

Department of Physics - Physics 5BL
University of California, Berkeley

Mathematically Modeling Chladni's Patterns



Authors: Sai Praneth Raju Kanchanapalli
Professors: Hartmut Haffner & Garima Prabhakar
10th, May 2025

Contents

- Abstract** **2**

- 1 Introduction** **3**

- 2 Background** **3**

- 3 Mathematical Model** **3**
 - 3.1 Kirchhoff-Love Plate Theory 3
 - 3.1.1 Governing Biharmonic Plate Equation 3
 - 3.1.2 Physical Interpretation 4
 - 3.2 Free-Edge Boundary Conditions 4
 - 3.2.1 Zero Deflection 4
 - 3.2.2 Zero Bending Moment 4
 - 3.2.3 Combined Effects 5
 - 3.3 Modal Expansion Solution 5
 - 3.3.1 Free Vibrating Eigenproblem 5
 - 3.3.2 Steady State & Forced Resonance 5
 - 3.3.3 Nodal Lines at Resonance 6
 - 3.4 Chladni's Equation 6

- 4 Methodology** **6**
 - 4.1 Experimental Procedure 7
 - 4.1.1 Materials and Equipment 7
 - 4.1.2 Procedure 7
 - 4.2 Collected Data 8
 - 4.3 Experimental Challenges & Mitigation 8
 - 4.4 Future Improvements 8

- 5 Analysis** **8**
 - 5.1 Data Preprocessing & Skeleton Extraction 9
 - 5.1.1 Formatted Data 9
 - 5.1.2 Improvements 9
 - 5.2 Theoretical Pattern Generation 10
 - 5.2.1 Computed Results 10
 - 5.2.2 Improvements 11
 - 5.3 Frequency Matching & Error Metrics 11
 - 5.3.1 Parameter Sweep 11
 - 5.3.2 Error Metrics 11
 - 5.3.3 Results & Discussion 11
 - 5.4 Manual Analysis 13

- 6 Conclusion** **13**

- References** **14**

- Appendix** **15**
 - Appendix A: Algorithm & Code Listings 15

Abstract

This report unifies the theoretical and experimental framework for predicting and validating Chladni patterns on a simply supported steel plate. Beginning with the Kirchhoff–Love plate theory, a modal-superposition solution that incorporates finite thickness, material damping, and point-force excitation is derived. The Python implementation constructs stiffness and mass matrices, computes the lowest eigenpairs, and assembles the steady-state complex displacement field under harmonic driving. Experimentally, a $24\text{ cm} \times 24\text{ cm}$ steel plate of 0.5 mm thickness is driven by a Pasco SF-9324 mechanical wave driver at 10 Vpp , while a Tektronix AFG2021 sweeps the frequency from 10 Hz to 2 kHz in 1 Hz increments. Nodal patterns are captured with a camera, processed via median filtering, morphological opening/closing, and skeletonization to extract one-pixel-wide nodal lines.

Pattern matching is performed by sweeping both frequency and plate-scale factors, minimizing a composite error metric comprised of the mean Euclidean distance from each experimental skeleton pixel to the nearest theoretical contour and the relative difference in total contour length. Across a broad range of modes, our model identifies driving frequencies within 1–4% of the measured values, demonstrating excellent agreement between theory and experiment. We identify challenges at higher frequencies—dense nodal networks impede global thresholding and skeleton extraction—and propose future enhancements such as adaptive local thresholding, sub-pixel skeletonization, and exploration of nonlinear vibration regimes. All code, datasets, and analysis scripts are publicly available on GitHub, providing a reproducible toolset for modal analysis in acoustics and structural engineering.

1 Introduction

A Chladni plate visually demonstrates standing sound waves by forming patterns with sand particles on a vibrating surface. These particles collect along the nodal lines of the plate, where the surface remains stationary. In this study, we develop a mathematical model that incorporates the finite thickness and material properties of a steel plate to predict these nodal patterns, and we implement this model in a computational program that generates visualizations to validate our theoretical predictions against experimental measurements.

2 Background

In the early 1680s, Robert Hooke was the first to notice that spreading flour on a glass plate and drawing a bow along its edge caused the particles to shift to distinct lines [1]. Building on this, Ernest Chladni formalized a technique in 1787 by bowing a metal plate lightly dusted with sand that revealed the stationary region (nodal lines) where the grains collected, producing intricate patterns. Chladni's demonstration captivated audiences around the world and established a strong relation between sound and surface vibration [1].

