Journal of Applied Engineering Mathematics

Volume 10, December 2023

MATHEMATICAL MODEL OF A CHLADNI PLATE

Suzanna Gilbert & Noah Housley

Mechanical Engineering Department
Brigham Young University
Provo, Utah 84602
noahhousley@gmail.com, gsuzi26@byu.edu

ABSTRACT

Chladni plates are known for their beautiful salt patterns that seem to magically appear when excited by a sound. When driven by vibrating source, a Chladni plate develops 2D standing wave patterns that can be visualized when salt is poured onto the plate. In this paper, a mathematical model for a Chladni plate is developed and solved to find these mode shapes mathematically. Although the solutions resemble 2D mode shapes, they do not match up very well with the mode shapes observed experimentally on a Chladni plate. Possible reasons for this discrepancy are discussed. It is concluded that the mathematical model used is not accurate enough to produce a solution identical to the physical plate.

NOMENCLATURE

γ	Damping Constant
ν	Transverse Wave Speed
S	Source Function
s_0	Source Amplitude
L	Length of Plate
M	Width of Plate
δ	Dirac Delta Function
ω	Source Frequency
Mode	Standing wave shape that occurs at a natural frequency
Node	Location of no displacement in a standing wave.

INTRODUCTION

Ernst Chladni first discovered in the late 1700s that he could make beautiful patterns appear on a powder-covered piece of metal by playing its edge with his bow, but at the time, little was known about this fascinating phenomenon. His trick impressed people enough that before long, many well-known mathematicians raced to discover the explanation for these Chladni patterns, at the promise of a monetary reward [1].

Today, although the physical process behind the Chladni plate is well understood, it is still used frequently as a demonstration to fascinate both scientists and laymen alike. The thin metal plate, when excited by a vibrating source, develops 2D standing waves at its natural frequencies. When salt is poured onto the plate, the grains settle into the nodes of the standing wave, allowing the user to visualize the standing wave patterns.

This paper seeks to model a Chladni plate mathematically using the 2D wave equation. A model will be presented for a square Chladni plate driven at the center, and the solution process will be explained. Then the theoretical mode shapes will be compared to the patterns observed on a real Chladni plate of similar dimensions.

Figure 1. A modern Chladni plate driven at the center.

MATHEMATICAL MODEL FOR A CHLADNI PLATE

Equation (1) shows the equation chosen to model the plate, where ν is the transverse wave speed in the plate and γ is a damping term. This equation does not account for the thickness of the plate but approximates it as a membrane. The plate is relatively thin, so this approximation was assumed to be appropriate.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + S = \frac{1}{\nu^2}\frac{\partial^2 u}{\partial t^2} + 2\gamma^2 \frac{\partial u}{\partial t} \quad (1)$$

Some Chladni plates are driven in the center with a harmonic source, while others are played along the edge with a bow. The

former is modeled here. The source, *S*, is a sinusoidal impact force in the center of the plate, with amplitude s_0. The impact force is modeled using Dirac Delta functions, as shown in equation (2).

$$S = s_0 * \delta\left(x - \frac{L}{2}\right) * \delta\left(y - \frac{M}{2}\right) * \sin(\omega t) \quad (2)$$

Most Chladni plates are fixed at the center with the edges free. To simulate this, Neumann boundary conditions were used along each edge. This means that the derivative of the function at the plate boundaries is always equal to zero (see figure 2).

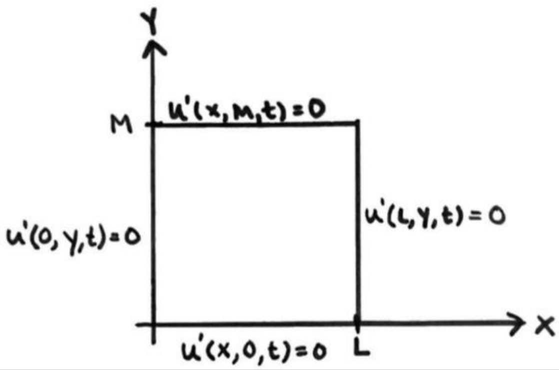

Figure 2. Boundary conditions of Chladni plate model.

Finally, to simulate a plate driven from rest, the initial position and velocity were set to zero (equation 3). This greatly simplifies the solution process.

$$u'(x, y, 0) = u(x, y, 0) = 0 \quad (3)$$

SOLVING THE WAVE EQUATION

The solution to the 2D wave equation is broken down into six steps.

