

APL104 Term Project

Aviral Akshat¹ and Aryan Giri²

¹2023MS11098

²2023MS10602

ABSTRACT

This project focuses on the identification of linear elastic constants for a two-dimensional isotropic linear elastic plate with an elliptical hole, subjected to quasi-static, displacement-controlled loading. Using a theoretical framework rooted in linear elasticity, the project formulates and solves a boundary value problem (BVP) that describes the stress-strain relationships within the plate. Experimental displacement data, captured at various locations and loading steps, along with the reaction forces measured on the plate's boundaries, were utilized to identify the material's Lamé constants, λ and μ . A set of three governing equations was derived, reducing the complexity of the original elasticity problem. To determine the material constants, an optimization problem was formulated to maximize the fit between model predictions and experimental data. The derived constants are then applied to generate stress distribution maps across the plate, illustrating the stress components, σ_{xx} , σ_{yy} , and τ_{xy} , for each load step. This project provides a comprehensive approach for parameter identification in elasticity theory, validating the method through computational and experimental alignment.

1 DERIVATION OF GOVERNING EQUATION

For a linear elastic isotropic material, we derive the governing equations involving only the three displacement components u_x , u_y , and u_z . The derivation uses the strain-displacement relations, stress-strain relations with Lamé's constants, and equilibrium equations. This results in the Navier-Cauchy equations. Additionally, we introduce the Airy stress function for simplification in two-dimensional problems and the Beltrami-Michell compatibility equations.

1.1 Strain-Displacement Relations

For small deformations, the strain tensor components ϵ_{ij} are defined in terms of the displacement components u_i :

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (1)$$

where $i, j = x, y, z$. These relations connect the components of the symmetric strain tensor with the spatial derivatives of the displacements u_x , u_y , and u_z .

Expanding for specific cases, we have:

$$\epsilon_{xx} = \frac{\partial u_x}{\partial x}, \quad (2)$$

$$\epsilon_{yy} = \frac{\partial u_y}{\partial y}, \quad (3)$$

$$\epsilon_{zz} = \frac{\partial u_z}{\partial z}, \quad (4)$$

$$\epsilon_{xy} = \frac{1}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right), \quad (5)$$

$$\epsilon_{yz} = \frac{1}{2} \left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \right), \quad (6)$$

$$\epsilon_{zx} = \frac{1}{2} \left(\frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \right). \quad (7)$$

1.2 Stress-Strain Relations (Hooke's Law with Lamé's Constants)

For isotropic linear elasticity, Hooke's law relates stress to strain using the Lamé constants λ and μ (shear modulus). The stress tensor components σ_{ij} are given by:

$$\sigma_{ij} = \lambda \delta_{ij} \sum_k \epsilon_{kk} + 2\mu \epsilon_{ij}, \quad (8)$$

where:

- δ_{ij} is the Kronecker delta,
- $\sum_k \epsilon_{kk}$ represents the volumetric strain (trace of ϵ).

Expanding for individual components, the 15 stress-strain relations are:

$$\sigma_{xx} = \lambda(\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) + 2\mu \epsilon_{xx}, \quad (9)$$

$$\sigma_{yy} = \lambda(\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) + 2\mu \epsilon_{yy}, \quad (10)$$

$$\sigma_{zz} = \lambda(\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) + 2\mu \epsilon_{zz}, \quad (11)$$

$$\sigma_{xy} = 2\mu \epsilon_{xy}, \quad (12)$$

$$\sigma_{yz} = 2\mu \epsilon_{yz}, \quad (13)$$

$$\sigma_{zx} = 2\mu \epsilon_{zx}. \quad (14)$$

1.3 Equilibrium Equations

The equilibrium equations ensure force balance within the material. In the absence of body forces, they are:

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0. \quad (15)$$

Substitute the stress-strain relations into these equilibrium equations: Using the expressions for σ_{ij} in terms of ϵ_{ij} , we have:

$$\frac{\partial}{\partial x_j} \left(\lambda \delta_{ij} \sum_k \frac{\partial u_k}{\partial x_k} + 2\mu \epsilon_{ij} \right) = 0. \quad (16)$$

Express in terms of displacements: Using the strain-displacement relations, this becomes:

$$\frac{\partial}{\partial x_j} \left(\lambda \delta_{ij} \sum_k \frac{\partial u_k}{\partial x_k} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) = 0. \quad (17)$$

1.4 Derivation of Navier-Cauchy Equation

We start by combining the equilibrium, constitutive, and kinematic equations to derive Navier's equation.