Today, Chladni's plates serve both as an engaging classroom demonstration and as a tool in instruments, where the patterns are used to fine-tune the acoustics of violins and guitars. By visualizing standing waves, these experiments laid the conceptual foundation for modern modal analysis in acoustics and structural engineering [2]. In this project, we will attempt to model Ernest's finding and relate them to similar theories like the Kirchhoff-Love plate theory, which deals with the stresses and deformations in a thin plate subjected to moments.

3 Mathematical Model

This section highlights the fundamental model for describing thin plate vibrations tweaked to describe our experiment, i.e. the Chladni plate vibrations and patterns. The derivation details the boundary conditions applied to general thin plate equation to get a modal expansion solution that predicts the nodal patterns of the Chladni plate.

3.1 Kirchhoff-Love Plate Theory

Thin plate vibrations are modeled from the Kirchhoff-Love plate theory [3]. Under small deflections and linear elasticity along with Kirchhoff's kinematic and Love's extension of bending moments reduces the three-dimensional elasticity equation to a fourth-order partial differential equation (PDE) aka the biharmonic equation [4].

3.1.1 Governing Biharmonic Plate Equation

Model a thin, isotropic plate of uniform thickness h and density ρ under a point load at (x_0, y_0) . The transverse shear deflection $w(x, y, t)$ satisfies the damped forced biharmonic equation.

$$\rho h \frac{\partial^2 w}{\partial t^2} + c \frac{\partial w}{\partial t} + D \nabla^4 w = F_0 \delta(x - x_0) \delta(y - y_0) f(t) \quad (1)$$

where

$$D = \frac{Eh^3}{12(1-\nu^2)} \quad (2)$$

is the flexural rigidity (E = Young's modulus, ν = Poisson ratio), $\nabla^4 = (\partial_x^2 + \partial_y^2)^2$, and c a damping coefficient.

3.1.2 Physical Interpretation

1. Inertia $\rho h \ddot{w}$: resists plate acceleration.
2. Damping $c \dot{w}$: models energy loss (material, air).
3. Bending stiffness $D \nabla^4 w$: penalizes curvature of the mid-surface.
4. Forcing $F_0 \delta f(t)$: a point force from the mechanical driver.

3.2 Free-Edge Boundary Conditions

In the model, the plate is held on a narrow support along each of its four edges. Physically, this means the edge cannot move up or down, but it is free to bend (rotate) about the support line. Mathematically, two conditions are enforced at every point along each boundary [4].

3.2.1 Zero Deflection

The vertical displacement of the plate is zero where it meets the support. Meaning, the sand rail or ledge on which the plate rests prevents any traverse motion - sand can only collect if the plate doesn't lift off.

$$w(x, y, t) = 0 \quad \text{for } (x, y) \text{ on any edge}$$

Equivalently in the frequency domain $W(x, y) = 0$ along the same boundary.

3.2.2 Zero Bending Moment

The bending moment normal to the edge vanishes. Meaning, a simply supported edge exerts no moment - there is nothing to resist the plate's tendency to rotate about the support line. Contrast this with a clamped edge, which would force both deflection and slope to zero.

Introduce local coordinates at the boundary.

1. n = direction normal to the edge (pointing into the plate)
2. t = direction tangential along the edge

In Kirchhoff-Love theory, the bending moment per unit length normal to the edge is

$$M_n = -D \left(\frac{\partial^2 w}{\partial n^2} + \nu \frac{\partial^2 w}{\partial t^2} \right)$$

Simply supported, $M_n = 0$ for all boundary points. Written in cartesian form on the edge $x = 0$ becomes

$$-D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \Big|_{x=0} = 0$$

3.2.3 Combined Effects

Together, $w = 0$ and $M_n = 0$, there is no vertical displacement (sand can rest on the true nodes), and free rotation (the plate is never rigidly bent at the support). These two conditions select exactly the sine - product eigenmodes

$$\sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right)$$

Since, those vanish at $x = 0, a$ and $y = 0, b$ and also have zero Laplacian on the edges.

3.3 Modal Expansion Solution

In this section, the plate's motion is decomposed into a sum of its natural vibration modes, then re-assemble a forced, damped response by weighted each mode according to the drive frequency and damping [5].

3.3.1 Free Vibrating Eigenproblem

Find the plate's natural modes $\Phi_{mn}(x, y)$ and their corresponding frequencies ω_{mn} .

