentadas na Figura 5.17. Podemos observar, para $P(S)$, forte concordância com um espectro do tipo GOE. Para a rigidez espectral, encontramos o mesmo tipo de discrepância observada no estádio de Sinai com dessimetrização planar.

Figura 5.17: (a) $P(S)$ e (b) rigidez espectral obtidos com o triângulo T_2 (retângulo).

As grandezas termodinâmicas são apresentadas na Figura 5.18, de onde podemos concluir que $u_{T_2} = 0,42$ e $C'_{T_2} = 0,47$. Comparando este resultado, assim como os das grandezas termodinâmicas, com os do estádio de Sinai com dessimetrização planar, notamos que eles são compatíveis se considerarmos as variações obtidas. Isso mostra que, mesmo que não estejam em conformidade total com os resultados teóricos e com os obtidos numericamente, os valores das grandezas termodinâmicas u e C' podem ser significativos para a caracterização de espectro de

ressonadores acústicos.

Figura 5.18: Energia e calor específico do espectro do triângulo T_2 .

5.4.3 Triângulo T3 (agudo)

Na primeira análise dos traços do ressonador T3, nós obtivemos um espectro com 1237 ressonâncias, mas, devido a diferenças sistemáticas encontrados no início e no final do espectro, ao procedermos com o desdobramento, tivemos que desprezar níveis e obtivemos um espectro com 969 níveis. Os valores de $P(S)$ (Figura 5.19(a)) são coerentes com um espectro GOE, entretanto podemos observar certo desvio da curva na região de $1 < S < 1,5$. O. Bohigas e M. P. Pato [48, 49] sistematizaram o estudo de perda de níveis de espectros gaussianos e notaram desvios semelhantes, o que pode indicar que esta seja a causa da diferença. Outro fator que nos leva a acreditar nesta hipótese é o alto valor da contagem encontrado para $S=2,3$.

Quando analisamos a rigidez espectral (Figura 5.19(b)), encontramos valores muito próximos daqueles observados em T_2 . Se, por um lado, este resultado corrobora o fato de que o espectro obtido é do tipo GOE, temos em contrapartida que não pudemos verificar as expectativas de A. G. Miltenburg e Th. W. Ruijgrok. Pretendemos em breve medir o espectro de

um triângulo com um ângulo maior do que $\frac{\pi}{2}$ a fim de obtermos resultados mais conclusivos.

Em relação às grandezas termodinâmicas, apresentadas na Figura 5.20, o valor $u_{T_3} = 0,41$ é novamente alcançado para um espectro do tipo GOE, por outro lado, o valor $C'_{T_3} = 0,42$ é menor do que o encontrado para os outros ressonadores GOE, ficando mais próximo do valor obtido para as simulações com matrizes aleatórias.

Figura 5.19: (a) $P(S)$ e (b) rigidez espectral obtidos com o triângulo T_3 (genérico).

Figura 5.20: Energia e calor específico do espectro do triângulo T_3 .

5.5 Sistema circulares

Uma membrana circular é um sistema integrável que possui seus modos de ressonância definidos por dois parâmetros (m,n) inteiros, sendo a frequência de ressonância de um determinado modo o valor da n -ésima raiz da função de Bessel $J_m(x)$ de ordem m [50]. No caso de um bilhar com formato de $\frac{1}{4}$ de círculo, as novas condições de contorno impõem a restrição de que $m=\text{par}$.

A partir dos zeros da função de Bessel, geramos um espectro ordenado e obtivemos a DPV, a rigidez espectral, assim como as grandezas termodinâmicas U e C' . Utilizamos 3 espectros com os 10000 primeiros zeros das funções de Bessel, sendo um deles com todas as ordens de J_m e os outros dois apenas com as ordens pares ou ímpares. As estatísticas $P(S)$ e Δ_3 são apresentadas na Figura 5.21. Observamos que, no caso de $P(S)$, os três espectros estão de acordo com os resultados esperados. Ao observarmos as estatísticas de Δ_3 notamos que, apesar dos três espectros apresentarem os mesmos resultados, eles se distanciam do valor esperado à medida que L aumenta.

Figura 5.21: (a) Distribuição de espaçamento e (b) rigidez espectral de espectros formados pelos zeros das funções de Bessel com e sem separação das ordens da função. Apesar de um desvio em Δ_3 para L grande, os três resultados estão em perfeito acordo.

Observando agora os resultados de u , mostrados na Figura 5.22, verificamos que as energias dos espectros formados somente com J_{par} e com J_{impar} são iguais, e menores do que o espectro onde se utiliza todos os

J_m . Este é um resultado ao qual devemos chamar atenção, pois mostra que algumas particularidades do espectro que são difíceis de serem notadas com as análises usuais podem ser detectadas com o cálculo de u . Já no caso de C' , não é possível diferenciar os espectros. Entretanto, nos chamam a atenção os altos valores obtidos para esta grandeza com espectros regulares.

Figura 5.22: Energia e calor específico dos espectros dos zeros de J_m . No caso da energia, é possível se distinguir o espectro em que foram utilizadas todas as ordens daqueles em que foram utilizadas somente as ordens pares ou ímpares.

5.5.1 Ressonador circular

Obtivemos, para o triângulo equilátero, resultados que se mostraram diferentes daqueles esperados para um sistema integrável. Decidimos assim, medir o espectro de outro cujo bilhar análogo é integrável mas, desta vez fazer as medidas para os casos sem a dessimetrização planar e, posteriormente, removendo a simetria em relação ao plano. Devemos lembrar, entretanto, que as amostras são dessimetrizadas.

As medidas foram efetuadas na faixa entre 50Hz e 1024kHz e obtivemos um espectro de 4050 níveis. Os gráficos de $P(S)$ e Δ_3 obtidos são apresentados na Figura 5.23. Podemos notar que, aparentemente, a DPV de um espectro tipo semi-Poisson é a que melhor descreve os resultados experimentais. Entretanto, como o este ressonador não é dessimetrizado em relação ao plano, é de se esperar que não haja acoplamento de níveis e que espectro do tipo Poisson. Desta maneira, levantamos a hipótese de que os resultados poderiam ser explicados considerando perda de níveis, devido ao fator de qualidade das amostras.

Os estudos de O. Bohigas e M. P. Pato [48, 49] para perda de níveis foram feitos de maneira a supor que as perdas de níveis no espectro se dão de maneira aleatória. Entretanto, neste caso, estamos interessados em estudar os efeitos causados pela perda de níveis ocasionada pela limitação experimental.

Para estimarmos o efeito da perda de níveis muito próximos, fizemos comparações com o espectro formado pelos zeros das funções J_{par} , que simulam, em primeira aproximação, os modos de vibração de um ressonador circular dessimetrizado [48]. O estudo é feito removendo os níveis muito próximos do espectro formado por $J_{par}(x)$, de maneira a simular a perda de níveis dos espectros experimentais. O tamanho do espectro, depois das remoções, terá o mesmo número de níveis do espectro experimental. O desdobramento do espectro é feito depois de desprezarmos os níveis.

Figura 5.23: $P(S)$ e Δ_3 de um espectro de um ressonador circular. Podemos notar discrepâncias entre os resultados obtidos e os esperados para um espectro do tipo Poisson.

A condição para que um nível x_i do espectro ordenado não seja considerado é que:

$$x_{i+1} - x_i < \frac{x_{i+1} + x_i}{2Q_{\text{exp}}}, \quad (5.16)$$

onde Q_{exp} é dado pela equação (3.1), evitando escolhas arbitrárias de Q . Como apresentado na seção 3.2, o valor de Q_{exp} estimado com medidas no vácuo foi de $Q \sim 10^4$. Os resultados da simulação de perdas de níveis são apresentados para a faixa de $Q = 1 \times 10^4$ até $Q = 4 \times 10^4$.

Os resultados de $P(S)$ e Δ_j são apresentados na Figura 5.24. A DPV nos mostra que a perda de níveis gera alterações coerentes com as observadas experimentalmente. Por outro lado, o gráfico da rigidez espectral é extraordinariamente compatível com perdas de níveis que simulam $Q = 2 \times 10^4$. Queremos ressaltar que este valor é muito próximo do obtido experimentalmente.

Se observarmos agora o gráfico da energia (Figura 5.25), notaremos novamente que a simulação de perda de níveis corrige a discrepância observada anteriormente. Entretanto, ao compararmos os valores do calor específico, notaremos que a simulação de perda de níveis gera grande instabilidade para os valores menores de Q , além de distanciar do valor obtido experimentalmente.

Acreditamos que os resultados encontrados mostram que a hipótese de perda de níveis de um sistema integrável é razoável para caracterizar o espectro experimental do ressonador circular.

Figura 5.24: (a) $P(S)$ e (b) Δ_3 obtidos para os zeros das funções de Bessel J_{par} simulando a perda de níveis para diferentes valores do fator de qualidade Q . Na comparação com os resultados experimentais, a adoção de $Q=2 \times 10^4$ parece adequada.

Figura 5.25: Grandezas termodinâmicas u e C' calculadas com simulação de perda de níveis devido ao fator de qualidade. Novamente obtemos coerência com o valor $Q=2 \times 10^4$ para a energia u . Por outro lado, o valor de C é bastante sensível à perda de níveis.

5.5.2 Ressonador circular com dessimetrização planar

Tendo verificado a integrabilidade do ressonador circular, fizemos sua dessimetrização planar, a fim de investigar o efeito do acoplamento dos modos nos ressonadores regulares. Esperamos, em princípio, obter um espectro do tipo semi-Poisson, a exemplo do resultado obtido com o triângulo equilátero.

Como pode ser visto na Figura 5.26(a), as estatísticas de $P(S)$ são diferentes de um espectro semi-Poisson, mas também não concordam com um espectro GOE. Esta segunda observação fica mais clara ao observarmos a rigidez espectral, vide Figura 5.26(b). Para a energia e calor específico (Figura 5.27) obtivemos os valores $u = 0,54$ e $C' = 3,8$. Se compararmos com os valores obtidos para o triângulo equilátero ($u_{T1} = 0,64$ e $C'_{T1} = 2,95$), notaremos que os as energias são diferentes,

mas os valores de C' são próximos quando comparados aos obtidos nos outros casos.

Figura 5.26: (a) $P(S)$ e (b) Δ_3 obtidos para o ressonador circular com dessimetrização planar.

Figura 5.27: Energia e calor específico obtidos para o ressonador circular com dessimetriação planar.

Conclusões

Montamos um arranjo experimental para o estudo de ressonadores acústicos, que permite obter espectros com precisão ainda não encontrada na literatura, atestada com a verificação da estabilidade espectral durante a aquisição de dados, que é um procedimento inédito nesse tipo de estudo.

A alta qualidade dos espectros foi alcançada graças a estudos sistemáticos dos parâmetros envolvidos na aquisição, como a temperatura e precisão da varredura, e também com as análises detalhadas dos traços. Estes procedimentos, apesar de terem sido demorados, nos recompensaram com análises inéditas como a determinação das energias dos espectros experimentais e de caracterizações de estatísticas intermediárias.

Para compararmos de maneira mais clara os valores das grandezas termodinâmicas dos 7 espectros experimentais, construímos uma tabela com estes valores, além de uma classificação dos espectros:

Sistema	u	C'	Tipo de espectro
estádio de Sinai com dessimetrização planar	0,41	0,58	GOE
triângulo retângulo	0,42	0,47	GOE
triângulo agudo	0,41	0,42	GOE
estádio de Sinai	0,67	0,87	2GOE
círculo	1,04	7,8	Poisson (com perda de níveis)
triângulo equilátero	0,64	2,95	semi-Poisson
círculo com dessimetrização planar	0,54	3,8	

Grandezas termodinâmicas u e C' dos espectros experimentais dos ressonadores estudados experimentalmente.

Como podemos observar, os sistemas estádio de Sinai com dessimetrização planar, o triângulo retângulo e o triângulo agudo, apresentaram estatísticas que permitiram suas caracterizações como

sistemas GOE. Nas simulações numéricas a energia para esta estatística foi de 0,36 e experimentalmente obtivemos valores experimentalmente iguais de $\bar{u} = 0,41$. Acreditamos que seria interessante a comparação com resultados experimentais obtidos com outros sistemas (como cavidades de microondas, por exemplo), a fim de verificar se os mesmo valores são encontrados. Esta comparação poderia auxiliar no entendimento das divergências observadas em relação ao modelo teórico e às simulações numéricas.

Os valores do calor específico para estes três casos foram muito diferentes entre si e do valor da simulação de 0,28. Esta flutuação não nos permite utilizar o calor específico como parâmetro de classificação dos espectros.

Comparando agora os valores da energia de um espectro GOE com os espectros que foram caracterizados como 2GOE e Poissônico com perda de níveis, verificamos que a energia dos espectros aumentam à medida em que os espectros se aproximam de um espectro do tipo Poisson.

Para os sistemas originalmente integráveis, mas com dessimetrização planar, observamos que os valores das energias são intermediárias entre as energias de sistemas GOE e 2GOE.

As estatísticas de semi-Poisson e de Poisson com perdas de níveis ainda não foram encontradas na literatura como estatísticas de ressonadores acústicos, mostrando que para a simulação numérica explicar os dados experimentais é necessário adequar o modelo a características de modelos experimentais.

O surgimento da estatística intermediária semi-Poisson no triângulo equilátero com dessimetrização planar foi associado ao acoplamento de modos. No caso do ressonador circular, o desvio de uma distribuição poissônica aparentemente pode ser explicada tanto por esta estatística quanto por um sistema do tipo Poisson com perda de níveis. O excelente acordo deste modelo com os dados experimentais nos faz acreditar que esta é a explicação mais plausível.

Os nossos resultados apresentaram grande riqueza de detalhes mostrando que ainda existem muitos aspectos a serem analisados, como a passagem de um sistema integrável para um semi-Poisson e também verificar a distribuição de Poisson com perda de níveis em outros sistemas, por exemplo, numa esfera.