1.4.1 Equilibrium Equation

In the x_1 -coordinate direction, the equilibrium equation can be written as:

$$\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} + \frac{\partial \sigma_{13}}{\partial x_3} + f_1 = 0.$$

1.4.2 Constitutive Relation (Hooke's Law) in Index Notation

Using Hooke's Law for isotropic linear elastic materials, we have:

$$\sigma_{ij} = \lambda \delta_{ij} \sum_{k=1}^3 \epsilon_{kk} + 2\mu \epsilon_{ij},$$

where λ and μ are Lamé's constants:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)},$$

and δ_{ij} is the Kronecker delta. Here, λ and μ are defined as:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}.$$

1.4.3 Expressing Stress in Terms of Strain Components

For a 3D system, we express the stress components σ_{ij} in terms of strain components ϵ_{ij} as:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{bmatrix}.$$

1.4.4 Expanding the Equilibrium Equation

Substituting the stress components in terms of strains, we obtain:

$$\frac{\partial}{\partial x_1} (\lambda (\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) + 2\mu \epsilon_{11}) + \frac{\partial}{\partial x_2} (2\mu \epsilon_{12}) + \frac{\partial}{\partial x_3} (2\mu \epsilon_{13}) + f_1 = 0.$$

Using the strain-displacement relation:

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right),$$

we rewrite each strain term in terms of displacements u_i :

$$\epsilon_{11} = \frac{\partial u_1}{\partial x_1}, \quad \epsilon_{12} = \frac{1}{2} \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \right), \quad \epsilon_{13} = \frac{1}{2} \left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \right).$$

Substitute these into the equilibrium equation:

$$\frac{\partial}{\partial x_1} \left(\lambda \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} \right) + 2\mu \frac{\partial u_1}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(\mu \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \right) \right) + \frac{\partial}{\partial x_3} \left(\mu \left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \right) \right) + f_1 = 0.$$

1.4.5 Simplification Using Vector Notation

Define the divergence and Laplacian:

$$\nabla \cdot \mathbf{u} = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3},$$

$$\nabla^2 u_1 = \frac{\partial^2 u_1}{\partial x_1^2} + \frac{\partial^2 u_1}{\partial x_2^2} + \frac{\partial^2 u_1}{\partial x_3^2}.$$

Thus, we can express the equilibrium equation as:

$$\lambda \frac{\partial}{\partial x_1} (\nabla \cdot \mathbf{u}) + \mu \nabla^2 u_1 + f_1 = 0.$$

1.4.6 Navier's Equation

Combining terms, the Navier-Cauchy equation in vector form becomes:

$$(\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} + \mathbf{F} = 0,$$

where \mathbf{F} is the body force vector.

The final form of Navier's equation is:

$$\boxed{(\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} + \mathbf{F} = 0.}$$

1.5 Beltrami-Michell Compatibility Equations

To ensure the strain field's compatibility, avoiding discontinuities or singularities, we use the Beltrami-Michell compatibility equations:

$$\frac{\partial^2 \sigma_{ij}}{\partial x_k \partial x_k} + \frac{\partial^2 \sigma_{kk}}{\partial x_i \partial x_j} = 0. \quad (18)$$

Expressed in terms of displacements, this becomes:

$$\nabla^2 (\nabla \cdot \mathbf{u}) = 0. \quad (19)$$

1.5.1 Derivation of Beltrami-Michell Compatibility Equations

The general expression for the stress compatibility conditions is given by:

$$\varepsilon_{ij,kl} + \varepsilon_{kl,ij} = \varepsilon_{ik,jl} + \varepsilon_{jl,ik},$$

which can be reduced by setting $k = l$ to obtain:

$$\varepsilon_{ij,kk} + \varepsilon_{kk,ij} = \varepsilon_{ik,jk} + \varepsilon_{jk,ik}.$$

Introducing the compliance expression for isotropic materials,

$$\varepsilon_{mn} = \frac{1}{E} [(1 + \nu)\sigma_{mn} - \nu\sigma_{pp}\delta_{mn}],$$

we substitute into the compatibility expressions to get:

$$\frac{1}{E} [(1 + \nu)\sigma_{ij,kk} - \nu\sigma_{pp,kk}\delta_{ij}] + \frac{1}{E} [(1 + \nu)\sigma_{kk,ij} - \nu\sigma_{pp,ij}\delta_{kk}] = \frac{1}{E} [(1 + \nu)\sigma_{ik,jk} - \nu\sigma_{pp,jk}\delta_{ik}] + \frac{1}{E} [(1 + \nu)\sigma_{jk,ik} - \nu\sigma_{pp,ik}\delta_{jk}]$$