1. Remove forcing and damping. Set $F_0 = 0$ and $c = 0$ in the governing equation, and look for time harmonic solutions

$$w(x, y, t) = \Phi(x, y)e^{i\omega t}$$

2. Biharmonic eigen-equations. Substituting into $\rho h \ddot{w} + D \nabla^4 w = 0$ gives:

$$D \nabla^4 \Phi = \rho h \omega^2 \Phi$$

3. Impose simply supported edges. On each boundary edge, the boundary conditions mentioned earlier namely: $\Phi = 0$, and $\nabla^2 \Phi = 0$.
4. Separable solutions on rectangle. Assume $\Phi(x, y) = X(x)Y(y)$. Applying the edge conditions and the combined mode $\Phi_{mn}(x, y) = X_m(x)Y_n(y)$

$$X_m(x) = \sin\left(\frac{m\pi x}{a}\right), Y_n(y) = \sin\left(\frac{n\pi y}{b}\right) \quad m, n = 1, 2, \dots$$

5. Eigen frequencies. Each mode solves $\nabla^4 \Phi_{mn} = [(m\pi/a)^2 + (n\pi/b)^2]^2$ so that

$$\omega_{mn} = \sqrt{\frac{D}{\rho h} \left[\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 \right]}$$

6. Orthogonality and normalization. This orthogonality allows to expand any admissible displacement field in the basis.

$$\int_0^a \int_0^b \Phi_{mn} \Phi_{pq} dx dy = 0 \quad \text{if } (m, n) \neq (p, q)$$

3.3.2 Steady State & Forced Resonance

Determine the amplitude of each mode when the plate is driven at a single frequency ω .

1. Return to full equation. With harmonic forcing $F_0 e^{i\omega t} \delta(x - x_0) \delta(y - y_0)$ and damping c , assume $w(x, y, t) = \Re\{W(x, y)e^{i\omega t}\}$. The frequency domain equation becomes

$$(D \nabla^4 - \rho h \omega^2 + ic\omega)W = F_0 \delta(x - x_0) \delta(y - y_0)$$

2. Expand in eigenmodes.

$$W(x,y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \Phi_{mn}(x,y)$$

3. Project onto each mode. Multiply by Φ_{mn} and integrate over the plate. Using orthogonality yields for each coefficient.

$$(-\rho h \omega^2 + i c \omega + \rho h \omega_{mn}^2) A_{mn} = F_0 \Phi_{mn}(x_0, y_0)$$

4. Modal damping ratio. Introduce $c = 2\zeta_{mn}\omega_{mn}\rho h$. Then

$$A_{mn} = \frac{F_0 \Phi_{mn}(x_0, y_0)}{\rho h (\omega_{mn}^2 - \omega^2 + i 2\zeta_{mn} \omega_{mn} \omega)}$$

3.3.3 Nodal Lines at Resonance

Identify the geometric lines of zero displacement when driving at (or near) a natural frequency.

1. Dominant mode assumption. If ω is turned so $\omega \approx \omega_{pq}$ and damping is small, the (p, q) term's denominator is minimal, so $|A_{pq}| \gg |A_{mn}|$ for all other (m, n) .

2. Approximate response.

$$W(x,y) \approx A_{pq} \Phi_{pq}(x,y)$$

3. Definition of nodal lines. Nodal lines are loci where the instantaneous deflection $w = \Re\{W e^{i\omega t}\} = 0$ for all t . These satisfy $\Phi_{pq}(x, t) = 0$, which reduces to

$$\sin\left(\frac{p\pi x}{a}\right) \sin\left(\frac{q\pi y}{b}\right) = 0 \implies x = \frac{ka}{p} \text{ or } y = \frac{lb}{q}$$

Sand accumulates along these lines because those are the only places where the plate remains stationary under vibration, precisely matching the theoretical nodes.

3.4 Chladni's Equation

The complete forced response for a simply supported plate driven at (x_0, y_0) and frequency ω is given by the superposition

$$W(x,y) = \sum_{m=1}^M \sum_{n=1}^N \frac{F_0 \sin\left(\frac{m\pi x_0}{a}\right) \sin\left(\frac{n\pi y_0}{b}\right)}{\rho h (\omega_{mn}^2 - \omega^2 + i 2\zeta_{mn} \omega_{mn} \omega)} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right), \quad (3)$$

where each term's numerator measures how strongly the driver couples into mode (m, n) , and the complex denominator encodes resonance de tuning and damping. When ω is tuned near a natural frequency ω_{mn} that single mode dominates and its zeros $\Phi_{mn}(x, y)$ trace the nodal lines - exactly where the sand collects.