Apêndice A

Listagem dos Programas desenvolvidos:

- I. Detecção dos Picos. (C++)
- II. Ajuste dos Picos. (MatLab)
- III. Método de *cut-off*. (MatLab)
- IV. Cálculo da energia e calor específico de espectros experimentais. (MatLab)

I Detecção dos picos

```
//-----  
-  
#include <clx.h>  
#pragma hdrstop  
USEFORM("Principal.cpp", Form1);  
USEFORM("Detalhe.cpp", Form2);  
USEFORM("ParamArqs.cpp", Atributos); /* TFrame: File Type */  
USEFORM("Configura.cpp", Form3);  
//-----  
-  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TForm1), &Form1);  
        Application->CreateForm(__classid(TForm2), &Form2);  
        Application->CreateForm(__classid(TForm3), &Form3);  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    catch(...)  
    {  
        try  
        {  
            throw Exception("");  
        }  
        catch(Exception &exception)  
        {  
            Application->ShowException(&exception);  
        }  
    }  
    return 0;  
}  
//-----  
-
```



```
//-----  
-  
  
#ifndef PrincipalH  
#define PrincipalH  
//-----  
-  
  
#include <Classes.hpp>  
#include <QControls.hpp>  
#include <QStdCtrls.hpp>  
#include <QForms.hpp>  
#include <QComCtrls.hpp>  
#include <QExtCtrls.hpp>  
#include <QDialogs.hpp>  
#include <QTypes.hpp>  
  
#define NSpecMax 7  
#define NPosPicosMax 20000  
//-----  
-  
  
class TForm1 : public TForm  
{  
    __published: // IDE-managed Components  
        TImage *Image1;  
        TImage *Image2;  
        TImage *Image3;  
        TSpinEdit *PlotaInicio;  
        TSpinEdit *PlotaRange;  
        TButton *Button1;  
        TSpinEdit *Deslocamento;  
        TStatusBar *StatusBar1;  
        TEdit *NomeArqSalva;  
        TButton *Button2;  
        TCheckBox *Sobrescrever;  
        TLabel *Inicio;  
        TLabel *Intervalo;  
        TLabel *Desloc;  
        TTimer *Timer1;  
        TSpinEdit *Velocidade;  
        TLabel *Veloc;  
        void __fastcall IniciaClick(TObject *Sender);  
        void __fastcall FormResize(TObject *Sender);  
        void __fastcall DeslocamentoExit(TObject *Sender);  
        void __fastcall PlotaInicioExit(TObject *Sender);  
        void __fastcall PlotaRangeExit(TObject *Sender);  
        void __fastcall Image1MouseDown(TObject *Sender,  
            TMouseButton Button, TShiftState Shift, int X, int Y);  
        void __fastcall Image1MouseUp(TObject *Sender, TMouseButton Button,  
            TShiftState Shift, int X, int Y);  
        void __fastcall Button1Click(TObject *Sender);  
        void __fastcall Image1MouseMove(TObject *Sender, TShiftState Shift, int X,  
            int Y);  
        void __fastcall FormMouseWheel(TObject *Sender, TShiftState Shift,  
            int WheelDelta, TPoint &MousePos, bool &Handled);  
        void __fastcall FormMouseWheelDown(TObject *Sender, TShiftState Shift,  
            TPoint &MousePos, bool &Handled);  
        void __fastcall FormMouseWheelUp(TObject *Sender, TShiftState Shift,  
            TPoint &MousePos, bool &Handled);  
        void __fastcall Button2Click(TObject *Sender);  
        void __fastcall NomeArqSalvaChange(TObject *Sender);  
};
```

```
void __fastcall FormMouseDown(TObject *Sender, TMouseButton Button,  
    TShiftState Shift, int X, int Y);  
void __fastcall VelocidadeChanged(TObject *Sender, int NewValue);  
void __fastcall Timer1Timer(TObject *Sender);  
private: // User declarations  
    int __fastcall TForm1::Desenha();  
    void __fastcall TForm1::AlteraDeslocamento(int);  
    float __fastcall TForm1::TelaToFreq(int);  
    int __fastcall TForm1::TelaToFaixa(int);  
    int __fastcall TForm1::FreqToIndice(int, int);  
    int __fastcall TForm1::DesenhaEspectro(int);  
    void __fastcall TForm1::DesenhaEscala();  
    void __fastcall TForm1::DesenhaPicos(int);  
    int __fastcall TForm1::FreqToTela(float, int, bool);  
  
public: // User declarations  
    __fastcall TForm1(TComponent* Owner);  
};  
//-----  
-  
extern PACKAGE TForm1 *Form1;  
//-----  
-  
#endif
```

```
//-----  
-  
  
#include <clx.h>  
#pragma hdrstop  
  
#include "Principal.h"  
#include "Detalhe.h"  
#include "Configura.h"  
#include "maxmin.h"  
#include "math.h"  
  
//-----  
-  
  
#pragma package(smart_init)  
#pragma resource "*.xfrm"  
  
TForm1 *Form1;  
extern TForm2 *Form2;  
extern TForm3 *Form3;  
  
extern int NSpecCarregados, NSpecMostrados;  
extern DynamicArray<double> Dados[NSpecMax+1];  
extern float SpecParFreq[3][NSpecMax];  
extern double PosPicos[NPosPicosMax], PosPicos2[NPosPicosMax];  
extern void OrdenaDouble(void *, size_t);  
extern int NPicos, NPicos2;  
//int NPicosTemp;  
  
int ClickPosX, ClickPosY;  
int UnclickPosX, UnclickPosY;  
float FreqTelaClick;  
int FaixaClick;  
TAtributos *Atrib[NSpecMax];  
bool WheelVez;  
bool SentidoAnimacao;  
int SpecPos[NSpecMax];  
double FatorSemShift=10;  
double FatorComShift=5;  
  
//-----  
-  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-  
void __fastcall TForm1::IniciaClick(TObject *Sender)  
{  
    Desenha();  
}  
//-----  
-  
void __fastcall TForm1::FormResize(TObject *Sender)  
{  
    Desenha();  
}
```

```
}
//-----
-
void __fastcall TForm1::DeslocamentoExit(TObject *Sender)
{
    Desenha();
}
//-----
-
void __fastcall TForm1::PlotaInicioExit(TObject *Sender)
{
    Desenha();
}
//-----
-
void __fastcall TForm1::PlotaRangeExit(TObject *Sender)
{
    Desenha();
}
//-----
-
void __fastcall TForm1::ImagemMouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    ClickPosX=X;
    ClickPosY=Y;
}
//-----
-5

void __fastcall TForm1::ImagemMouseUp(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    UnclickPosX=X;
    UnclickPosY=Y;
    int DeltaDrag=ClickPosX-UnclickPosX;

    if(NSpecMostrados==0)
        return;
    if(Timer1->Enabled){
        Timer1->Enabled=0;
        return;
    }

    if(abs(DeltaDrag)>3){
        AlteraDeslocamento(DeltaDrag);
        return;
    }
    else if(Button==0)
    {
        double AdPico=TelaToFreq(ClickPosX);
        int i=NPicos;
        while(i>0 && AdPico<PosPicos[i-1]){
            PosPicos[i]=PosPicos[i-1];
            i--;
        }
        PosPicos[i]=AdPico;
        NPicos++;
    }
    else if(Button==1 && NPicos>0)
```

```
{
    int Indice=0;
    double FreqClick=TelaToFreq(ClickPosX), dist, distemp;
    dist=fabs(FreqClick-PosPicos[0]);
    for(int i=1;i<NPicos;i++)
    {
        distemp=fabs(FreqClick-PosPicos[i]);
        if(distemp<=dist)
        {
            Indice=i;
            dist=distemp;
        }
    }
    for(int i=Indice;i<NPicos;i++)
        PosPicos[i]=PosPicos[i+1];
    NPicos--;
}
Desenha();
}
//-----
-
void __fastcall TForm1::AlteraDeslocamento(int Desloca)
{
    Deslocamento->Value+=PlotaRange->Value*Desloca/Image1->Width;;
    Desenha();
}
//-----
-
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Form3->Show();
}
//-----
-
void __fastcall TForm1::Image1MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
    float value=TelaToFreq(X);
    int value2=TelaToFaixa(Y);
    StatusBar1->Panels->operator [] (1)->Text= Format("Freq:
%f",ARRAYOFCONST((value)));
    if(value2!=-1)
        StatusBar1->Panels->operator [] (0)->Text= Format("Espec:
%d",ARRAYOFCONST((SpecPos[value2])));
}
//-----
-
void __fastcall TForm1::FormMouseWheel(TObject *Sender, TShiftState Shift,
    int WheelDelta, TPoint &MousePos, bool &Handled)
{
    WheelVez=!WheelVez;
}
//-----
-
void __fastcall TForm1::FormMouseWheelDown(TObject *Sender,
```

```
TShiftState Shift, TPoint &MousePos, bool &Handled)
{
    TSpinEdit *Altera;
    double fator;

    if(WheelVez==false)
        return;

    if(!Shift.Contains(ssCtrl))
        Altera=PlotaInicio;
    else
        Altera=PlotaRange;

    if(!Shift.Contains(ssAlt))
        fator=FatorSemShift;
    else
        fator=FatorComShift;

    Altera->Value+=PlotaRange->Value/fator;
    Desenha();
}
//-----

void __fastcall TForm1::FormMouseWheelUp(TObject *Sender,
    TShiftState Shift, TPoint &MousePos, bool &Handled)
{
    TSpinEdit *Altera;
    double fator;

    if(WheelVez==false)
        return;

    if(!Shift.Contains(ssCtrl))
    {
        Altera=PlotaInicio;
        fator=FatorSemShift;
        if(Shift.Contains(ssAlt))
            fator=FatorComShift;
    }
    else
    {
        Altera=PlotaRange;
        fator=0.5;
        if(Shift.Contains(ssAlt))
            fator=2;
    }
    Altera->Value-=PlotaRange->Value/fator;
    Desenha();
}
//-----

//          INICIO DAS FUNCOES DE CONVERSAO
//-----

int __fastcall TForm1::FreqToIndice(int freq,int NSpec)
{
    double A,B;
    int Pos;
```

```
    A=SpecParFreq[0][NSpec];
    B=SpecParFreq[1][NSpec];
    Pos=(int)((freq-A)/B);
    return (int)Pos;
}

//-----

float __fastcall TForm1::TelaToFreq(int xtela)
{
    float freq;
    freq= PlotaInicio->Value + PlotaRange->Value*xtela/Image1->Width;
    return freq;
}

//-----

int __fastcall TForm1::TelaToFaixa(int ytela)
{
    if(NSpecMostrados==0)
        return -1;
    int Spec;
    int Faixa=0;
    int TamFaixa=Image1->Height/NSpecMostrados;
    int FaixaMax=TamFaixa;
    while(ytela>=FaixaMax)
    {
        Faixa++;
        FaixaMax+=TamFaixa;
    }
    return Faixa;
}

//-----

int __fastcall TForm1::FreqToTela(float Freq ,int Faixa,bool verifica)
{
    int Tela, TelaFloat;
    float Correcao;
    Freq+=SpecParFreq[2][Faixa]*Deslocamento->Value;

    TelaFloat=(float)Image1->Width/(float)PlotaRange->Value*(Freq-(float)PlotaIn
icio->Value);
    Tela=(int)TelaFloat;
    if(verifica)
    {
        if(Tela>Image1->Width||Tela<0)
        {
            Tela=-1;
        }
    }
    return Tela;
}

//-----
//          FIM DAS FUNCOES DE CONVERSAO
//-----
```

```
//-----  
//          INICIO DAS FUNCOES DE PLOTAGEM  
//-----  
  
int __fastcall TForm1::DesenhaEspectro(int Posicao)  
{  
    double EscalaX, EscalaY, MaxY, MinY;  
    int IndiceInicial, IndiceFinal;  
    double TelaX, TelaY;  
    int i;  
    int FreqInicial, FreqDesloca;  
    TCanvas *Espec = Imagem->Canvas;  
    DynamicArray<double> teste;  
    int EspectroAtual=SpecPos[Posicao];  
  
    Espec->Pen->Color=clWhite;  
    if (Form3->ArqReferencia->Value==EspectroAtual)  
        Espec->Pen->Color=Espec->Pen->Color;  
  
    FreqDesloca=Deslocamento->Value*(Atrib[EspectroAtual]->FatorDesloca->Value-  
Atrib[Form3->ArqReferencia->Value]->FatorDesloca->Value);  
    IndiceInicial=FreqToIndice(PlotaInicio->Value + FreqDesloca, EspectroAtual);  
    IndiceFinal=FreqToIndice(PlotaInicio->Value + FreqDesloca +  
PlotaRange->Value, EspectroAtual);  
  
    if (IndiceInicial>=Dados[EspectroAtual].Length || IndiceFinal<=0)  
        return 1;  
    IndiceFinal=min(IndiceFinal, Dados[EspectroAtual].Length-1);  
    FreqInicial=PlotaInicio->Value;  
    if (IndiceInicial<0)  
    {  
        IndiceInicial=0;  
        FreqInicial=SpecParFreq[0][EspectroAtual];  
    }  
  
    teste.Length=IndiceFinal-IndiceInicial+1;  
    teste=Dados[EspectroAtual].CopyRange(IndiceInicial, teste.Length);  
    //Busca dos Valores limites do grafico  
    MaxY=Dados[EspectroAtual][IndiceInicial];  
    MinY=Dados[EspectroAtual][IndiceInicial];  
    for (i=IndiceInicial; i<IndiceFinal; i++)  
    {  
        MaxY=max(Dados[EspectroAtual][i], MaxY);  
        MinY=min(Dados[EspectroAtual][i], MinY);  
    }  
    if (MaxY==MinY)  
    {  
        MaxY+=1.0;  
        MinY-=1.0;  
    }  
  
    //Plotagem dos pontos  
  
    EscalaX=(double) Imagem->Width/PlotaRange->Value;  
    EscalaY=(double) Imagem->Height/(NSpecMostrados*(MaxY-MinY));  
  
    float NGraficos=StrToFloat(String(Atrib[EspectroAtual]->Resolucao->Text));  
    float F0=PlotaInicio->Value;
```



```
float Altura=Imagem1->Height;

//Rotina para Desenho de Linhas
TelaX=(FreqInicial- F0)*EscalaX;
TelaY=(MinY - Dados[EspectroAtual][IndiceInicial])*EscalaY +
(Posicao+1)*Altura/NSpecMostrados;
Espec->MoveTo(TelaX,TelaY);
Espec->Pen->Width=3;
if(!Form3->PlotaTipoPontos->Checked){
    for (int i=IndiceInicial+1; i<IndiceFinal; i++)
    {
        TelaX=(FreqInicial- F0 +(i-IndiceInicial)*NGraficos)*EscalaX;
        TelaY=(MinY - Dados[EspectroAtual][i])*EscalaY +
(Posicao+1)*Altura/NSpecMostrados;
        Espec->LineTo(TelaX,TelaY); //Aprox. Metade do Tempo de Pixels
    }
}

//Rotina para Desenho de pontos
if(Form3->PlotaTipoPontos->Checked){
    for (int i=IndiceInicial; i<IndiceFinal; i++)
    {
        TelaX=(FreqInicial- F0 +(i-IndiceInicial)*NGraficos)*EscalaX;
        TelaY=(MinY - Dados[EspectroAtual][i])*EscalaY +
(Posicao+1)*Altura/NSpecMostrados;
        Espec->Pen->Width=5;
        Espec->DrawPoint(TelaX,TelaY); //Aprox. Metade do Tempo de Pixels
    }
}

return 0;
}
//-----
void __fastcall TForm1::DesenhaEscala()
{
    TColor CorEscala=c1Yellow;
    TCanvas *Escala = Image2->Canvas;
    TCanvas *Espec = Imagem1->Canvas;
    int TickMenor=5,TickMaior=7; //Tamanho dos Ticks em pixels
    int Tamanho1,Tamanho2;

    Escala->Pen->Color=CorEscala;
    Espec->Pen->Color=CorEscala;
    Escala->Font->Color=CorEscala;

    // Desenha Contorno do Grafico
    Espec->MoveTo(0,0);
    Espec->LineTo(0,Imagem1->Height-1);
    Espec->LineTo(Imagem1->Width,Imagem1->Height-1);
    Espec->MoveTo(Imagem1->Width-1,Imagem1->Height-1);
    Espec->LineTo(Imagem1->Width-1,0);
    Espec->LineTo(0,0);
    Imagem1->BringToFront();

    //Desenho do eixo e escala X
    Escala->MoveTo(0,0);
    Escala->LineTo(0,Image2->Height/4);
    Escala->MoveTo(Image2->Width/2,0);
```

```
Escala->LineTo(Image2->Width/2, Image2->Height/4);
Escala->MoveTo(Image2->Width-1, 0);
Escala->LineTo(Image2->Width-1, Image2->Height/4);
for(int i=1; i<=9; i++)
{
    Espec->MoveTo(Image1->Width*i/10, Image1->Height);
    Espec->LineTo(Image1->Width*i/10, Image1->Height-TickMenor);
    Espec->MoveTo(Image1->Width*i/10, 0);
    Espec->LineTo(Image1->Width*i/10, TickMenor);
    Escala->MoveTo(Image2->Width*i/10, 0);
    Escala->LineTo(Image2->Width*i/10, Image2->Height/6);
}

//Texto da escala
Escala->TextOut(5, Image2->Height/4, FloatToStrF((double)PlotaInicio->Value, 5, 5, 10));

Tamanho1 =
Escala->TextWidth(FloatToStrF((double)(PlotaInicio->Value+PlotaRange->Value), 5, 5, 10));
Escala->TextOut(Image2->Width-Tamanho1-5, Image2->Height/4, FloatToStrF((double)(PlotaInicio->Value+PlotaRange->Value), 5, 5, 10));

Tamanho2 =
Escala->TextWidth(FloatToStrF((double)(2*PlotaInicio->Value+PlotaRange->Value)/2, 5, 5, 10));
Escala->TextOut((Image2->Width-Tamanho2)/2, Image2->Height/4, FloatToStrF((double)(PlotaInicio->Value+PlotaRange->Value/2), 5, 5, 10));
}
//-----
int __fastcall TForm1::Desenha()
{
    TCanvas *Espec = Image1->Canvas;
    TCanvas *Escala = Image2->Canvas;
    Espec->SetClipRect(Espec->ClipRect);

    if(NSpecMostrados==0 || PlotaRange->Value==0)
        return 1;

    Espec->Pen->Color=clBlack;
    Escala->Pen->Color=clBlack;
    //Limpa Imagens
    Espec->Brush->Color=clBlack;
    Escala->Brush->Color=clBlack;
    Espec->FillRect(Image1->ClientRect); //limpa Image1
    Escala->FillRect(Image2->ClientRect); //limpa Image2

    DesenhaEscala();
    for(int Posicao=0; Posicao<NSpecMostrados; Posicao++)
        DesenhaEspectro(Posicao);
        DesenhaPicos(1);
}
//-----
void __fastcall TForm1::DesenhaPicos(int Referencia)
{
    TCanvas *Espec = Image1->Canvas;
    int YMax=Image1->Height, XTela;
```

```
//Plota lista 2 de Picos
Espec->Pen->Color=clGreen;
for(int i=0;i<NPicos2;i++)
{
    XTela = FreqToTela(PosPicos2[i],0,1);
    if(XTela>=0)
    {
        Espec->MoveTo(XTela,0);
        Espec->LineTo(XTela,YMax);
    }
}
```

```
//Plota lista 1 de Picos
Espec->Pen->Color=clRed;
for(int i=0;i<NPicos;i++)
{
    XTela = FreqToTela(PosPicos[i],0,1);
    if(XTela>=0)
    {
        Espec->MoveTo(XTela,0);
        Espec->LineTo(XTela,YMax);
    }
}
}
```

```
//-----
//      FIM DAS FUNCOES DE PLOTAGEM
//-----
```