Step 1: Identify the eigenvalues and eigenfunctions.
The eigenvalues and eigenfunctions depend on the boundary conditions. The eigenvalues and eigenfunctions corresponding to Neumann boundary conditions in both the x and y directions are summarized in Table 1.

Table 1. Eigenvalues and eigenfunctions for Neumann-Neumann boundary conditions.

Dimension	Eigenvalues	Eigenfunctions	Squared Norm
X	$\mu_0 = 0$ $\mu_n = \frac{n\pi}{L}, n = 1, 2, 3 \ldots$	$X_0 = 1$ $X_n = \cos(\mu_n x)$	$\frac{L}{2}$
Y	$\lambda_0 = 0$ $\lambda_m = \frac{m\pi}{M}, m = 1, 2, 3 \ldots$	$Y_0 = 1$ $Y_m = \cos(\lambda_m y)$	$\frac{M}{2}$

Step 2: Take the Fourier transform in both the x and y directions.
This will eliminate the second-order partial derivatives in x and y. The Fourier transform of equation (1) in both x and y becomes the following:

$$-\mu_n^2 \bar{\bar{u}}_{nm} - \lambda_m^2 \bar{\bar{u}}_{nm} + s_0 \cos\left(\mu_n \frac{L}{2}\right) \cos\left(\lambda_m \frac{M}{2}\right) \sin(\omega t) = \frac{1}{\nu^2} \frac{\partial^2 u}{\partial t^2} + 2\gamma^2 \frac{\partial u}{\partial t} \quad (4)$$

Step 3: Take the Laplace transform.
The Laplace transform eliminates the time derivatives so that the equation can be solved for *u*. Equation (5) shows the transformed function.

$$-\mu_n^2 \bar{\bar{U}}_{nm}(s) - \lambda_m^2 \bar{\bar{U}}_{nm}(s) + s_0 \cos\left(\mu_n \frac{L}{2}\right) \cos\left(\lambda_m \frac{M}{2}\right) \mathcal{L}\{\sin(\omega t)\} = \frac{1}{\nu^2} s^2 U(s) + 2\gamma^2 s U(s) \quad (5)$$

Step 4: Solve for U in the Laplace domain.
Solving for U yields equation (6):

$$\bar{\bar{U}}_{nm}(s) = \frac{\nu^2 \phi \, \mathcal{L}\{\sin(\omega t)\}}{s^2 + 2\gamma^2 \nu^2 s + \mu_n^2 \nu^2 + \lambda_m^2 \nu^2} \quad (6)$$

where

$$\phi = s_0 \cos\left(\mu_n \frac{L}{2}\right) \cos\left(\lambda_m \frac{M}{2}\right) \quad (7)$$

Step 5: Take the inverse Laplace Transform.
After completing the square in the denominator and introducing another variable β, the equation is in a recognizable form for the inverse Laplace transform (equations 8 and 9). The inverse Laplace transform is shown in (10), where * denotes the convolution.

$$\bar{\bar{U}}_{nm}(s) = \frac{\nu^2 \phi \, \mathcal{L}\{\sin(\omega t)\}}{\beta} \frac{\beta}{(s + \gamma^2 \nu^2)^2 + \beta^2} \quad (8)$$

where

$$\beta = \sqrt{\mu_n^2 \nu^2 + \lambda_m^2 \nu^2 - \gamma^4 \nu^4} \quad (9)$$

$$\mathcal{L}^{-1}\left\{\bar{\bar{U}}_{nm}(s)\right\} = \bar{\bar{u}}_{nm} = \frac{\nu^2 \phi}{\beta} \sin(\omega t) * e^{-\gamma^2 \nu^2} \sin(\beta t) \quad (10)$$

Step 6: Take the inverse Fourier transform in x and y using a sum.
The finite inverse Fourier transform is carried out using a sum in x and y. The result obtained in the previous step is multiplied by the eigenfunctions and divided by the squared norms (see Table 1) to get the final solution (equation 11).

$$u(x,y,t) = \frac{4}{LM}\sum_{n=0}\sum_{m=0}\bar{\bar{u}}_{nm}\cos(\mu_n x)\cos(\lambda_m y) \qquad (11)$$

VISUALIZING THE MODE SHAPES

By plotting the solution derived above, we can visualize the theoretical mode shapes of our approximated Chladni plate. First, physical values must be assigned to the constants γ, ν, s_0, ω, M, and L. The length (L) and width (M) of the plate were both set to 0.22 m to match the dimensions of a real Chladni plate found at BYU. The amplitude, s_0, does not affect the mode shapes in the analytical solution, however at 0.01 m it is much smaller than the dimensions of the plate, which is realistic. Experimentally, s_0 would control how much the salt is bounced around on the plate.