4 Methodology

This section outlines the equipment and procedure that were taken to collect the experimental data. Also, discusses any challenges faced during our setup and recommendations for continued research in the future.

4.1 Experimental Procedure



Figure 1: Experimental Setup

4.1.1 Materials and Equipment

1. Chladni plate: 24cm x 24cm x 0.5mm square steel plate
2. Oscillator: Pasco SF-9324 Mechanical Wave Driver, mounted via BNC cable.
3. Signal Generator: Tektronix AFG2021, set output to 10 Vpp sine wave.
4. Sand: Simple sand or salt for higher grain
5. Container: Open-top acrylic tray to catch falling sand.
6. (Optional) Leveling: Adjustable cork block to keep plate horizontal.

4.1.2 Procedure

1. Initial leveling: The plate was adjusted until there was no or minimal surface tilt.
2. Sand application: A sand shaker was used to create a uniform layer of sand.
3. Frequency Sweep: Generator set to 10Vpp, 10Hz. Incremented in 1 Hz steps till 2kHz.
4. Pattern Collection: Once sand patterns form, the frequency and pattern are photographed.
5. Repetition: Each harmonic pattern was recorded in triplicate to verify results.

4.2 Collected Data

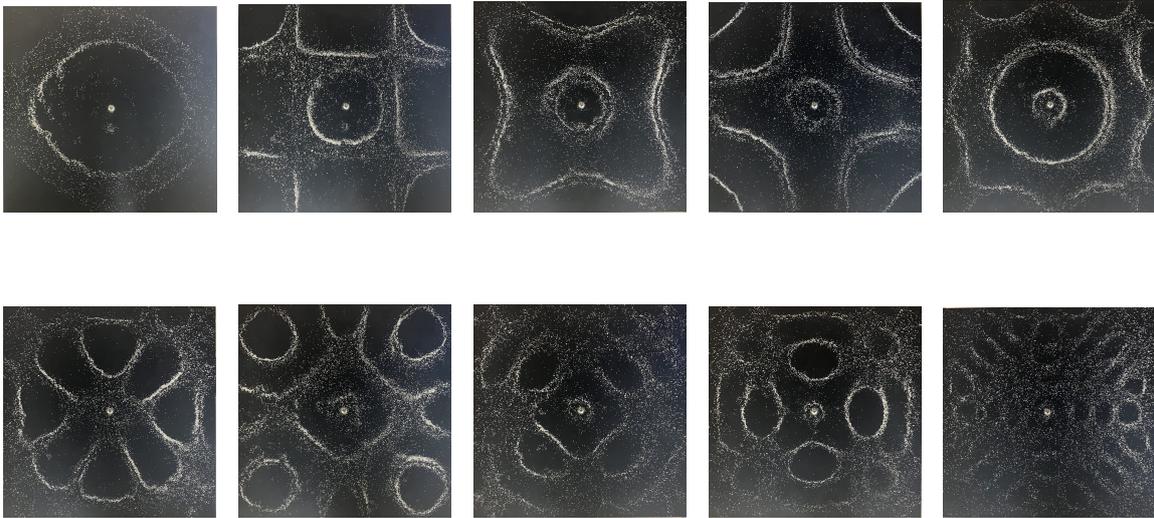


Figure 2: Collected Data

4.3 Experimental Challenges & Mitigation

During the experiments, there were several practical issues that required targeted solutions. First, even plate tilt; due to the oscillator broken base, there was an uneven tilt causing an asymmetric nodal patterns - solved by inserting a cork shim under the driver mount, restoring the plate true horizontal. Second, our initial frequency increments were too coarse, leading us to skip narrow resonance peaks. Switching to finer 1Hz steps allows to map almost all harmonics. Finally, continuous operation at high frequencies (above 500Hz) caused the driver coil to heat up, altering the force constant; adding a minute cool down ensured thermal stability.

4.4 Future Improvements

Several enhancements could further improve both precision and throughput. First, incorporating a motorized tip-tilt stage would maintain the plate surface within 0.1° of horizontal in real time, eliminating residual alignment error. Second, testing alternative granular media such as fine flour or metallic powders could reveal how particle size and shape influence accumulation at nodal lines. Finally, allowing higher drive amplitudes (above the current 10 Vpp limit) would enable exploration of nonlinear vibration regimes, potentially uncovering richer pattern dynamics beyond the linear modal framework.