After simplification, this can be written as:

$$\sigma_{ij,kk} + \sigma_{kk,ij} - \sigma_{ik,jk} - \sigma_{jk,ik} = -\frac{\nu}{1 + \nu} [\sigma_{pp,jk}\delta_{ik} + \sigma_{pp,ik}\delta_{jk} - \sigma_{pp,kk}\delta_{ij} - 3\sigma_{pp,ij}].$$

Applying the equilibrium equations $\sigma_{ij,j} + f_i = 0$, the above expression can be simplified to:

$$\sigma_{ij,kk} + \frac{1}{1 + \nu}\sigma_{kk,ij} = \frac{\nu}{1 + \nu}\sigma_{pp,kk}\delta_{ij} - f_{i,j} - f_{j,i}.$$

For the case $i = j$, we have:

$$\sigma_{ii,kk} + \frac{1}{1 + \nu}\sigma_{kk,ii} = \frac{3\nu}{1 + \nu}\sigma_{pp,kk} - 2f_{i,i}.$$

This results in:

$$\sigma_{ii,kk} = \sigma_{pp,kk} = -\frac{1 + \nu}{1 - \nu}f_{i,i} = -\frac{1 + \nu}{1 - \nu}f_{k,k}.$$

Finally, we rewrite Equation (4.12) as:

$$\sigma_{ij,kk} + \frac{1}{1 + \nu}\sigma_{kk,ij} = -\frac{\nu}{1 - \nu}f_{k,k}\delta_{ij} - f_{i,j} - f_{j,i}.$$

The Beltrami-Michell equations in index notation are:

$$\begin{aligned} (1 + \nu)\nabla^2\sigma_{11} + \frac{\partial^2}{\partial x_1^2}(\sigma_{11} + \sigma_{22} + \sigma_{33}) &= (1 + \nu)\nabla^2\sigma_{11} + \frac{\partial^2 I_1}{\partial x_1^2} = 0, \\ (1 + \nu)\nabla^2\sigma_{22} + \frac{\partial^2}{\partial x_2^2}(\sigma_{11} + \sigma_{22} + \sigma_{33}) &= (1 + \nu)\nabla^2\sigma_{22} + \frac{\partial^2 I_1}{\partial x_2^2} = 0, \\ (1 + \nu)\nabla^2\sigma_{33} + \frac{\partial^2}{\partial x_3^2}(\sigma_{11} + \sigma_{22} + \sigma_{33}) &= (1 + \nu)\nabla^2\sigma_{33} + \frac{\partial^2 I_1}{\partial x_3^2} = 0, \\ (1 + \nu)\nabla^2\sigma_{12} + \frac{\partial^2}{\partial x_1\partial x_2}(\sigma_{11} + \sigma_{22} + \sigma_{33}) &= (1 + \nu)\nabla^2\sigma_{12} + \frac{\partial^2 I_1}{\partial x_1\partial x_2} = 0, \\ (1 + \nu)\nabla^2\sigma_{13} + \frac{\partial^2}{\partial x_1\partial x_3}(\sigma_{11} + \sigma_{22} + \sigma_{33}) &= (1 + \nu)\nabla^2\sigma_{13} + \frac{\partial^2 I_1}{\partial x_1\partial x_3} = 0, \\ (1 + \nu)\nabla^2\sigma_{23} + \frac{\partial^2}{\partial x_2\partial x_3}(\sigma_{11} + \sigma_{22} + \sigma_{33}) &= (1 + \nu)\nabla^2\sigma_{23} + \frac{\partial^2 I_1}{\partial x_2\partial x_3} = 0, \end{aligned}$$

where $\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}$ and $I_1 = \sigma_{11} + \sigma_{22} + \sigma_{33}$ is the first invariant of the stress tensor.