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    int i;
    FILE *ArqSalva;
    ArqSalva=fopen(AnsiString(NomeArqSalva->Text).c_str(),"r");
    if(ArqSalva!=NULL&&Sobrescrever->Checked==0){
        Application->MessageBox("O Arquivo Já Existe e a Opção Para
Sobrescrever Está Desabilitada","O Arquivo Não Foi Salvo");
        fclose(ArqSalva);
        return;
    }
    fclose(ArqSalva);
    ArqSalva=fopen(AnsiString(NomeArqSalva->Text).c_str(),"w");
    for(i=0;i<NPicos;i++)
        fprintf(ArqSalva,"%f\n",PosPicos[i]);
    fclose(ArqSalva);
}
```

```
//-----
-
```

```
void __fastcall TForm1::NomeArqSalvaChange(TObject *Sender)
{
    Sobrescrever->Checked=0;
}
//-----
-
```

```
void __fastcall TForm1::FormMouseDown(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    if(Timer1->Enabled){
        Timer1->Enabled=0;
        return;
    }
    if(Button==0 || Button==1){
        SentidoAnimacao=Button;
        Timer1->Enabled=1;
    }
}
//-----
-
```

```
void __fastcall TForm1::VelocidadeChanged(TObject *Sender, int NewValue)
{
    Timer1->Interval=1000/Velocidade->Value;
}
//-----
-
```

```
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    if(SentidoAnimacao==0)
        PlotaInicio->Value+=PlotaRange->Value/(2*FatorSemShift);
    if(SentidoAnimacao==1)
        PlotaInicio->Value-=PlotaRange->Value/(2*FatorSemShift);
    Desenha();
}
//-----
-
```

```
//-----  
-  
#ifndef ConfiguraH  
#define ConfiguraH  
//-----  
-  
#include <Classes.hpp>  
#include <QControls.hpp>  
#include <QStdCtrls.hpp>  
#include <QForms.hpp>  
#include "ParamArqs.h"  
#include <QComCtrls.hpp>  
#include <QExtCtrls.hpp>  
#include <QButtons.hpp>  
#include <QDialogs.hpp>  
#include <QActnList.hpp>  
//-----  
-  
class TForm3 : public TForm  
{  
    __published: // IDE-managed Components  
        TAttributos *Atributos2;  
        TAttributos *Atributos3;  
        TAttributos *Atributos4;  
        TAttributos *Atributos5;  
        TAttributos *Atributos6;  
        TAttributos *Atributos7;  
        TAttributos *Atributos1;  
        TBevel *Bevel2;  
        TLabel *Referencia;  
        TCheckBox *ValorUnico;  
        TEdit *FreqInicial;  
        TEdit *Resolucao;  
        TSpinEdit *ArqReferencia;  
        TButton *Fechar;  
        TEdit *NomeArquivoPico;  
        TCheckBox *AtivoPico;  
        TRadioButton *CarregadoPico;  
        TLCDNumber *NumPontosPico;  
        TSpeedButton *SpeedButton1Pico;  
        TOpenDialog *OpenFileDialogPico;  
        TLabel *Label1;  
        TButton *Unidade;  
        TSpeedButton *SpeedButton1Pico2;  
        TEdit *NomeArquivoPico2;  
        TCheckBox *AtivoPico2;  
        TRadioButton *CarregadoPico2;  
        TLCDNumber *NumPontosPico2;  
        TButton *Unidade2;  
        TOpenDialog *OpenFileDialogPico2;  
        TLabel *Mostra;  
        TLabel *Nome;  
        TLabel *FreqInic;  
        TLabel *Resoluc;  
        TLabel *Label2;  
        TLabel *Label3;  
        TLabel *Label4;  
        TLabel *Label5;  
        TLabel *Label6;
```

```
TLabel *Label7;
TLabel *Label8;
TLabel *Label9;
TCheckBox *PlotaTipoPontos;
void __fastcall FormCreate(TObject *Sender);
void __fastcall ValorUnicoClick(TObject *Sender);
void __fastcall FreqInicialChange(TObject *Sender);
void __fastcall ResolucaoChange(TObject *Sender);
void __fastcall FecharClick(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
void __fastcall SpeedButton1PicoClick(TObject *Sender);
void __fastcall SpeedButton2PicoClick(TObject *Sender);
void __fastcall UnidadeClick(TObject *Sender);
void __fastcall UnidadeClick2(TObject *Sender);
void __fastcall Atributos1SpeedButton1Click(TObject *Sender);
private: // User declarations
    void __fastcall TForm3::Ordena();
public: // User declarations
    __fastcall TForm3(TComponent* Owner);
};
//-----
extern PACKAGE TForm3 *Form3;
//-----
#endif
```

```
//-----  
-  
  
#include <clx.h>  
#include <stdio.h>  
#pragma hdrstop  
  
#include "Configura.h"  
#include "Principal.h"  
//-----  
-  
  
#pragma package(smart_init)  
#pragma link "ParamArqs"  
#pragma resource "*.xfrm"  
TForm3 *Form3;  
  
extern TAttributos *Atrib[NSpecMax];  
TAttributos *AtributoAtual;  
DynamicArray<double> Dados[NSpecMax+1];  
  
double PosPicos[NPosPicosMax];  
double PosPicos2[NPosPicosMax];  
extern int SpecPos[NSpecMax];  
float SpecParFreq[3][NSpecMax]; //vetor com frequencia, resolucao e fator  
de deslocamento  
int NSpecCarregados=0, NSpecMostrados=0;  
int NPicos=0, NPicos2=0;  
  
//-----  
-  
__fastcall TForm3::TForm3(TComponent* Owner)  
: TForm(Owner)  
{  
}  
//-----  
-  
  
void __fastcall TForm3::FormCreate(TObject *Sender)  
{  
    int i;  
  
    //NomeAtributos->Caminho->Text=ExtractFilePath(Application->ExeName);  
    Atrib[0]=Atributos1;  
    Atrib[1]=Atributos2;  
    Atrib[2]=Atributos3;  
    Atrib[3]=Atributos4;  
    Atrib[4]=Atributos5;  
    Atrib[5]=Atributos6;  
    Atrib[6]=Atributos7;  
    ArqReferencia->Max=NSpecMax;  
  
    for(i=0; i<NSpecMax; i++)  
    {  
        AtributoAtual=Atrib[i];  
        AtributoAtual->NumArq->Value=i;  
    }  
}
```

```
//-----  
-  
void __fastcall TForm3::ValorUnicoClick(TObject *Sender)  
{  
    for(int i=0;i<NSpecMax;i++)  
    {  
        AtributoAtual=Atrib[i];  
        AtributoAtual->FreqInicial->Enabled=!ValorUnico->Checked;  
        AtributoAtual->Resolucao->Enabled=!ValorUnico->Checked;  
        FreqInicial->Enabled=ValorUnico->Checked;  
        Resolucao->Enabled=ValorUnico->Checked;  
    }  
}  
//-----  
-  
void __fastcall TForm3::FreqInicialChange(TObject *Sender)  
{  
    for(int i=0;i<NSpecMax;i++)  
    {  
        AtributoAtual=Atrib[i];  
        AtributoAtual->FreqInicial->Text=FreqInicial->Text;  
        AtributoAtual->Resolucao->Text=Resolucao->Text;  
    }  
}  
//-----  
-  
void __fastcall TForm3::ResolucaoChange(TObject *Sender)  
{  
    FreqInicialChange(Sender);  
}  
//-----  
-  
void __fastcall TForm3::FecharClick(TObject *Sender)  
{  
    Form3->Close();  
}  
//-----  
-  
void __fastcall TForm3::FormClose(TObject *Sender, TCloseAction &Action)  
{  
    int pos=0;  
    DecimalSeparator='.';  
    NSpecCarregados=0;  
    NSpecMostrados=0;  
    for(int i=0;i<NSpecMax;i++)  
    {  
        if(Atrib[i]->Enabled&&Atrib[i]->Ativo->Checked){  
            SpecPos[pos]=i;  
            pos++;  
        }  
    }  
  
    if(Atrib[i]->Carregado->Checked)  
        NSpecCarregados++;  
}
```



```
        if(Atrib[i]->Ativo->Checked)
            NSpecMostrados++;

        SpecParFreq[0][i]=StrToFloat(String(Atrib[i]->FreqInicial->Text));
        SpecParFreq[1][i]=StrToFloat(String(Atrib[i]->Resolucao->Text));
        SpecParFreq[2][i]=Atrib[i]->FatorDesloca->Value;
    }
}