The value of ω needs to be set to the natural frequencies of the plate, because those are the frequencies where the mode shapes occur. The natural frequencies are the eigenvalues of the function, so ω was set to be the eigenvalue for each mode. Conveniently, the eigenvalues are the same in x and y because the plate is square (M and L are the same).

Wave speed ν and damping γ depend on the physical properties of the Chladni plate such as its stiffness or tension, which are not easily measured. A damping value of 0.02 was chosen, which is in the normal range for metal plates. The transverse wave speed was set to ν=0.5 m/s, which is much slower than the transverse wave speed in a metal plate. When the wave speed was set to a more realistic value, the code broke down and plotted the same mode shape for every frequency. It was found that setting ν to a much smaller value allowed the mode shapes to plot well. This is likely because a higher wave speed corresponds to a much stiffer plate and the frequencies being used are too low to excite complex mode shapes in a plate of this stiffness. Potentially, this could be corrected by choosing higher frequency values. However, higher values for frequency or wave speed greatly increase the time it takes for the solution code to run, so this option was not explored.

Finally, the number of terms in the summation is an important parameter; it must be sufficient to capture the mode shape, but not much larger because adding terms increases the run time of the code. For modes 1 through 5, it was found that 10 terms were sufficient and that adding more terms did not change the solution. For modes 6 through 8, 20 terms were used. With this choice of parameters, the first 8 mode shapes are shown in figure 3.

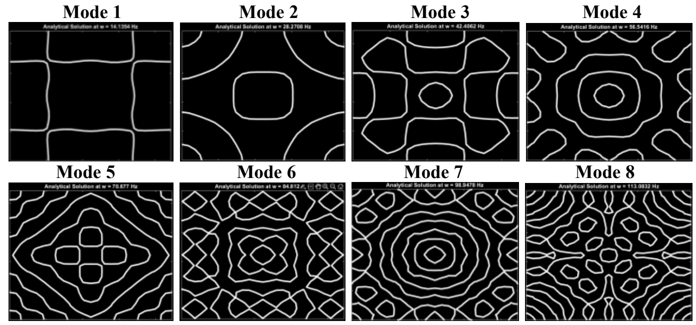

Figure 3. First 8 mode shapes found using solution to the 2D wave equation.

For comparison, the first ten mode shapes observed on a real Chladni plate [2] are shown in figure 4.

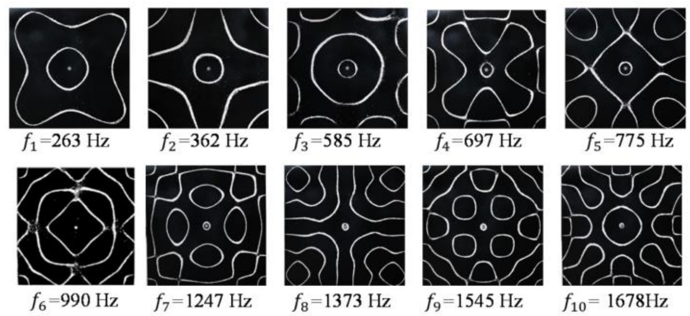

Figure 4. Mode shapes observed esperimentally on a square Chladni plate of similar size

DISCUSSION

The 8 modes in figure 3 resemble Chladni patterns in general, but they are significantly different than those observed on a real Chladni plate. Mode 2 is the only mode shape that matches between the mathematical and the experimental. Some of the other modes have similar features, such as mathematical mode 3 with experimental mode 5, and mathematical mode 5 with experimental mode 9.

There are many possible reasons that the model used in this study was insufficient to predict the mode shapes of a Chladni plate. One significant reason may be that the model assumes that the plate is a membrane, rather than a plate. Perhaps if thickness was accounted for, the solutions would be more accurate. Perhaps this would require a different model than the 2D wave equation (for example, the Kirchoff-Love plate theory).

Another source of the discrepancy could be that none of the physical constants in the equation matched the experimental plate. These parameters would likely have to be measured to find their correct values. Possibly an optimization algorithm could be used to match the solutions to the experimental data by varying the parameters in question.