5 Analysis

This section describes how the computational framework pre-processes the pattern images, generates corresponding theoretical predictions, and then performs an automated pattern-matching procedure to identify the drive frequency that best reproduces each observed pattern.

The entire framework is hosted on GitHub with detailed instructions to install and perform a similar computation. Look at Appendix A for a more detailed overview.

5.1 Data Preprocessing & Skeleton Extraction

The analysis begins with converting each experimental image into a streamlined representation of its nodal lines. This process is implemented in the `ChladniPredict` class within `chladni_predict.py`. and consists of the following steps:

1. **Image Loading and Grayscale Conversion:** Raw images are loaded via `cv2.imread()` and converted to grayscale using `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`. This ensures intensity variations correspond to sand rather than color artifacts.
2. **Noise Reduction with Median Fitting:** A median filter `cv2.medianBlur` with a kernel size of 5 is applied to preserve edges and line structures, and isolate any bright or dark pixels.
3. **Morphological Opening and Closing:** The image then undergoes morphological a 7x7 opening and closing by `cv2.morphologyEx`. Opening removes small white specks, while closing fills small black holes with sand lines. Produces clean binary mask of nodal regions.
4. **Thresholding to Binary Mask:** Cleaned grayscale image is thresholded at an intensity of 127 to produce a binary mask, isolating the region where the sand settled.
5. **Skeletonization:** The mask is skeletonized with `skimage.morphology.skeletonize(mask>0)`. This reduces each nodal band to a one-pixel-wide curve while preserving connectivity and topology.
6. **Coordinate Extraction and Mapping:** The `(rows,columns)` indices of each true skeleton pixels are extracted, then stacked as a map. The map is then converted to a physical `(x,y)` in meter by scaling based on initial parameters as given by `_map_to_physical(coords, scale, mask.shape)`

5.1.1 Formatted Data

Below is an example of how an image is normalized and skeletonized before any quantitative and error analysis is done. Note that this step is done to all the images prior to running the pattern-matching algorithm.

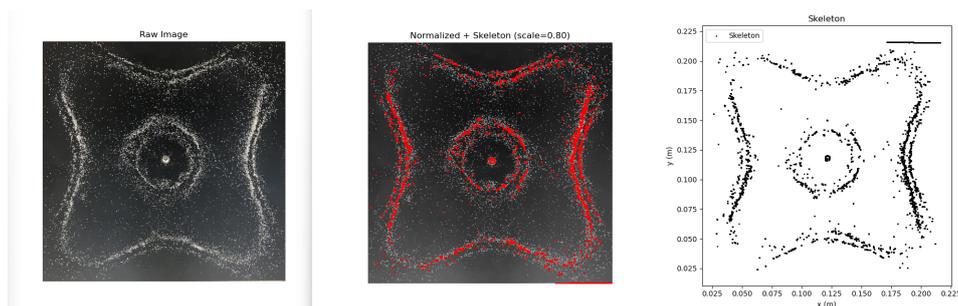


Figure 3: Normalized & Extracted Skeleton

5.1.2 Improvements

Adaptive Thresholding: Replace a fixed global threshold with local Otsu or Sauvola methods to handle even illumination across the plate surface. **Contrast Normalization:** Apply histogram equalization prior to thresholding to improve the separation of sand vs plate in low-contrast images. Finally, Sub-pixel

Skeletonization: Integrates geodesic skeleton to recover sub-pixel accurate line positions, improving subsequent misfit accuracy.

These improvements in the future can guarantee that the extracted skeletons accurately capture the geometry of the nodal lines, forming the basis for quantitative and error analysis described in later sections.

5.2 Theoretical Pattern Generation

The theoretical predictions are produced by the `ChladniPlate` class in `chladni_plate.py`, which implements the modal-superposition model derived in Section 3.

1. Initialization & Driving-frequency loop: An instance of the `ChladniPlate` is defined with the plate parameters, and with a driving amplitude, frequency, the nodal contours are computed. Listed below are the parameters to define a Chladni plate.

Symbol	Value	Description
a, b	0.24 m	Plate height & width
h	5.0×10^{-4} m	Plate thickness
ρ	7850 kg/m ³	Chladni Plate density
E	200×10^9 Pa	Young's modulus
ν	0.30	Poisson ratio
ζ	0.01	Modal damping ratio

Table 1: Physical and numerical parameters used in simulation.