//-----

void __fastcall TForm3::SpeedButton1PicoClick(TObject *Sender)
{
    FILE *flepico;
    float lixo,Maior=0;
    AnsiString Nome,Ext;
    if(OpenFileDialogPico->Execute()){
        Unidade->Caption=AnsiString("Hz");
        Nome=OpenFileDialogPico->FileName;
        Form1->NomeArgSalva->Text=Nome;
        Form1->Sobrescrever->Checked=False;
        flepico=fopen(Nome.c_str(),"r");
        Ext=ExtractFileExt(Nome);
        NPicos=0;
        if(Ext==".par"){
            while(fscanf(flepico,"%f %f %lf %f %f %f
%f\n",&lixo,&lixo,&PosPicos[NPicos],&lixo,&lixo,&lixo,&lixo)!=EOF&&NPicos<NP
osPicosMax)
                NPicos++;
        }
        else{
            while(fscanf(flepico,"%lf\n",&PosPicos[NPicos])!=EOF&&NPicos<NPosPicosMax)
                NPicos++;
        }
        fclose(flepico);
        NumPontosPico->Value=NPicos;
        Ordena();
        if(PosPicos[NPicos-1]<1000.0)
            UnidadeClick(Sender);
    }
}

//-----

void __fastcall TForm3::SpeedButton2PicoClick(TObject *Sender)
{
    FILE *flepico;
    float lixo,Maior=0;
    AnsiString Nome,Ext;
    if(OpenFileDialogPico2->Execute()){
        Unidade2->Caption=AnsiString("Hz");
        Nome=OpenFileDialogPico2->FileName;
        flepico=fopen(Nome.c_str(),"r");
        Ext=ExtractFileExt(Nome);
        NPicos2=0;
        if(Ext==".par"){
            while(fscanf(flepico,"%f %f %lf %f %f %f
%f\n",&lixo,&lixo,&PosPicos2[NPicos2],&lixo,&lixo,&lixo,&lixo)!=EOF&&NPicos2
```

```
<NPosPicosMax)
                                NPicos2++;
    }
    else{

while(fscanf(flepico, "%lf\n", &PosPicos2[NPicos2]) != EOF && NPicos2 < NPosPicosMax
)
                                NPicos2++;
    }
    fclose(flepico);
    NumPontosPico2->Value=NPicos2;
    Ordena();
    if(PosPicos2[NPicos2-1]<1000.0)
        UnidadeClick2(Sender);
}
//-----
-

void __fastcall TForm3::UnidadeClick(TObject *Sender)
{
    int i;

    if(Unidade->Caption==AnsiString("Hz"))
    {
        for(i=0; i<NPicos; i++)
            PosPicos[i]*=1000;
        Unidade->Caption="kHz";
    }

    else if(Unidade->Caption==AnsiString("kHz"))
    {
        for(i=0; i<NPicos; i++)
            PosPicos[i]/=1000;
        Unidade->Caption="Hz";
    }
}
//-----
-

void __fastcall TForm3::UnidadeClick2(TObject *Sender)
{
    int i;

    if(Unidade2->Caption==AnsiString("Hz"))
    {
        for(i=0; i<NPicos2; i++)
            PosPicos2[i]*=1000;
        Unidade2->Caption="kHz";
    }

    else if(Unidade2->Caption==AnsiString("kHz"))
    {
        for(i=0; i<NPicos2; i++)
            PosPicos2[i]/=1000;
        Unidade2->Caption="Hz";
    }
}
```

```
}  
//-----  
-  
  
void __fastcall TForm3::Atributos1SpeedButton1Click(TObject *Sender)  
{  
    Atributos1->SpeedButton1Click(Sender);  
  
}  
//-----  
-  
  
void __fastcall TForm3::Ordena()  
{  
    int i,j;  
    double a;  
  
    for(j=1;j<NPicos;j++){  
        a=PosPicos[j];  
        i=j-1;  
        while (i >=0 && PosPicos[i] > a) {  
            PosPicos[i+1]=PosPicos[i];  
            i--;  
        }  
        PosPicos[i+1]=a;  
    }  
}
```

File: D:\Thiago\Doutorado\Programas\Analise de Tracos\Grafico.h 18/2/2005, 0
8:37:22

```
#ifndef GraficoH
#define GraficoH
//-----
-
#include <stdio.h>
#include <Classes.hpp>
#include <QControls.hpp>
#include <QStdCtrls.hpp>
#include <QForms.hpp>
#include <QMenus.hpp>
#include <QTypes.hpp>
#include <QComCtrls.hpp>
#include <QFileCtrls.hpp>
#include <QButtons.hpp>
#include <QExtCtrls.hpp>
#include <Math.hpp>
//-----
--
struct TGrafico
{
    DynamicArray<double> X;
    DynamicArray<double> Y;
    double XMax, XMin, YMax, YMin;
    int NGrafYTotal, NGrafYAtual;
    double AchaMax(DynamicArray<double> &A);
    double AchaMin(DynamicArray<double> &A);
    bool chart;
    void AchaXMax();
    void AchaXMin();
    void AchaYMax();
    void AchaYMin();
    void Plota(TImage *Imagem);
    TGrafico();
};

#endif
```

```
//-----  
-  
#pragma hdrstop  
#include "Grafico.h"  
//-----  
-  
#pragma package(smart_init)  
//-----  
-  
TGráfico::TGráfico()  
{  
    chart=false;  
    X.Length=0;  
    Y.Length=0;  
    NGrafYTotal=1;  
    NGrafYAtual=1;  
    XMax=NULL;  
    XMin=NULL;  
    YMax=NULL;  
    YMin=NULL;  
}  
//-----  
-  
void TGráfico::AchaXMax()  
{  
    XMax=AchaMax(X);  
}  
//-----  
-  
void TGráfico::AchaXMin()  
{  
    XMin=AchaMin(X);  
}  
//-----  
-  
void TGráfico::AchaYMax()  
{  
    YMax=AchaMax(Y);  
}  
//-----  
-  
void TGráfico::AchaYMin()  
{  
    YMin=AchaMin(Y);  
}  
//-----  
-  
double TGráfico::AchaMax(DynamicArray<double> &Vetor)  
{  
    if(Vetor.Length<=0)  
        return NULL;  
    double Max=Vetor[0];  
    for(int i=1;i<Vetor.Length;i++)  
    {  
        if(Vetor[i]>Max)  
            Max=Vetor[i];  
    }  
    return Max;  
}  
//-----
```

```
-
double TGrafico::AchaMin(DynamicArray<double> &Vetor)
{
    if(Vetor.Length<=0)
        return NULL;
    double Min=Vetor[0];
    for(int i=1;i<Vetor.Length;i++)
    {
        if(Vetor[i]<Min)
            Min=Vetor[i];
    }
    return Min;
}
//-----
-
void TGrafico::Plota(TImage *Imagem)
{
    double EscalaX,EscalaY;
    TCanvas *Espec = Imagem->Canvas;
    int XTela, YTela;

    //sai da funcao se nao ha dados ou graficos para serem plotados
    if(Y.Length==0 || NGrafYTotal<=0 || NGrafYAtual>NGrafYTotal)
    {
        Application->MessageBox("Verifique se existem pontos a serem plotados
(Y.Length!=0)\n e se as variaveis NGrafYTotal e NGrafYAtual estao
corretas.", "Erro ao Desenhar");
        return;
    }
    //ajuste automático de alguns parametro referentes aa escala Y
    if(YMax==NULL)
        AchaYMax();
    if(YMin==NULL)
        AchaYMin();
    if(YMax==YMin)
    {
        YMax+=100;
        YMin-=100;
    }
    EscalaY=(double) Imagem->Height / (NGrafYTotal*(YMax-YMin));
    int TamImagemY=Imagem->Height;

    //ajuste automático de alguns parametro referentes aa escala X
    if(X.Length==0)
        chart=true;

    int TamImagemX=Imagem->Width;
    if(chart==true)
        EscalaX=(double) TamImagemX/Y.High;
    else //chart==false
    {
        if(XMax==NULL)
            AchaXMax();
        if(XMin==NULL)
            AchaXMin();
        if(YMax==YMin)
        {
            XMax+=100;
            XMin-=100;
        }
    }
}
```

```
    EscalaX=(double)TamImagemX/(XMax-XMin);  
}  
  
//Desenho dos Pontos  
for (int i=0; i<Y.Length; i++)  
{  
    if(chart==true)  
        XTela=i*EscalaX;  
    else //chart==false  
        XTela=(X[i]-XMin)*EscalaX;  
    YTela=(YMin - Y[i])*EscalaY + TamImagemY*(NGrafYAtual)/NGrafYTotal;  
    //So desenha caso esteja dentro da figura (economia de tempo)  
    if(XTela>=0 && YTela>=0 && XTela<=TamImagemX && YTela<=TamImagemY)  
        Espec->DrawPoint(XTela,YTela);  
}  
}
```

```
//-----  
-  
  
#ifndef ParamArqsH  
#define ParamArqsH  
//-----  
-  
#include <Classes.hpp>  
#include <QControls.hpp>  
#include <QStdCtrls.hpp>  
#include <QForms.hpp>  
#include <QComCtrls.hpp>  
#include <QExtCtrls.hpp>  
#include <QButtons.hpp>  
#include <QDialogs.hpp>  
//-----  
-  
class TAttributos : public TFrame  
{  
    __published: // IDE-managed Components  
        TEdit *NomeArquivo;  
        TSpinEdit *FatorDesloca;  
        TEdit *CaminhoArquivo;  
        TPanel *Status;  
        TCheckBox *Ativo;  
        TRadioButton *Carregado;  
        TEdit *FreqInicial;  
        TEdit *Resolucao;  
        TLCDNumber *NumArq;  
        TLCDNumber *NumPontos;  
        TSpeedButton *SpeedButton1;  
        TOpenDialog *OpenFileDialog;  
        void __fastcall SpeedButton1Click(TObject *Sender);  
private: // User declarations  
    int __fastcall AbreLe(AnsiString,DynamicArray<double> &A);  
public: // User declarations  
    __fastcall TAttributos(TComponent* Owner);  
};  
//-----  
-  
extern PACKAGE TAttributos *Attributos;  
//-----  
-  
#endif
```



```
//-----  
-  
  
#include <clx.h>  
#include <stdio.h>  
#pragma hdrstop  
  
#include "ParamArqs.h"  
#include "Principal.h"  
#include "Configura.h"  
  
//-----  
-  
  
#pragma package(smart_init)  
#pragma resource "*.xrm"  
TAttributos *Atributos;  
extern DynamicArray<double> Dados[NSpecMax+1];  
//-----  
-  
  
__fastcall TAttributos::TAttributos(TComponent* Owner)  
: TFrame(Owner)  
{  
}  
//-----  
-  
  
void __fastcall TAttributos::SpeedButton1Click(TObject *Sender)  
{  
    int NumSpec;  
    NumSpec=StrToInt(NumArq->Value);  
    if(OpenFileDialog->Execute() &&  
!AbreLe(OpenFileDialog->FileName,Dados[NumSpec])){  
        NomeArquivo->Text=OpenFileDialog->FileName;  
        NumPontos->Value=Dados[NumSpec].Length;  
        Carregado->Checked=1;  
        Ativo->Enabled=True;  
        Ativo->Checked=1;  
        FatorDesloca->Enabled=True;  
    }  
}  
  
//-----  
-  
  
int __fastcall TAttributos::AbreLe(AnsiString NomeCompleto,  
DynamicArray<double> &ValoresLidos)  
{  
    double DadoAtual=0.0;  
    long int Ndados=0;  
    FILE *fp;  
    DynamicArray<double> Leitura;  
    int total;  
  
    fp=fopen(NomeCompleto.c_str(),"r");  
    if(fp==NULL)  
        return 1;  
    Leitura.Length=10000000;
```

```
total=Leitura.Length;
while(fscanf(fp,"%lf\n",&Leitura[Ndados])!=EOF&&Ndados<total)
    Ndados++;
if(Ndados==0)
    return 1;
ValoresLidos.Length=Ndados;
ValoresLidos=Leitura.CopyRange(0,Ndados);
fclose(fp);
return 0;
}
```

```
//-----
-----
```

```
#include <stdio.h>

int Compara (const void * a, const void * b){
    double dif;
    dif=*(double*)a - *(double*)b);
    if(dif<0) return(-1);
    if(dif==0) return(0);
    if(dif>0) return(1);
}
void OrdenaDouble(void *Vetor, size_t Tamanho)
{
    qsort(Vetor,Tamanho,sizeof(double),Compara);
}
```

```
#include <stdio.h>

int Compara (const void * a, const void * b){
    double dif;
    dif=(*(double*)a - *(double*)b);
    if(dif<0) return(-1);
    if(dif=0) return(0);
    if(dif>0) return(1);
}

void OrdenaDouble(void *Vetor, size_t NElementos)
{
    qsort(Vetor,NElementos,sizeof(double),Compara);
}
```

```
//-----  
-  
  
#ifndef NomeArqsH  
#define NomeArqsH  
//-----  
-  
#include <Classes.hpp>  
#include <QControls.hpp>  
#include <QStdCtrls.hpp>  
#include <QForms.hpp>  
#include <QButtons.hpp>  
#include <QComCtrls.hpp>  
//-----  
-  
class TNomeAtributos : public TFrame  
{  
    published: // IDE-managed Components  
    TSpeedButton *SpeedButton1;  
    TLabel *Label1;  
    TEdit *NomeRaiz;  
    TLabel *Raiz;  
    TSpinEdit *ValorInicial;  
    TSpinEdit *ValorFinal;  
    TEdit *Caminho;  
    TLabel *Inicial;  
    TLabel *Final;  
    TEdit *Extensao;  
    TLabel *Ext;  
    TButton *Carregar;  
    TCheckBox *AdicionarNumero;  
    TSpinEdit *NZeros;  
    TLabel *Zeros;  
    TCheckBox *AdicionarZeros;  
private: // User declarations  
public: // User declarations  
    __fastcall TNomeAtributos(TComponent* Owner);  
};  
//-----  
-  
extern PACKAGE TNomeAtributos *NomeAtributos;  
//-----  
-  
#endif
```

File: D:\Thiago\Doutorado\Programas\Analise de Tracos\NomeArqs.cpp 18/2/2005
08:37:22

```
//-----  
-  
#include <clx.h>  
#pragma hdrstop  
  
#include "NomeArqs.h"  
//-----  
-  
#pragma package(smart_init)  
#pragma resource "*.xfrm"  
TNomeAtributos *TNomeAtributos;  
//-----  
-  
__fastcall TNomeAtributos::TNomeAtributos(TComponent* Owner)  
: TFrame(Owner)  
{  
}  
//-----  
-
```

II Ajuste dos picos

File: D:\Thiago\Doutorado\Programas\Matlab\EspecAnal\EspectroAnalThi.m 18/9/
2005, 12:52:34