The source excitation in the model was also slightly different than the real Chladni plate. The model uses a Dirac Delta function, which is infinitesimally narrow, but on the real plate there is a finite area excited by the driver and bolt. Inertial effects could also influence the modes of the plate as the plate vibrates up and down. The 2D wave equation does not account for this.

A plate free on all edges is a problem that is known for being difficult or even impossible to solve analytically and get solutions identical to physical experiments. People often resort to numerical methods such as finite-element analysis methods to solve a Chladni plate. With this history in mind, the solution presented in the paper was a noble attempt, though perhaps overly simplistic.

CONCLUSIONS

This experiment explores one of the interesting applications of the 2D wave equation. With some approximations, solutions were found that resemble the standing wave patterns in a Chladni plate. However, as this paper demonstrates, modeling physical phenomena accurately can be complicated. This is why more sophisticated modeling methods such as FEA are often used. To improve the solutions found in this paper, perhaps a different modeling approach other than the 2D wave equation could be used that better accounts for the physical intricacies of the Chladni plate. Still, this paper demonstrates how to solve a 2D wave equation and displays the beautiful solutions that result.

ACKNOWLEDGMENTS

Thanks to Clark Snelgrove, the Chladni plate expert in the BYU demo lab 😊

REFERENCES

[1] Coughlin, A. M. (2000). (tech.). *Patterns in the Sand: A Mathematical Exploration of Chladni Patterns* (1st ed., Vol. 3, pp. 131–146). The Journal of Mathematics and Science.

[2] Shu, Y.-H., Tseng, Y.-C., Lai, Y.-H., Yu, Y.-T., Huang, K.-F., & Chen, Y.-F. (2022). Exploring the origin of maximum entropy states relevant to resonant modes in modern Chladni plates. *Entropy*, *24*(2), 215. https://doi.org/10.3390/e24020215

APPENDIX

Code used the generate solution plots:

```
clear; close all; clc;

% Dimensions in meters
L = 8.75*.0254;
M = 8.75*.0254;

% Frequency in Hz
omega =  7* pi / M;

% Time
t_fixed = 4;

% Other parameters
damping_adjustment = 0.02;
v = 0.5;
s0 = 0.01;

% Grid for x and y
x = linspace(0, L, 25);
y = linspace(0, M, 25);
[X, Y] = meshgrid(x, y);

% Calculate the solution over the grid
Z = zeros(length(x), length(y));  % Initialize Z as a 2D array
for i = 1:length(x)
    for j = 1:length(y)
        Z(i, j) = analytical_solution(X(i, j), Y(i, j), t_fixed,
 damping_adjustment, v, s0, L, M, omega);
    end
end
% Plotting as a contour plot
figure('Color', 'black');  % Set the background color of the figure
contour_plot = contour(X, Y, Z', 'LevelList',
 0, 'LineColor', 'white','linewidth',3);
% figure; surf(X,Y,Z)
xlabel('X axis', 'Color', 'white');  % Set X axis label color to white
ylabel('Y axis', 'Color', 'white');  % Set Y axis label color to white
title(['Analytical Solution at w = ', num2str(omega), '
 Hz'], 'Color', 'white');  % Set title color to white
set(gca, 'Color', 'black');  % Set the axes background color to black
set(gca, 'XColor', 'white', 'YColor', 'white');  % Set the color of the axes
 to white

function result = analytical_solution(x, y, t, gamma, v, s0, L, M, omega)
    sum_result = 0;
    n_range = 20;  % Number of terms in the summation for n
    m_range = 20;  % Number of terms in the summation for m

    for n = 1:n_range
        for m = 1:m_range
            mu = n * pi / L;
            lamda = m * pi / M;
            X_n = cos(mu * x);
            Y_m = cos(lamda * y);
            phi = s0 * cos(mu * L / 2) * cos(lamda * M / 2);
            beta = sqrt(mu^2 * v^2 + lamda^2 * v^2 - gamma^4 * v^2);

            integrand = @(tau) sin(omega * (tau - t)) .* exp(-gamma^2 * v^2 *
 tau) .* sin(beta * tau);
            integral_results = integral(integrand, 0, t);
            sum_result = sum_result + v^2 * phi / beta * X_n * Y_m *
 integral_results * 4 / (L * M);
        end
    end

    result = sum_result;
end
```