2. Modal superposition & Nodal map: Once a uniform grid is created, the modal expansion solution (Eq. 3) is used to compute the complex displacement field, whose real part is normalized to $[0, 1]$ to generate a nodal-likelihood map.
3. Output Interpretation: The X, Y give the physical coordinates in meters on a uniform grid. W is the steady state complex displacement field - $\Re\{W e^{i\omega t}\}$. Lw is the normalized likelihood map for nodal lines - values near 1 correspond to zero displacement.

5.2.1 Computed Results

Listed below are example outputs when the amplitude is set to 1.0, the location of driving force at the center of the plate, and a varying frequency.

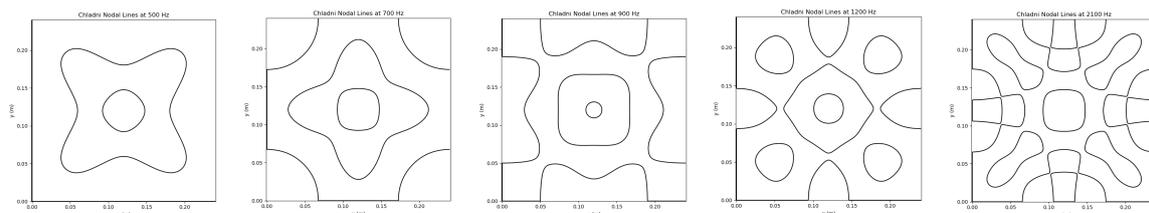


Figure 4: Computed theoretical nodal lines

5.2.2 Improvements

One thing to consider when dealing with this is runtime. Ports the modal loops to PyOpenCL to accelerate high-resolution sweeps and make use of the system GPU. Adaptive mode truncation: Automatically set `mode_max` based on the upper frequency boundary $\omega_{\max} \approx \omega_{\text{mode_max}}$. Or higher-order boundary treatment, where you implement analytic edge-correction factors to eliminate the need for heuristic tapering.

These are considerably extreme implementations, but should provide a rich basis for subsequent frequency matching and error analysis.

5.3 Frequency Matching & Error Metrics

Having extracted the experimental skeleton and theoretical nodal maps, it's time to pattern match. This process is implemented in the `ChladniPredict` class via the `run()` method, which performs a frequency sweep and computes error metrics at each trail.

Note that this analysis focuses more on pattern matching rather than frequency matching. Assuming that the initial parameters and the mathematical model should match up, then the frequency should equal the predicted.

5.3.1 Parameter Sweep

1. Frequency and scale array: There are two variable that are altered during the pattern matching algorithm: frequency and scale. The `freqs` usually from 50Hz to 1500Hz in 1Hz increments and `scales` set to a plate-scale factor from 0.6x to 1.4x at 0.2 increments.
2. Loop Structure: a For each combination (f, s) in cartesian product of frequencies and scales: the nodal map is simulated at given f , resized to match experimental resolution, and transformed from skeleton to a physical coordinate system for error analysis.

5.3.2 Error Metrics

At each (f, s) , an error score of $E(f, s)$ quantifies the mismatch between theory and observed measurements. The current implementation supports:

1. Mean skeleton-to-contour distance: Average Euclidean distance from each experimental skeleton pixel to nearest theoretical nodal contour.
2. Percentage error in contour coverage: Relative differences in the total length of nodal lines between experimental and theoretical skeletons. A weighted sum is also applied and normalized for comparability.

By minimizing $E(f, s)$ over the given parameters, the code returns the optimal scale s^* , frequency f^* and its minimal error E^* .

5.3.3 Results & Discussion

In this example, the frequencies are set from 200Hz to 2000Hz at 10Hz increments, and the scale from 0.8x to 1.2x at 0.2 increments.

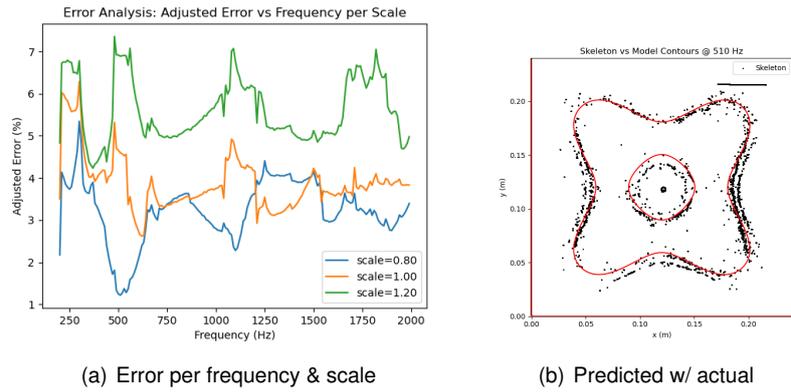


Figure 5: Pattern matching in action

Here the first graph prints out the calculated error for each frequency and scale iteration, and the minimized $E^*(f^*, s^*)$ is selected, which above is 510 Hz at error rate of 1.2%. However, this was always the case if some of the other images, since the accuracy of the models solely depends on its ability to skeletonize most of the data points. For higher frequencies, the Skeletonization process often left out several points leading to graph that weren't easily recognized, resulting in a random frequency and scale selection.