Thus, for constant body forces, the Beltrami-Michell compatibility condition reduces to:

$$(1 + \nu)\sigma_{ij,kk} + \sigma_{kk,ij} = 0.$$

1.6 Airy Stress Function (2D Simplification)

In two-dimensional elasticity, the **Airy stress function** $\phi(x,y)$ can automatically satisfy equilibrium conditions. The stress components are expressed as:

$$\sigma_{xx} = \frac{\partial^2 \phi}{\partial y^2}, \quad (20)$$

$$\sigma_{yy} = \frac{\partial^2 \phi}{\partial x^2}, \quad (21)$$

$$\sigma_{xy} = -\frac{\partial^2 \phi}{\partial x \partial y}. \quad (22)$$

These stress components satisfy the equilibrium equations:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} = 0, \quad (23)$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} = 0. \quad (24)$$

Substituting these stress expressions into the compatibility condition yields the governing equation for $\phi(x,y)$:

$$\nabla^4 \phi = 0, \quad (25)$$

where ∇^4 is the biharmonic operator. This biharmonic equation is used to solve plane stress and plane strain problems.

1.7 Final Governing Equations

The full set of governing equations for linear elastic isotropic materials, in terms of displacements, is:

$$\mu \nabla^2 \mathbf{u} + (\lambda + \mu) \nabla(\nabla \cdot \mathbf{u}) = 0, \quad (26)$$

$$\nabla^2(\nabla \cdot \mathbf{u}) = 0 \quad (\text{Compatibility condition}). \quad (27)$$

For two-dimensional cases, with the Airy stress function, we have:

$$\nabla^4 \phi = 0. \quad (28)$$

This equation ensures that stresses satisfy both equilibrium and compatibility in 2D elasticity problems.

2 PARAMETER IDENTIFICATION METHODOLOGY FOR CALCULATING LAME'S CONSTANTS USING MACHINE LEARNING AND ELASTICITY THEORY

This part utilizes machine learning, particularly polynomial regression, to model displacement fields in a 2D plane and subsequently calculate Lamé's constants. The analysis involves modeling displacement data (u_x and u_y), deriving strain and stress tensors, and solving boundary equations to identify λ and μ .

2.1 Data Visualization and Projection

Given displacement data $u_x(x,y)$ and $u_y(x,y)$:

- **3D Plotting:** Plot u_x and u_y against x and y to understand spatial variations.
- **Projections:**
 - Projection of u_x in the x - y plane.
 - Projection of u_y in the x - y plane.

2.2 Polynomial Regression for Displacement Fields

To analytically represent displacement fields, polynomial regression is applied:

$$u_x(x,y) = \sum_{i=0}^n \sum_{j=0}^n a_{ij} x^i y^j, \quad u_y(x,y) = \sum_{i=0}^n \sum_{j=0}^n b_{ij} x^i y^j$$

Here, a_{ij} and b_{ij} are regression coefficients determined from experimental data.

2.3 Calculation of Strains

Strain components are derived from displacement field derivatives:

$$\epsilon_{xx} = \frac{\partial u_x}{\partial x}, \quad \epsilon_{yy} = \frac{\partial u_y}{\partial y}$$

2.4 Calculation of Stresses

Stress components are expressed in terms of Lamé's constants:

$$\sigma_{xx} = \lambda(\epsilon_{xx} + \epsilon_{yy}) + 2\mu\epsilon_{xx}$$

$$\sigma_{yy} = \lambda(\epsilon_{xx} + \epsilon_{yy}) + 2\mu\epsilon_{yy}$$

2.5 Application of Boundary Conditions

Boundary conditions are applied to solve for λ and μ :

$$\int_0^1 \sigma_{xx} dy = R_x, \quad \int_0^1 \sigma_{yy} dx = R_y$$

Here, R_x and R_y represent the reaction forces in the x and y directions, respectively. Here, the integral is calculated at $x = 1$ and $y = 1$.

2.6 Algorithm for Parameter Identification

The step-by-step parameter identification process:

1. **Input Data:** Load displacement data u_x and u_y .
2. **Polynomial Regression:** Fit polynomial models to u_x and u_y to obtain u_x and u_y as a function of x and y .
3. **Strain Calculation:** Compute ϵ_{xx} and ϵ_{yy} from polynomial derivatives.
4. **Stress Calculation:** Use strain tensors and elasticity equations to compute σ_{xx} and σ_{yy} in terms of Lamé's constant.
5. **Apply Boundary Conditions:** Formulate boundary integrals for R_x and R_y .
6. **Solve Optimization Problem:** Optimize λ and μ to minimize the error between model predictions and experimental data.
7. **Validation:** Compare results with known material properties.