```
function q=EspectroAnalThi(x,y,PrePosTemp,NumSpec);  
%function q=EspectroAnalThi(x,y,PrePos,NumSpec);  
%  
%x -> vetor das frequencias  
%y -> vetor das amplitudes  
%PrePos -> vetor com posicoes aproximadas dos picos  
%NumSpec -> numero do espectro (geralmente o numero do inicio do nome do  
%          arquivo de aquisicao  
  
global Resolucao;  
global PrePos;  
global Atual;  
global NPicosAjustados;  
global Nome2;  
  
NPicosAjustados=0;  
PrePos=PrePosTemp/1000;  
Nlorenz=1;  
R2=input('Nome do arquivo saida =','s');  
Nome2=strcat(R2,'.par');  
  
%verifica se arquivo já existe e atualiza o número de picos em caso  
%afirmativo  
Atual=0;  
NomeTemp=strcat('./',Nome2);  
fpTemp=fopen(NomeTemp);  
if(fpTemp~-=-1)  
    fclose(fpTemp);  
    Temp=load(Nome2,'-ASCII');  
    Atual=size(Temp,1);  
end  
clear NomeTemp fpTemp Temp;  
  
x=x/1000;  
aux=size(x);  
if aux(1)>aux(2)  
    x=x';  
end  
aux=size(y);  
if aux(1)>aux(2)  
    y=y';  
end  
  
Resolucao=x(2)-x(1)  
finicial=x(1)  
%subplot(2,1,1);  
%plot(x,y);  
finictemp=input('Frequencia inicial Grafico (kHz) (Enter para inicio do  
Espectro) = ','s');  
if ~isempty(finictemp) & finictemp<x(length(x))  
    finicial=str2num(finictemp);  
end  
f0=finicial;  
%parfisico=[];  
%save parfisico.par parfisico /ascii;  
clf;  
ini0=find(x>f0-Resolucao & x<f0+Resolucao);  
ixini0=fix(mean(ini0));
```



```
interv=[ixini0 ixini0+1/Resolucao]; %intervalo de 1kHz
%interv=[1 1000];
save inter.dat interv /ascii;
bot=0;

hold off;
while bot~=27
    parr=[];
    %plot(x,y);
    q=osc(x,y);
    lexp=length(q(:,1));
    xexp=q(3:lexp,1)';
    yexp=q(3:lexp,2)';

    fmin=q(1,1);
    fmax=q(2,1);
    deltaf=fmax-fmin;
    ampmax=q(1,2);

    plot(xexp,yexp);
    % largura padrao em kHz
    largpadrao=0.008/Nlorenz;
    alfa=largpadrao/(fmax-fmin);
    alfa=alfa^2/4;
    npontos=length(xexp);
    % estima linha base incial
    ca=(yexp(npontos)-yexp(1))/(xexp(npontos)-xexp(1));
    cl=(yexp(npontos)+yexp(1))/2-(xexp(npontos)+xexp(1))/2*ca;

    %procura posicao das
    lorenztianas*****
    x0=0.5;
    subplot(2,1,1);
    plot(xexp,yexp,'o','MarKerSize',2);
    subplot(2,1,2);
    plot(xexp,log10(yexp),'o','MarKerSize',2);
    title('Botao direito para marcar os primeiros picos e esquerdo para o
    último');
    hold on;

    deltaX=abs((xexp(2)-xexp(1)));
    qr=[];
    bot=0;
    [xo,yo,bot]=ginput(1);
    title('NLOrenz = 1');
    NL=1;
    xnL=xo;
    CCL=yo;
    while bot~=1
        NL=NL+1;
        flaglorenz=1;
        [xo,yo,bot]=ginput(1);
        if isempty(bot)
            NL=NL-1;
            break
        end
        xnL(NL)=xo;
        CCL(NL)=yo;
    end
```

```
subplot(2,1,1);
hold on;
plot(xnL,CCL,'*r','MarkerSize',4);
subplot(2,1,2);
hold on;
plot(xnL,log10(CCL),'*r','MarkerSize',4);
SNL=strcat('NLorenz = ',num2str(NL));
title(SNL);
end
CL=CCL*alfa;
larguraL=alfa*ones(size(CCL));

%Comentado por Thiago
% if (bot==1 | bot==28)
%     flagloren=1;
%     NL=1;
%     xnL=xo;
%     CCL=yo;
%     plot(xnL,CCL,'*r','MarkerSize',4);
%     sound(xnL);
%     NL=length(xnL);
%     CL=CCL*alfa;
%     larguraL=alfa;
%     end;
%
% if (bot==3 | bot==29)
%     flagloren=1;
%     qr=plnr;
%     if isempty(qr)~=1,
%         xnL=qr(:,1);
%         CCL=qr(:,2);
%     end;
%
%     plot(xnL,CCL,'*r','MarkerSize',4);
%     NL=length(xnL);
%     CL=CCL*alfa;
%     larguraL=alfa*ones(size(CCL));
%     NL=length(xnL);
%
%     end;
DL=0.01*CL

%-----fim das lorenztians
hold off;

Nlorenz=NL;

NN=4*Nlorenz;

parr(1:4:NN)=CL;
parr(2:4:NN)=DL;
parr(3:4:NN)=xnL;
parr(4:4:NN)=larguraL;
parr(NN+2)=ca;
parr(NN+1)=cl;
```

```
parsave=parr;  
%end;  
% parametros da funcao basica  
  
hold off;  
  
xt=FunXEsp(parr,xexp,yexp);  
plot(xexp,xt,'y');  
yc=FunXEsp(parr,xexp,yexp);  
plot(xexp,yexp,'ko',xexp,yc,'b','MarkerSize',2);  
title('Inicial');  
  
D=length(parr);  
erroredu=0.5;  
  
% Inicia ajuste  
parr0=parr;  
q=GradEsp(parr,xexp,yexp,200,1e-10,1);  
parG=q(1:D);  
errofim=q(D+1);  
flagerro=q(D+2)  
parr=parG;  
if flagerro==0  
    hold off;  
    load handel;  
    sound(y,Fs);  
    q=LevyEsp(parsave,xexp,yexp,2000,.15);  
    parr=q(1:D);  
    fitini=parr;  
    erroini=q(D+1);  
    parr=fitNLEsp(parr,xexp,yexp);  
  
end;  
  
%pause(2);  
  
hold off;  
xc=0:0.001:1;  
yc=FunXesp(parr,xc,xc);  
  
subplot(2,1,1);  
plot(xexp,yexp,'o',xc,yc,'MarkerSize',2);  
  
subplot(2,1,2);  
plot(xexp,log10(abs(yexp)),'o',xc,log10(abs(yc)),'MarkerSize',2);  
  
% retorna aos valores fsicos  
%parfin=[C' D' xn' largura' B0' B1']  
  
C=parr(1:4:NN);  
D=parr(2:4:NN);  
xn=parr(3:4:NN);  
largura=parr(4:4:NN);  
B0=parr(NN+1);  
B1=parr(NN+2);  
  
xnhz=fmin+(fmax-fmin)*xn;  
x0hz=fmin+(fmax-fmin)*x0;  
larghz=2*sqrt(abs(largura))*(fmax-fmin);
```

```
xx0=x0hz*ones(size(xnhz));

C=C*ampmax*(fmax-fmin)^2;
D=D*ampmax*(fmax-fmin);

bb0=B0*ones(size(xnhz))*ampmax;
bb1=B1*ones(size(xnhz))*ampmax/(fmax-fmin);
parfisc=[C' D' xnhz' larghz' bb0' bb1' xx0'];
parfisc=sortrows(parfisc,3);

title('Grava: Botao Esquerdo, Grava com Ressalvas: Botão Do Meio.
Descarta: direito. Sair: ESC');

[xa,ya,bot]=ginput(1);
if (bot==1 | bot==2)
if(bot==1)
    qualidade=NumSpec;
end
if(bot==2)
    qualidade=0;
end
    sz=size(parfisc);
    fid = fopen(Nome2,'a');
    for i=1:sz(1)
        parbuf=[parfisc(i,:) qualidade fmin fmax deltaf];
        fprintf(fid,'%12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %1.0f
%12.8f %12.8f %12.8f\n',parbuf)
        end;
        fclose(fid);
        NPicosAjustados=NPicosAjustados+sz;
    end;

end;

%*****fim do principal*****

%-----
function q=plnr;

[x,y]=ginput(15);
q=[x y];
plot(x,y,'r*');

%-----ZOOM

function q=zoom(x,y);
%q=zoom(x,y);
% zoom e normaliza para ampmax=1 intervalo de freq. =1;
qrr=[1 1];
while isempty(qrr)~=1,
    [xxz,yyz]=ginput(2);
    qrr=[xxz,yyz]
    if isempty(qrr)~=1,
        xxmax=max(xxz);
        xxmin=min(xxz);
        n=find(x>=xxmin & x<xxmax);
```

File: D:\Thiago\Doutorado\Programas\Matlab\EspecAnal\EspectroAnalThi.m 18/9/
2005, 12:52:34

```
xz=x(n);  
yz=y(n);  
plot(xz,yz);  
title(' Aperte qq tecla se zoom estiver bom ou click de novo para  
selecionar')  
    end;  
end;  
%-----
```

File: D:\Thiago\Doutorado\Programas\Matlab\EspecAnal\fitFunXesp.m 21/2/2005,
17:00:20

```
function err = fitFunXesp(par,x,y,handle)
```

```
Nlorenz=length(par);  
par0=par;  
nlx=length(x);  
FX(1:nlx)=0;  
k=0;
```

```
FX=FunXesp(par,x,y);
```

```
err = norm(FX-y)
```

```
set(gcf,'DoubleBuffer','on');  
set(handle,'ydata',FX)  
drawnow  
%pause(.04)
```

File: D:\Thiago\Doutorado\Programas\Matlab\EspecAnal\fitNLEsp.m 18/2/2005, 1
1:39:00

```
%% Optimal Fit of a Non-linear Function
% This is a demonstration of the optimal fitting of a non-linear function to
a
% set of data. It uses FMINSEARCH, an implementation of the Nelder-Mead
simplex
% (direct search) algorithm, to minimize a nonlinear function of several
% variables.
%
% Copyright 1984-2002 The MathWorks, Inc.
% $Revision: 5.15 $ $Date: 2002/04/02 17:52:33 $

%%
% First, create some sample data and plot it.
function q=fitNLEsp(par,x,y)
plot(x,y,'ro'); hold on; h = plot(x,y,'b'); hold off;
title('Input data');

%type fitFunX

%%
% Make a guess for initial estimate of lambda (start) and invoke FMINSEARCH.
It
% minimizes the error returned from FITFUN by adjusting lambda. It returns
the
% final value of lambda.

start = par;
%[1;0];
options = optimset('TolX',0.1);

q= fminsearch('fitFunXesp',start,options,x,y,h);
```

```
function FX=FunXesp(par,x,y)
% ajuste de Lorentzianas

Npar=length(par);
par0=par;
nlx=length(x);
FX(1:nlx)=0;

x0=0.5;
DX=x-x0;

B0=par0(Npar-1);
B1=par0(Npar);

for i=1:4:Npar-2
    C=(par0(i));
    D=(par0(i+1));
    xn=(par0(i+2));
    alfa=abs(par0(i+3));
    FX=FX+(C+D*DX)./( (x-xn).^2+alfa);
end;
FX=FX+B0+B1*DX;
%FX=FX+B0+(B1+(B2+(B3+(B4+B5*DX).*DX).*DX).*DX).*DX;
```



```
function q=GradEsp(par,xexp,yexp,ni,tol,flagplot);
global pos;
hold off;
clf;
D=length(par);
if flagplot==1,
    cor='k';
    plotador(par,xexp,yexp,cor);
end;
par0=par;
npar=length(par);
npontos=length(xexp);
N1=npar+1;
% global npar npontos xexp;

%yc=FunX(xexp);
E0=erro(par,xexp,yexp);
EX=sum(E0.*E0);
%pause(1);
par1=par;
ERR=EX;

kk=0;
while kk<ni
    % calculo do jacobiano
    kk;
    if flagplot==1,
        cor='r';
        plotador(par,xexp,yexp,cor);
    end;
    for k=1:npar,
        delta=1e-6*par(k);
        if abs(par(k))<1e-12,
            delta=1e-6;
        end;
        parsave=par(k);
        par(k)=par(k)+delta;
        E=erro(par,xexp,yexp);
        par(k)=parsave;

        deltaE=(E-E0)/delta;
        %linha seguinte substitui as comentadas
        J(1:npontos,k)=deltaE(1:npontos);
        %for i=1:npontos,
        % J(i,k)=deltaE(i);
        %end;
    end;

    A=transpose(J)*J;
    C=-transpose(J)*E';

    flagerro=1;
    da=abs(det(A))
    %pause;
    if da<10^(-25),
        flagerro=0;
```

```
    break;
end;

%if rcond(A)<eps
% flagerro=0;
% break;
% end;
if flagerro==1,
    AC=A^(-1)*C;
    par1=par+AC' ;
end;
% acha novos parmetros
L=0+11*(flagerro==0);
SF=2;
flag=0;
while (L<11)

    SF=.5*SF;
    par1=par+SF*AC';

    E0=erro(par1,xexp,yexp);
    EX1=sum(E0.*E0);
    if EX1<=EX,
        flag=1;
        L=11;
        if flagplot==1
            cor='y';
            plotador(par1,xexp,yexp,cor);
        end
    end;
    L=L+1;
end;
par=par1;
[EX EX1]

if EX1<tol,
    break;
end;
if abs(EX-EX1)<1e-9,
    break;
end;
EX=EX1;
kk=kk+1;
end;

[par EX ];
if flagplot==1,
    cor='b';
    plotador(par,xexp,yexp,cor);
    title('fim');

end;
q=[par EX flagerro];

%-----
function plotador(par,xexp,yexp,cor);
global pos;
par;

hold off;
```

File: D:\Thiago\Doutorado\Programas\Matlab\EspecAnal\GradEsp.m 22/2/2005, 14:50:00

```
ycc=FunXesp(par,xexp);  
plot(xexp,yexp,'*g','MarkerSize',2);  
hold on;
```

```
plot(xexp,ycc,cor);  
title('gradiente');
```

```
drawnow;
```

```
%-----
```

```
function er=erro(par,xexp,yexp)  
global pos;  
yc=FunXesp(par,xexp,yexp);
```