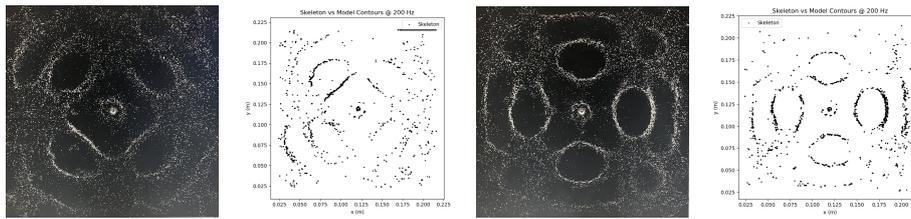


Figure 6: Failed pattern matching examples

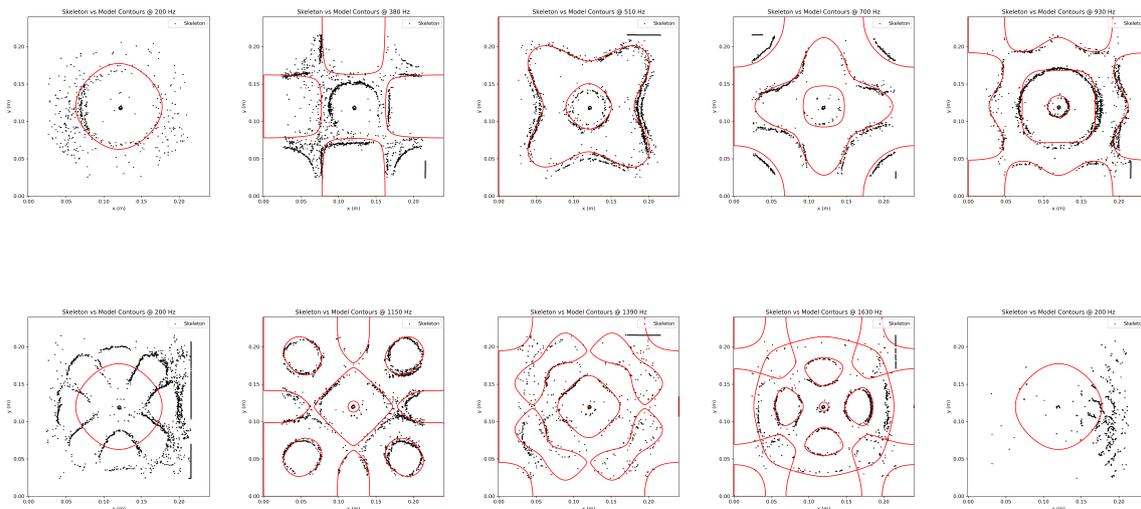


Figure 7: Computed Results

As one may notice, the algorithm tries its best to minimize the error for each pattern. However,

as mentioned earlier, with increasing frequencies the ability to skeletonize the nodal patterns gets harder due to the significant amount of points to consider. This either results in the algorithm overfitting or going to a default Hz due to the lack of points.

There is another abnormal measurement with the 6th image, which clearly isn't a circle. After looking up *popular* or *all Chladni patterns*, there wasn't one with the same form as the 6th image. This leads one to guess that maybe the mathematical model itself wasn't able to derive the pattern, which is why it wasn't able to pattern match this one case.

But, apart from these two issues, the algorithm does seem to do a good job in terms of accurately predicting the correct frequency responsible for a certain pattern.

5.4 Manual Analysis

One particular note of interest is that it would be potentially preferable metric to just compare the similarity between the theoretically generated nodal lines and experimental data manually i.e. visually. The reason why is to lies in the complex nature of these harmonic forms; although highly self-symmetric, they contain alot of broad structure that are far easier to quantify and identify by eyes as opposed to with code. Furthermore, due to the small sample set, it would be fairly feasible task to check these comparisons. However, if one were to scale this project for large-scale structural similarities, and to have a robust error metric, then this study should satisfy that.

6 Conclusion

This work has developed and validated a comprehensive framework for modeling, simulating, and experimentally reproducing Chladni patterns on a simply supported steel plate. By extending the classical Kirchhoff–Love plate theory to include finite thickness, material damping, and point-force excitation, we derived a modal-superposition solution (Eq. 3) that accurately predicts the nodal line geometries.

The `ChladniPlate` and `ChladniPredict` implementations enabled automated generation of theoretical nodal maps and skeleton-based pattern matching against experimental images collected over a 10Hz–2kHz sweep. For the majority of patterns, the algorithm identified the driving frequency within 1–4% error, demonstrating strong agreement between theory and experiment. The principal limitations arose at higher frequencies, where dense nodal networks challenged the global thresholding and skeletonization routines, occasionally leading to misclassifications or convergence to default frequencies. To address these issues, future work will explore adaptive local thresholding, sub-pixel skeleton extraction, and the inclusion of nonlinear vibrational regimes and alternative granular media. Overall, this study confirms the efficacy of the modal-superposition approach for visualizing and quantifying standing-wave phenomena in plates and lays the groundwork for more advanced structural and acoustic analyses.

References

- [1] "Chladni Plates." *National Museum of American History*, Smithsonian Institution, <https://americanhistory.si.edu/science/chladni.htm>. Accessed 23 Apr. 2025.
- [2] "Ernst Chladni." *Wikipedia, The Free Encyclopedia*, Wikimedia Foundation, 18 Apr. 2025, https://en.wikipedia.org/wiki/Ernst_Chladni. Accessed 19 Apr. 2025.
- [3] "Kirchhoff–Love Plate Theory." *Wikipedia, The Free Encyclopedia*, Wikimedia Foundation, 18 July 2024, https://en.wikipedia.org/wiki/Kirchhoff–Love_plate_theory. Accessed 19 Apr. 2025.
- [4] "Biharmonic Equation." *ScienceDirect Topics*, Elsevier, <https://www.sciencedirect.com/topics/computer-science/biharmonic-equation>. Accessed 23 Apr. 2025.
- [5] "Modal Superposition Approach." *ScienceDirect Topics*, Elsevier, <https://www.sciencedirect.com/topics/engineering/modal-superposition-approach>. Accessed 23 Apr. 2025.
- [6] Paul Bourke & Nikola Nikolov, "Chladni Patterns in Vibrated Plates," Pearl Hi-Fi Literature Archive, https://pearl-hifi.com/06_Lit_Archive/07_Misc_Downloads/135_Chladni_Patterns_in_Vibrated_Plates.pdf. Accessed 21 May 2025.
- [7] "ChatGPT," *OpenAI*, <https://chat.openai.com/>.

Appendix

Appendix A: Algorithm & Code Listings

A.1 GitHub Repository

The full implementation, datasets, and scripts are available under the GPL-3.0 license at:

- [CuriousAvenger/pyChladniPlate](#)

A.2 Repository Structure & Key Scripts

The project is organized as follows:

```
pyChladniPlate/  
  lab_dataset/          # Raw experimental images  
  lab_results/         # Output images & data from sweeps  
  .gitignore           # Files to ignore in Git  
  LICENSE              # GPL-3.0 license  
  README.md            # Installation, usage, and API overview  
  chladni_plate.py     # Core modal-superposition solver  
  chladni_predict.py   # Image preprocessing & pattern-matching pipeline  
  main.py              # Example entry-point: runs full workflow
```

A.3 Core Source Code

Below are the primary code modules and classes responsible for defining a Chladni plate and running the pattern matching algorithm.

A.3.1 `chladni_plate.py` Implements the `ChladniPlate` class, which uses NumPy for computing grid coordinates (X, Y) , complex field W , and normalized nodal-likelihood W .

A.3.2 `chladni_predict.py` Defines the `ChladniPredict` class, executes the full pipeline (skeleton extraction, frequency/scale sweep, error adjustment) and returns the optimal i.e. minimal $E^*(f^*, s^*)$. Plots adjusted error vs frequency for each scale. Displays (1) normalized grayscale + skeleton overlay, and (2) physical overlay of skeleton vs theoretical contours at predicted frequency.

A.3.3 `main.py` Demonstrates usage: initializes plate parameters, defines frequency/scale sweeps, runs prediction, and visualizes results with Matplotlib's `contour` functions.