```
er=(yc-yexp);
```

File: D:\Thiago\Doutorado\Programas\Matlab\EspecaAnal\Levyesp.m 18/2/2005, 11:39:00

```
function q=LevyLorenz(par,xexp,yexp,ni,tol);
global pos;
format long;
hold off;
D=length(par);

qa=-5;
qv=2;
%qv=2;
r=3721;

%D=2;
d=(D-1)/2+1/(qv-1);
ebom=1e10;
% gera pontos experimentais
npts=length(xexp);

% chutes iniciais

fitbom=par;
fit=fitbom;
fit0=fitbom;
fitex=fitbom;
%erro inicial

xt=FunXesp(fitbom,xexp,yexp);
e2=sum((xt-yexp).*(xt-yexp));

plot(xexp,xt);

T0=100*e2; T=T0;
rr=1;

ebom=1e10;
fit=fit0;
fitex=fit;
xt=FunXesp(fit,xexp,yexp);
e2=sum((xt-yexp).*(xt-yexp));
hold off;
subplot(1,1,1);
plot(xexp,yexp,'o',xexp,xt,'MarkerSize',2);
pause(1);
ebom=e2;
fator=abs(log10(e2)/100+0.05);
fat=fator/20;
T0=100*e2; T=T0;
for i=1:ni,
    %qv=abs(2-(fat*(fat<1.1)+0.9*(fat>1.1)))
    % pause(1);
    fat=fator;
    fator=fat*(fat>0.05)+0.05*(fat<0.05);

    kk=0;
    deltax=(rand-0.5)*fator;
    rr=rand;
    prob=(1+(qv-1)*deltax*deltax/((T^(2/(3-qv))))^(-d);
    for k=1:ni;
```

```
% kk=kk+1;
deltax=(rand-0.5)*fator;
rr=rand;
prob=(1+(qv-1)*deltax*deltax)/((T^(2/(3-qv))))^(-d);
% [i rr prob]
%pause;
if rr<prob
    break
end;
end;
if k>10000
    break
end;

fitA=fit;

Q=fix(i/D);
R=i-Q*D;
R=R+1;
fitA(R)=fitA(R)+deltax;

% calculando novo erro

xt=FunXesp(fitA,xexp,yexp);
e21=sum((xt-yexp).*(xt-yexp));

% Metropolis

de=e21-e2;
if e21<tol,
    fitbom=fitA;
    fitex=fitbom;

    break;
end;
if e21<ebom
    [ebom fator]
    fator=0.98*fator;

    fitbom=fitA;
    fitex=fitbom;
    xt=FunXesp(fitA,xexp,yexp);
    hold off;
    clf;
    subplot(1,1,1);
    plot(xexp,yexp,'o',xexp,xt,'MarkerSize',2);
    title('Levy PENNA');
    drawnow;
    ebom=e21;
    %[i ebom/tol fator]
end;
fitbom=fitex;
fitA=fitex;
[i ebom];
% end;

if (de>=0)
    aux=(1+(qa-1)*de/T);
    if (aux>0)
        aux=aux^(1/(1-qa));
```

```
else aux=0;
end;
if (aux>1)
    aux=1;
end;

penna=aux;
rr=rand;
[e2 e21];

if(rr<penna)
    e2=e21;
    Q=fix(i/D);
    R=i-Q*D;
    R=R+1;
    fitA(R)=fit(R);
end;
else e2==e21;
    Q=fix(i/D);
    R=i-Q*D;
    R=R+1;
    fit(R)=fitA(R);

end;
T=T0*(2^(qv-1)-1)/((1+i)^(qv-1)-1);

e21=ebom;

end;

xt=FunXesp(fitex,xexp,yexp);
hold off;
subplot(1,1,1);
plot(xexp,yexp,'.',xexp,xt);

ebom=sum((xt-yexp).*(xt-yexp))
pause(1);
drawnow;
e2=ebom;
fit0=fitex;

q=[fitex ebom];

% gradiente
%q=GradLevy(fitbom,xexp,yexp,15,1e-5);
```

```
function q=osc(x,y)

global Resolucao;
global PrePos;
global Atual;
global Nome2;
flag=1;

rangepico=0.600/Resolucao; %intervalo a ser mostrado em numero de pontos
200Hz
range=fix(1/Resolucao); %intervalo a ser mostrado em numero de pontos 1kHz
ideltax=fix(range);

lx=length(x)
load inter.dat;
ixini0=fix(inter(1));
ixfim0=fix(inter(2))
ixini0MAX=fix(lx-range);
subplot(1,1,1);
xmax=x(ixfim0);
xmin=x(ixini0);
ymin=min(y);
ymax=max(y);

subplot(2,1,1);
hold on;
plot(x(ixini0:ixfim0),y(ixini0:ixfim0));
xmed=(xmin+xmax)/2;
PicPlota=PrePos(find(PrePos>=xmin & PrePos <=xmax))';
if ~isempty(PicPlota)
    plot([1;1]*PicPlota,[ymin ymax],'r');
end
if Atual>0 & Atual <=length(PrePos)
    fAtual=PrePos(Atual);
else
    fAtual=xmed;
end
plot([fAtual;fAtual],[ymin ymax],'y','LineWidth',2)
axis([x(ixini0) x(ixfim0) min(y(ixini0:ixfim0)) max(y(ixini0:ixfim0))]);
hold off;

subplot(2,1,2);
plot(x(ixini0:ixfim0),log10(y(ixini0:ixfim0)));
axis([x(ixini0) x(ixfim0) log10(min(y(ixini0:ixfim0)))
log10(max(y(ixini0:ixfim0)))]);
leg=strcat('Pico Atual: ',num2str(Atual),'      Freq. Atual:',num2str(xmed));
title(leg);

ixini=ixini0;
ixfim=ixfim0;

bot=1;
botb=1;
gro=[1,1];
while (fix(bot)~=2)
    figure(1);
    [xo,yo,bot]=ginput(1);
    gro=[xo,yo];
    flagloren=0;
    if isempty(bot)
```

```
    bot=2;
end
switch bot

    case 2 %sai do loop Modo Osciloscopio
        flagloren=1;
        break;

    case 30 %desloca espectro para tras: (seta para cima)
        flagloren=1;
        ixfim=ixfim0-ideltax;
        ixini=ixini0-ideltax;
        figure(1);

    case 31 %desloca espectro para frente: (seta para baixo)
        flagloren=1;
        ixfim=ixfim0+ideltax;
        ixini=ixini0+ideltax;
        figure(1);

    case 48 %coloca ZEROS no arquivo de parametros: 0
        flagloren=1;
        fid = fopen(Nome2,'a');
        parbuf=[0 0 0 0 0 0 0 0 0 0 0]
        fprintf(fid,'%12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %12.8f %1.0f\n',parbuf);
        fclose(fid);

    case 102 %altera frequencia central: f
        Centro=input('Valor da Freq. Central (Enter para Sair):');
        if ~isempty(Centro)
            flagloren=1;
            Centro=fix(mean(find(x>Centro-Resolucao & x<Centro+Resolucao)));
            ixfim=Centro+ideltax;
            ixini=Centro-ideltax;
            figure(1);
        end

    case {3, 29} %pico anterior: botao direito ou ->
        if (Atual>=2 & ~isempty(PrePos))
            flagloren=1;
            Atual=Atual-1;
            Temp=AchaIndicesPico(x,range(pico));
            ixini=Temp(1);
            ixfim=Temp(2);
            figure(1);
        end;

    case {1, 28} % proximo pico: botao esquerdo ou <-
        if (Atual<=length(PrePos)-1 & ~isempty(PrePos))
            flagloren=1;
            Atual=Atual+1;
            Temp=AchaIndicesPico(x,range(pico));
            ixini=Temp(1);
            ixfim=Temp(2);
            botb=1;
            figure(1);
        end;
```



```
case 112 %vai para a posicao do pico digitado: p
    NumPico=input('Numero do pico a ser plotado (Enter para Sair): ');
    if (~isempty(NumPico) & ~isempty(PrePos) & NumPico<=length(PrePos)-1 &
NumPico>0)
        flagloren=1;
        Atual=NumPico;
        Temp=AchaIndicesPico(x,range(pico);
        ixini=Temp(1)
        ixfim=Temp(2)
        figure(1);

        end

    end;
    figure(1);
    if flagloren==1
        if ixini<1
            figure(1);
            ixini=1;
            ixfim=ixini+range;
        elseif ixini>ixiniOMAX;
            figure(1);
            ixini=ixiniOMAX;
        end;

        if (ixfim>length(x) )
            ixfim=length(x);
        end;

    else ixfim=ixfim0;
        ixini=ixini0;
    end;
    figure(1);
    ixfim=fix(ixfim);
    ixini=fix(ixini);
    xmax=x(ixfim);
    xmin=x(ixini);

    figure(1);
    subplot(2,1,1);
    plot(x(ixini:ixfim),y(ixini:ixfim));
    hold on;
    xmed=(xmin+xmax)/2;

    PicPlota=PrePos(find(PrePos>=xmin & PrePos <=xmax))';
    if ~isempty(PicPlota)
        plot([1,1]*PicPlota,[ymin ymax],'r');
    end
    fAtual=PrePos(Atual);
    plot([fAtual,fAtual],[ymin ymax],'y','LineWidth',2)

    axis([x(ixini) x(ixfim) min(y(ixini:ixfim)) max(y(ixini:ixfim))]);
    figure(1);
    hold off;

    subplot(2,1,2);
    plot(x(ixini:ixfim),log10(y(ixini:ixfim)));
    axis([x(ixini) x(ixfim) log10(min(y(ixini:ixfim)))
log10(max(y(ixini:ixfim)))]);
```

Page: 4

```
        title(' Aperte qq tecla se zoom estiver bom ou click de novo para  
selecionar')  
        end;  
        clear xxz;  
        clear yyz;  
    end;  
  
    subplot(2,1,1);  
    plot(xz,yz,'o');  
  
    subplot(2,1,2);  
    plot(xz,log10(yz),'g');  
  
    fmax=xz(length(xz));  
    fmin=xz(1);  
    xz=(xz-fmin)/(fmax-fmin);  
    yz=yz/max(yz);  
    xz=[fmin fmax xz];  
    yz=[max(yz) max(yz) yz];  
    q=[xz' yz'];  
  
function q=AchaIndicesPico(x, intervalo)  
    global Atual;  
    global PrePos;  
    global Resolucao;  
    Centro=find(x>PrePos(Atual)-Resolucao & x<PrePos(Atual)+Resolucao);  
    Centro=fix(mean(Centro));  
    ixfim=Centro+intervalo/2;  
    ixini=Centro-intervalo/2;  
    if(ixini<1)  
        ixini=1;  
    end  
    if(ixfim>length(x))  
        ixfim=length(x)  
    end  
    q=[ixini ixfim];
```

```
        title(' Aperte qq tecla se zoom estiver bom ou click de novo para  
selecionar')  
    end;  
    clear xxz;  
    clear yyz;  
end;  
  
subplot(2,1,1);  
plot(xz,yz,'o');  
  
subplot(2,1,2);  
plot(xz,log10(yz),'g');  
  
fmax=xz(length(xz));  
fmin=xz(1);  
xz=(xz-fmin)/(fmax-fmin);  
yz=yz/max(yz);  
xz=[fmin fmax xz];  
yz=[max(yz) max(yz) yz];  
q=[xz' yz'];  
  
function q=AchaIndicesPico(x, intervalo)  
    global Atual;  
    global PrePos;  
    global Resolucao;  
    Centro=find(x>PrePos(Atual)-Resolucao & x<PrePos(Atual)+Resolucao);  
    Centro=fix(mean(Centro));  
    ixfim=Centro+intervalo/2;  
    ixini=Centro-intervalo/2;  
    if(ixini<1)  
        ixini=1;  
    end  
    if(ixfim>length(x))  
        ixfim=length(x)  
    end  
    q=[ixini ixfim];
```

III Método de *cut-off*

```
function q=cutoff(x)
%entra:
%   espectro desdobrado
%retorna:
%   Flutuação que foi retirada com o método
%   Flutuação do espectro após aplicação do método
%   Espectro corrigido

a=size(x);
if a(1)<a(2)
    x=x';
end
flut=x-[1:length(x)]';
y=fftshift(fft(flut));
yreal=real(y);
yplot=yreal(floor(length(yreal)/2)+1:length(yreal));
plot(yplot.^2);
title('Clique o Mouse na Posicao de Corte (Botao Direito para Zoom)');
[pos a bot]=ginput(1);
max=length(yplot);
while bot ~= 1
    if pos>max
        pos=max;
    end
    if pos>1
        plot(yplot(1:floor(pos)).^2)
    end
    [pos a bot]=ginput(1);
end
if pos>1 && bot==1
    corte=floor(pos)
    indice_corte=corte+floor(length(y)/2);
    nzeros=length(yreal)-indice_corte;
    y(1:nzeros)=0;
    y(indice_corte+1:length(y))=0;
end
flutcortada=real(ifft(ifftshift(y)));
flutcorrigida=(flut-flutcortada);
Spec=sort(flutcorrigida+[1:length(y)]');
%flutcorrigida=Spec-[1:length(y)]';
close all;

%retorna:
%   Flutuação que foi retirada com o método
%   Flutuação do espectro após aplicação do método
%   Espectro corrigido
q=[Spec flutcorrigida flutcortada];
corte
```

IV Energia e calor específico de espectros experimentais


```

Emin=Emin1+dE*(rand)/2;
%Emin=Emin1%+dE*(rand)/2;
ni=ni+1; % indice da linha do arquivo de saida
m=0; % m = indice de cada janela
U=0; % U: energia media com potencial em cada varredura
Ut=0; % Ut: energia media sem potencial em cada varredura
for E=Emin+1:dE/sobrepdE:(Emax-dE) % faz a varredura do espectro com
janelas de largura dE
    m=m+1; % m = indice de cada janela
    Ejmin=E; % inicio da janela
    Ejmax=E+dE; % final da janela
    q=find(desd >= Ejmin & desid <= Ejmax); % procura os niveis
dentro de cada janela
    Imin=min(q);
    Imax=max(q);
    N2(m,ni)=max(q)-min(q)+1; % N2 :numero de niveis da janela m
    % Calcula as duas somatorias:
    BW=0; % Soma W: dupla somatoria do ln subtraido da cerca
    %Dupla somatizria do ln iz calculada nas proximas 2 linhas

Matriz=tril(Dist(Imin:Imax-1,1:Imax-Imin))+triu(ones(Imax-Imin),1);
    SomaW=sum(sum(log(Matriz)));
    %%% Copio os niveis das janelas do espectro e da cerca para os
vetores desid2 e Cerc2
    % cria um 'desd2' com os niveis da janela
    desid2=desd(Imin:Imax);
    Cerc2=Cerc(Imin:Imax);

    L=dE/2; % define o intervalo de energia 2L da janela
    % coloca a origem no meio das janelas
    desid2=desid2-(Ejmin+L);
    Cerc2=Cerc2-(Ejmin+L);

    mx=(L+desd2);% redefinicao de variaveis do espectro
    nx=(L-desd2);
    mc=abs(L+Cerc2);% redefinicao de variaveis da cerca
    nc=abs(L-Cerc2);
    BQ=sum(nx.*(1-log(nx/dE)) + mx.*(1-log(mx/dE))); % Somatoria do
potencial do espectro
    Bc=sum(nc.*(1-log(nc/dE)) + mc.*(1-log(mc/dE))); % Somatoria do
potencial da cerca

    B=BQ-Bc; % Somatoria do potencial total
    WQ(m,ni)=-(SomaW+B)/N2(m,ni); % Energia da janela m COM
potencial
    W(m,ni)=-SomaW/N2(m,ni); % Energia da janela m SEM potencial
    U=((m-1)/m)*U+(WQ(m,ni)/m); % U: energia media COM potencial (m
janelas) em
cada varredura
    Ut=((m-1)/m)*Ut+(W(m,ni)/m); % Ut: energia media SEM potencial
(m janelas) em cada varredura
    WQ2(m,ni)=WQ(m,ni)*WQ(m,ni); % P/ calcular a varian?a COM
potencial
    W2(m,ni)=W(m,ni)*W(m,ni); % P/ calcular a varian?a SEM potencial
end
No=m; % numero total de janelas

if ~isempty(RaizArq) & ~isempty(SalvParc)
nomearq=strcat(RaizArq,'/dE',num2str(dE),'/',RaizArq,'_dE',num2str(dE),'_Rep

```

```
Num',num2str(Nvezes),'.dat');  
    varGrava=WQ(1:No,ni);  
    save(nomearq,'varGrava','-ASCII');  
end  
  
    % Calcula a varian?a:  
    SomaVQ=sum(WQ2(1:No,ni)); % Somatoria da varian?a COM potencial  
    SomaV=sum(W2(1:No,ni)); % Somatoria da varian?a SEM potencial  
    VQ=(SomaVQ/No)-(U*U); % Varian?a COM potencial  
    V=(SomaV/No)-(Ut*Ut); % Varian?a SEM potencial  
    % Calcula N2med: o numero medio de niveis das janelas de uma  
varredura ni  
    SomaN=sum(N2(1:No,ni));  
    N2med(ni,1)=SomaN/No; % N2med: o numero medio de niveis das janelas  
de uma varredura ni  
    CQ=VQ*N2med(ni,1); % Calor especifico COM potencial de uma varredura  
ni  
    C=V*N2med(ni,1); % Calor especifico SEM potencial de uma varredura  
ni  
    % Saida:  
    sai(ni,1)=dE; % N2med: o numero medio de niveis das janelas da  
varredura ni  
    sai(ni,2)=N2med(ni,1); % N2med: o numero medio de niveis das janelas  
da varredura ni  
    sai(ni,3)=No; % No: numero total de janelas da varredura ni  
    sai(ni,4)=Ut; % Ut: energia media SEM potencial da varredura ni  
    sai(ni,5)=C; % Calor especifico da varredura ni SEM potencial  
    sai(ni,6)=U; % U: energia media COM potencial da varredura ni  
    sai(ni,7)=CQ; % Calor especifico da varredura ni COM potencial  
end  
  
if ~isempty(RaizArq) & ~isempty(SalvParc)  
    nomearq=strcat(RaizArq,'/',RaizArq,'_dE',num2str(dE),'.dat');  
    save(nomearq,'sai','-ASCII');  
end  
  
    Media=mean(sai);  
    Desvio=std(sai);  
    TermoMedia(NdE,:)= [dE Media(4) Desvio(4) Media(5) Desvio(5) Media(6)  
Desvio(6) Media(7) Desvio(7)];  
    if ~isempty(RaizArq)  
        NomeFinal=strcat(RaizArq,'.term');  
        save(NomeFinal,'TermoMedia','-ASCII');  
    end  
    NdE=NdE+1;  
    toc  
end  
toc
```

Apêndice B

Cópia do artigo publicado.

Universality of rescaled curvature distributions

M. P. Pato,¹ K. Schaadt,^{2,3} A. P. B. Tufaile,¹ C. Ellegaard,² T. N. Nogueira,¹ and J. C. Sartorelli¹¹*Instituto de Física, Universidade de São Paulo, C.P. 66318, 05315-970 São Paulo, S.P., Brazil*²*Complexity Lab, Niels Bohr Institute, Blegdamsvej 17, 2100 Copenhagen Ø, Denmark*³*CORE A/S, Lersøparkalle 42, 2100 Copenhagen Ø, Denmark*

(Received 28 September 2004; revised manuscript received 23 November 2004; published 18 March 2005)

We applied a recently proposed rescaling of curvatures of eigenvalues of parameter-dependent random matrices to experimental data from acoustic systems and to a theoretical result. It is found that the data from four different experiments, ranging from isotropic plates to anisotropic three-dimensional objects, and the theoretical result always agree with the universal curvature distribution, if only the curvatures are rescaled such that the average of their absolute values is unity.

DOI: 10.1103/PhysRevE.71.037201

PACS number(s): 05.45.Mt, 05.30.Ch, 62.30.+d

I. INTRODUCTION

The usefulness of the study of statistical properties of eigenvalues and eigenvectors of quantum systems has been demonstrated in many areas of physics [1]. It has become clear that these properties follow universal patterns that can be modeled by probability distributions extracted from ensembles of random Hamiltonians [2] of the same class of the system under study. In particular, time-reversal invariant physical systems whose underlying classical motion is chaotic have correlated spectra with statistical properties consistent with the predictions of the Gaussian orthogonal ensemble (GOE) random matrix model. This result has been confirmed by analyses numerically performed for two degrees of freedom systems (one particle in a two-dimensional box), but also by experimentally studying resonances of microwave cavities with adequate boundaries or of quartz or metallic blocks (elastodynamics) (see, for instance, contributions by Richter [3], Stöckmann [4], Ellegaard [5], and Schaadt [6]).

In a pioneering work Wilkinson started the investigation of the dynamical aspect of the same problem, namely, how the correlations manifest themselves in the flow of the levels when an external parameter is varied [7]. A random matrix model has then been constructed to investigate the statistical properties of the first (velocities) and the second (curvatures) derivatives of GOE eigenvalues with respect to an adequately defined parameter. It has been found that the velocities are Gaussian distributed while the curvatures, after an appropriate rescaling, follow the simple distribution given by [8,9]

$$P(k) = \frac{1}{2(1+k^2)^{3/2}}. \quad (1)$$

Large curvatures are produced by close encounters and they probe repulsion between neighboring pair of levels. A simple estimation [10] shows that the power decay $|k|^{-3}$ predicted for large curvatures by the above equation follows from the linear behavior of the repulsion between GOE eigenvalues. Another feature of this distribution is that the curvatures were normalized in such a way that the average of their modulus is equal to one, i.e., $\langle|k|\rangle=1$ with the quantity

$\langle|k|\rangle$ playing the same role the average density plays in the usual level statistics. One and the other give in the respective dynamical and static cases, respectively, the scaling necessary to be able to talk about universality.

The difficult task of checking experimentally this prediction was undertaken by the experimental group at the Center for Chaos and Turbulence (CAT) of the Niels Bohr Institute. They studied the dependence on different external parameters of the spectrum of frequencies of four systems ranging from isotropic plates to anisotropic three-dimensional objects. The statistical properties for fixed value of the parameters were verified to follow the GOE statistics. The measured curvature distributions have shown slight but significant deviations from the above theoretical prediction.

By analyzing the data relative to a quartz block in which the temperature was the parameter [11], it has been shown however in Ref. [10] that the agreement with the universal prediction is remarkable if one rescales the curvature such that it satisfies the requirement that their average is equal to one, as is implied by Eq. (1). We extend here this procedure to the others systems studied by the CAT group. We show that rescaling the data the results of all four experiments agree remarkably with the universal curvature distribution.

The same procedure is also successfully applied to a theoretical curvature data obtained with the so-called Robnik

TABLE I. Overview of the parameters that govern each of the experiments. The first column specifies the system under study. The following columns contain the quality factor value of the system at a typical frequency, the typical level number, the number of levels included in the study, the number of parameters values, the ratio of the resonance width to the mean level spacing, the range covered on the normalized parameter axis, the ratio of the typical wavelength to the size of the system, and the number of independent points, roughly, in frequency-parameter space for calculation of statistical quantities like the curvature distribution.

System	Q	N	n_e	n_x	Γ/Δ	$x_{\max}$	L/λ	$n_e(x+1)$
(a)	10^4	2×10^2	171	63	30	3.3	5	735
(b)	10^4	3×10^2	275	63	30	2.0	7	825
(c)	10^5	1×10^3	466	59	25	0.8	6	839
(d)	10^5	3×10^3	709	101	15	1.8	7	1276

FIG. 1. Unfolded eigenfrequencies of the quartz block plotted as function of the mass removed from one surface of the block with polishing powder.

billiard (a quadratic conformal map of the unit disk) by varying its shape parameter [12]. The complete disagreement with Eq. (1) shown by the initial data has been reduced in Ref. [13] by the introduction of an appropriate unfolding procedure of the parameter. We show that the agreement becomes better if the curvatures are rescaled.

II. EXPERIMENTAL SETUP

Using an HP 3589A spectrum / network analyzer and piezoelectric transducers, we measure ultrasound transmission spectra of solid objects. There are three such transducers, of which one is transmitter and two others are receivers. The temperature of the system is kept constant to within 0.005 °C using a temperature controller, such that the eigenfrequencies are not affected by the fluctuations in room temperature. The pressure of the air surrounding the sample can be controlled and kept at a low value where air damping of the vibration is insignificant. During measurements, the system is resting only on three tiny spikes, making the vibrations as close to free as possible. For a more detailed description of this setup see Ref. [14].

III. EXPERIMENTS

Parametric experiments were carried out on four different systems. The first experiment (a) was carried out on an aluminum plate cut in the shape of a quarter stadium with a slit cut on one face to break the mirror symmetry through the middle plane of the plate. This ensures that flexural and in-plane modes are completely mixed, see [15] for details. The parameter was the size of the straight line section in the stadium. The second experiment (b) was carried out on an aluminum plate cut in the shape of one quarter of the Sinai stadium, but this time no slit was cut to mix modes. Instead, the two mode classes were separated experimentally using the air pressure, and the flexural modes were selected for

analysis. The eigenfrequencies were perturbed by gradually cutting off material from the plate [16]. The third experiment (c) was carried out on a three-dimensional (3D) system, a block of monocrystalline quartz cut in the shape of one eighth of a 3D Sinai billiard [17]. The block was carefully polished with fine grinding powder in order to perturb the eigenfrequencies by removing mass. The fourth experiment (d) was carried out on similar block, but this time the temperature was used as the external parameter. In Table I the most important information about the experiments is given. Finally, the fifth system (e) was a theoretical result from Robnik billiard performed by Leboeuf and Sieber [13].

IV. FITTING THE SPAGHETTI

The eigenfrequencies of measured resonance curves are determined using an interactive software tool that we have developed in the programming language IDL. The user points out a resonance on the screen, and the software fits the data with a "skew Lorentzian," see [18] for details, whose parameters include the eigenfrequency and the width. Once the eigenfrequencies are determined, the unfolding procedure is performed by fitting their cumulative density with a third order polynomial from which the new spectrum with a unitary average density is extracted.

In Fig. 1 we plot the unfolded frequencies as function of the mass removed. As one can see, the "spaghetti strings" corresponding to raw data contain noise. Since we are interested in slopes and curvatures, we must deal with this noise. We find that some standard noise reduction schemes, like smoothing, fitting and low pass filtering, cannot do a sufficiently good job. In our case, however, the following two points are bound to complicate any attempt to reduce noise: For some level, say ϵ , no prediction for the function $\epsilon_i(T)$ exists, i.e., we cannot just fit some function to the spaghetti string. In fact, the level motion is known to be very complicated. We know that the spaghetti strings have curvatures on all scales. In particular since we are interested in the tail of the curvature distribution, it is of great importance that the large curvatures are not altered by the noise reduction. We decided to use the algorithm explained in the following. Consider a point on some spaghetti string $\epsilon_i(T)$, corresponding to the parameter value $T=T'$. Fit a polynomial to the $2n+1$ points that include the point in question and the n neighboring points on both sides. If the polynomial is $P(T) = aT^2 + bT + c$, the slope and curvature of the spaghetti string at the considered point is simply $2aT' + b$ and $2a$, respectively. The parameter n is a constant of the algorithm and must be determined such that $2n+1$ is larger than the typical size of the noise but still small enough to satisfy locality. An improved version was introduced, where n is determined by the curvature in a converging procedure. The idea is that for a point on the spaghetti string where the curvature is large, one should use a small n to make sure that the polynomial really follows the sharp curve of the spaghetti string, whereas for a point of small curvature one should use a larger n to avoid fitting the noise. The algorithm starts out setting $n = N$ where N is an intermediate value and then fits $P(T) = aT^2 + bT + c$ to the $2n+1$ points in question. The resulting

FIG. 2. Curvature distributions for the five systems. In the first column the curves correspond to Eq. (1); in the second column they are the fitted curves using Eq. (4).

curvature $2a$ is translated into a new value of $n=n(2a)$ according to the above mentioned idea. This step is continued until the resulting curvature stays the same, i.e., the value of the curvature has converged to some number. This algorithm is used for every point on every spaghetti string and the resulting slopes and curvatures are stored for analysis. Tests show that the results are independent of the choice of N , which is reassuring.

V. CURVATURES

The normalization of the curvatures start by replacing the physical parameter T by a new dimensionless parameter t such that [13]

TABLE II. The fitted parameters from the method of least squares for the five systems corresponding to the rescaled curves in the second column in Fig. 2.

System	γ	χ^2
Stadium plate-slit cut, (a)	1.23	2.1×10^{-5}
Sinai-stadium plate, (b)	0.95	6.9×10^{-5}
3D Sinai billiard-mass, (c)	1.18	1.5×10^{-5}
3D Sinai billiard-temperature, (d)	1.30	9.3×10^{-6}
Robnik billiard, (e)	1.42	5.3×10^{-4}

$$\frac{dt}{dT} = \sqrt{\left\langle \left(\frac{d\epsilon}{dT} \right)^2 \right\rangle}, \quad (2)$$

where the average is made over the whole set of unfolded eigenfrequencies at a given value of T . The level curvature is then defined in terms of these new scaled variables as

$$k = \frac{1}{\pi} \frac{d^2 \epsilon}{dT^2} = \frac{1}{\pi \langle \epsilon^2 \rangle} \left(\ddot{\epsilon} - \frac{\langle \dot{\epsilon} \dot{\epsilon} \rangle}{\langle \epsilon^2 \rangle} \dot{\epsilon} \right), \quad (3)$$

where the dots denote derivatives with respect to the physical parameter T . In Eq. (3), $1/\pi$ is exactly the factor that normalizes the curvatures in the context of random matrix theory. The experimental data were fitted with the one parameter distribution

$$P(k) = \frac{\gamma^2}{2(\gamma^2 + k^2)^{3/2}} \quad (4)$$

with $\gamma = \langle |k| \rangle$. The results of the fitting for the five systems are shown in Fig. 2 and in Table II. The small values of chi-square show that the agreement with the data is not only visual. We do not observe any correlation between the values of γ in Table II and the material of the samples or the external parameter used.

VI. DISCUSSION AND CONCLUSION

In conclusion, we have shown that the rescaling introduced in Ref. [10] makes experimental and theoretical data of level curvatures to agree with the universal distribution Eq. (1). We draw from these results that random matrix theory correctly predicts the functional form of the curvature distribution but the curvature normalization factor. This factor has to be extracted from the data. We remark that this feature has also been observed in the case of another parametric distribution, the level correlation function, whose universality has been experimentally studied using quantum dots [19].

A clue to understand why the random matrix normalization factor $1/\pi$ is not sufficient is maybe provided by the fact that in all the present systems the average $\langle (d\epsilon/dT)^2 \rangle$ defining the new unfolded parameter is not a constant during the evolution. As a consequence, in the definition of the scaled curvature Eq. (3) the first derivative is subtracted from the second derivative.

Before concluding, we add that authors have attributed discrepancies with the universal curve to the presence of

symmetries [20,21]. Without ruling out completely this possibility we argue that symmetries should reduce the repulsion between levels and, as a consequence, make their trajectories closer to straight lines increasing and not reducing small curvatures as observed in the systems being here analyzed.

ACKNOWLEDGMENTS

A. Heine and P. Leboeuf are acknowledged for fruitful discussions. This work was supported in part by the Brazilian agencies FAPESP and CNPq.

-
- [1] T. Guhr, A. Müller-Groeling, and H. A. Weidenmüller, Phys. Rep. **299**, 189 (1998).
 - [2] M. L. Mehta, *Random Matrices* (Academic, Boston, 1991).
 - [3] A. Richter, Phys. Scr., T **T90**, 212, (2001).
 - [4] H. J. Stöckmann, Phys. Scr., T **T90**, 246 (2001).
 - [5] C. Ellegaard, K. Schaadt, and P. Bertelsen, Phys. Scr., T **T90**, 223 (2001).
 - [6] K. Schaadt, A. P. B. Tufaile, C. Ellegaard, Phys. Rev. E **67**, 026213 (2003).
 - [7] M. Wilkinson, J. Phys. A **21**, 4021 (1988).
 - [8] J. Zakrzewski and D. Delande, Phys. Rev. E **47**, 1650 (1993).
 - [9] F. von Oppen, Phys. Rev. E **51**, 2647 (1995).
 - [10] M. S. Hussein, C. P. Malta, M. P. Pato, and A. P. B. Tufaile, Phys. Rev. E **65**, 057203 (2002).
 - [11] P. Bertelsen, C. Ellegaard, T. Guhr, M. Oxborrow, and K. Schaadt, Phys. Rev. Lett. **83**, 2171 (1999).
 - [12] B. Li and M. Robnik, J. Phys. A **29**, 4387 (1996).
 - [13] P. Leboeuf and M. Sieber, Phys. Rev. E **60**, 3969 (1999).
 - [14] K. Schaadt, M.Sc. thesis, Niels Bohr Institute, Copenhagen, 1997. <http://www.nbidk/~schaadt/speciale/speciale.ps.gz>
 - [15] P. Bertelsen, C. Ellegaard, and E. Hugues, Eur. Phys. J. B **15**, 87 (2000).
 - [16] K. Schaadt and A. Kudrolli, Phys. Rev. E **60**, R3479 (1999).
 - [17] C. Ellegaard, T. Guhr, K. Lindemann, J. Nygård, and M. Oxborrow, Phys. Rev. Lett. **77**, 4918 (1996).
 - [18] H. Alt, P. von Brentano, H.-D. Gräf, R.-D. Herzberg, M. Philipp, A. Richter, and P. Schardt, Nucl. Phys. A **560**, 293 (1993).
 - [19] U. Sivan, Y. Imry, and A. Aronov, Europhys. Lett. **28**, 115 (1994).
 - [20] G. Ergun and Y. V. Fyodorov, Phys. Rev. E **68**, 046124 (2003).
 - [21] I. V. Yurkevich and V. E. Kravtsov, Phys. Rev. Lett. **78**, 701 (1997).

Bibliografia

- [1] R. L. Weaver, "Spectral statistics in elastodynamics," *J. Acoust. Soc. Am.*, vol. 85, no. 3, p. 1005, 1989.
- [2] C. Ellegaard, T. Guhr, K. Lindemann, H. Q. Lorensen, J. Nygard, and M. Oxborrow, "Spectral statistics of acoustic resonances in aluminum blocks," *Physical Review Letters*, vol. 75, no. 8, pp. 1546 – 1549, 1995.
- [3] K. Schaadt, "The Quantum Chaology of Acoustic Resonators," Master's thesis, Instituto Niels Bohr, Copenhagen, 1997.
- [4] V. I. Arnold, *Mathematical Methods of Classical Dynamics*. Springer, 1978.
- [5] A. N. Kolmogorov, *Dokl. Akad. Nauk*, vol. 98, 1954.
- [6] V. I. Arnold, "Small denominators and problems of stability of motion in classical and celestial mechanics (russian)," *Russ. Math. Surv.*, vol. 18, p. 85, 1963.
- [7] J. Moser, *Akad. Wiss. Göttinger*, vol. 1, 1962.
- [8] M. Berry, *Regular and irregular motion*, em "Lês Houches (1981) Session XXXVI". North-Holland Publ. Comp., 1983.
- [9] Y. G. Sinai, *Russ. Math. Surv.*, vol. 25, 1970.
- [10] O. P. J. L. Lebowitz, "Modern ergodic theory," *PHYSICS TODAY*, vol. 26 (2), pp. 23–29, 1973.
- [11] M. V. B. P. J. Richens, "Pseudointegrable systems in classical and quantum-mechanics," *PHYSICA D*, vol. 2 (3), pp. 495–512, 1981.
- [12] E. P. Wigner, *Phys. Rev.*, vol. 40, p. 1032.
- [13] F. Haake, *Quantum Signatures of Chaos*. Springer-Verlag, Nova Iorque, 1992.
- [14] M. L. Mehta, *Random Matrices and the Statistical Theory of Energy Levels*. Academic Press, Nova Iorque, 1967.
- [15] N. C. Snaith, "dissertação de mestrado: "random matrix theory and zeta functions"." Ph.D. dissertation, Univ. of Bristol, Bristol, 2000.
- [16] F. J. Dyson and M. L. Mehta, "Statistical theory of the energy levels

- of complex systems iv," *Journal of Mathematical Physics*, vol. 4, no. 5, p. 701, 1963.
- [17] O. BOHIGAS and M. GIANNONI, "Chaotic motion and random matrix theories." *LECTURE NOTES IN PHYSICS*, vol. 209, pp. 1-99, 1984.
- [18] O. Bohigas, *Random matrices and chaotic dynamics*, em "*Lés Houches (1989) Session LII*", North-Holland, 1991.
- [19] E. L. D. Landau, *Theory of Elasticity*, 3rd ed., Oxford: Pergamon Press, 1986.
- [20] K. F. Graff, *Wave motion in elastic solids*. New York: Dover, 1991.
- [21] J. D. Achenbach, *Wave Propagation in Elastic Solids*. AMERICAN ELSEVIER PUBLISHING COMPANY, INC. - NEW YORK, 1973.
- [22] C. E. E. Hugues, P. Bertelsen, "Distribution of eigenfrequencies for vibrating plates," *EUROPEAN PHYSICAL JOURNAL B*, vol. 15 (1), pp. 87-96, 2000.
- [23] K. Schaadt, "Experiments on Acoustic Chaology and Statistical Elastodynamics," Ph.D. dissertation, University of Copenhagen, 2001.
- [24] R. Balian and C. Bloch, "Distribution of Eigenfrequencies for the Wave Equation in a Finite Domain - I. Three-Dimensional Problem with Smooth Boundary Surface," *Annals Of Physics*, vol. 60, p. 401, 1970.
- [25] P. Bertelsen, "Quantum chaos and vibration of elastic plates," Master's thesis, Instituto Niels Bohr, Copenhagen, 1997.
- [26] R. W. Daniels, *An Introduction to Numerical Methods and Optimization Techiniques*. North Holland, New York, 1978.
- [27] H. Alt, P. von Brentano, H. -D. Gräf, R. -D. Herzberg, M. Philipp, A. Richter, and P. Schardt, "Resonances of a superconducting microwave cavity - a test of the breit-wigner formula over a large dynamic-range," *Nuclear Physics A*, vol. 560 (1), no. 1, pp. 293-305, 1993.
- [28] E. P. Wigner, "Resonance reactions and anomalous scattering," *PHYSICAL REVIEW*, vol. 70, no. 1-2, pp. 15-33, 1946.
- [29] G. Breit, "Schematic treatment of nuclear resonances," *PHYSICAL REVIEW*, vol. 69, no. 9-10, p. 472, 1946.

- [30] K. Schaadt, "Comunicação privada."
- [31] E. Bogolmony and E. Hugues, "Semiclassical theory of flexural vibrations os plates," *Phys. Rev E*, vol. 57 (5), p. 5404, 1998.
- [32] H.-D. Gräf, H. L. Harney, H. Lengeler, C. H. Lewenkopf, C. Rangacharyulu, A. Richter, P. Schardt, and H. A. Weidenmüller, "Distribution of eigenmodes in a superconducting stadium billiard with chaotic dynamics," *Physical Review Letters*, vol. 69, no. 9, p. 1296, 1992.
- [33] J. W. Cooley and O. W. Tukey, "An algorithm for the machine calculation of complex fourier series." *Math. Comput.*, vol. 19, pp. 297–301, 1965.
- [34] E. Chu and A. George, *Inside the FFT Black Box: Serial and Parallel Fast Fourier Transform Algorithms*. Boca Raton, FL: CRC Press, 2000.
- [35] M. Wilkinson, "Statistical aspects of dissipation by landau-zener transitions," *J. Phys. A*, vol. 21, p. 4021, 1988.
- [36] F. von Oppen, *Phys. Rev. E*, vol. 51, p. 2647, 1995.
- [37] P. Bertelsen, C. Ellegaard, T. Guhr, M. Oxborrow, and K. Schaadt, "Measurement of parametric correlations in spectra of resonating quartz blocks," *PHYSICAL REVIEW LETTERS*, vol. 83 (11), pp. 2171–2174, 1999.
- [38] M. P. A. T. M.S. Hussein, C.P. Malta, "Effect of symmetry breaking on level curvature distributions," *PHYSICAL REVIEW E*, vol. 65 (5), p. Art. No. 057203 Part 2, 2002.
- [39] P. Leboeuf and M. Sieber, "Universality in quantum parametric correlations," *PHYSICAL REVIEW E*, vol. 60 (4), pp. 3969–3972 Part A, 1999.
- [40] A. Andersen, C. Ellegaard, A. D. Jackson, and K. Schaadt, "Random matrix theory and acoustic resonances in plates with an approximate symmetry," *Phys. Rev. E*, vol. 63, no. 6, p. 066204, May 2001.
- [41] L. Arnold, "On wigner's semicircle law for the eigenvalues of random matrices," *Probability Theory and Related Fields*, vol. 19, p. 191, 1971.
- [42] T. Gorin, "Generic spectral properties of right triangle billiards," *J. Phys. A: Math. Gen.*, vol. 34, p. 8281–8295, 2001.
- [43] M.V.Berry and M. Wilkinson, "Diabolical points in the spectra of

- triangles," *Royal Society of London Proceedings Series A*, vol. 392, pp. 15–43, Mar. 1984.
- [44] A. Miltenburg and T. Ruijgrok, "Quantum aspects of triangular billiards," *Physica A*, vol. 210, pp. 476–488, 1994.
 - [45] Y. S. A. Shudo, "Extensive numerical study of spectral statistics for rational and irrational polygonal billiards," *Phys. Rev. E*, vol. 47, no. 1, pp. 54–62, Jan 1993.
 - [46] E. B. Bogolmony, U. Gerland, and C. Schmit, "Models of intermediate spectral statistics," *Phys. Rev. E*, vol. 59 (2), pp. R1315–R1318, 1999.
 - [47] T. Gorin, M. Müller, and P. Seba., "Comment on "models of intermediate spectral statistics"," *Physical Review E*, vol. 63:068201, 2001.
 - [48] O. Bohigas and M. P. Pato, "Missing levels in correlated spectra," *PHYSICS LETTERS B*, vol. 595, pp. 171–176, 2004.
 - [49] O. Bohigas and M. P. Pato, "Randomly incomplete spectra and intermediate statistics," *PHYSICAL REVIEW E*, vol. 74 (3), p. 036212, 2006.
 - [50] T. D. N. H. Fletcher and, *The Physics of Musical Instruments*. Springer, N. York, 1991.