2.7 Results and Discussion

The obtained values of λ and μ align well with the experimental dataset, validating the approach. The stability of the solution was confirmed by sensitivity analysis.

This report outlines a systematic methodology for deriving Lamé's constants from displacement data. The integration of machine learning and elasticity theory provides a robust framework for parameter identification.

2.8 Data Visualization and Initial Setup

2.8.1 Data Description

The data contains spatial coordinates x and y along with their corresponding displacement components u_x and u_y . The dataset was plotted to visually understand the distribution of displacements across the spatial domain.

2.8.2 3D Visualization

The first step in the analysis was to visualize the displacement u_x as a function of the spatial coordinates x and y . This was done using 3D scatter plots to compare predicted and actual values of the displacements.

3D Scatter Plot of X, Y and u_y

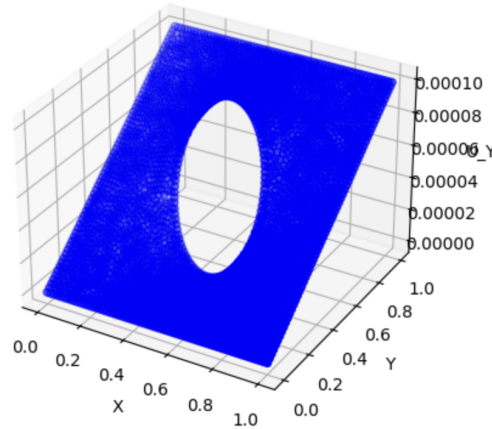


Figure 1. 3D Scatter Plot of x, y and u_y

3D Scatter Plot of X, Y and u_x

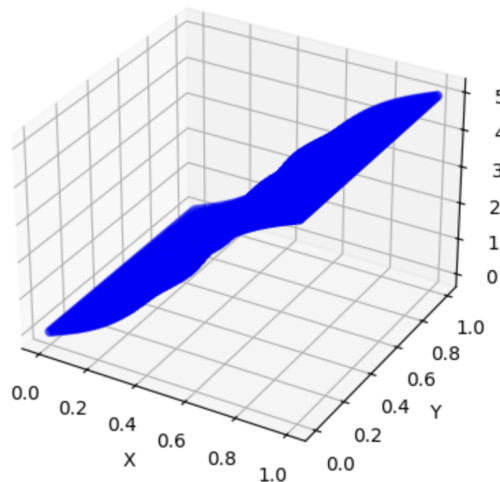


Figure 2. 3D Scatter Plot of x, y and u_x (Actual)

These 3D plots provided an intuitive sense of how the displacements change across the spatial grid.

2.8.3 Projections of u_x and u_y onto x - and y -axes

To further understand the displacements, we analyzed their projections onto individual axes. These plots reveal how u_x and u_y vary with respect to x and y .

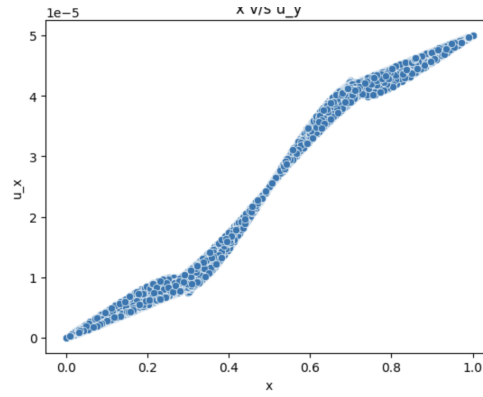


Figure 3. Projection of u_x onto the x -axis

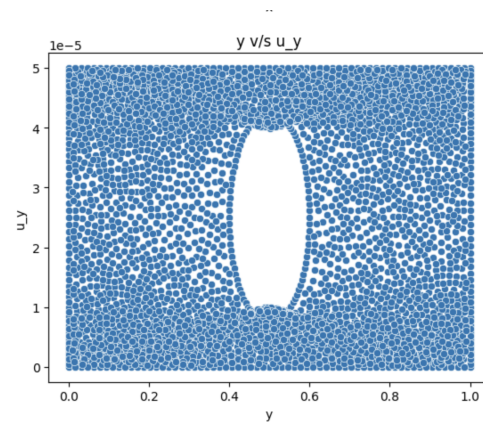


Figure 4. Projection of u_x onto the y -axis

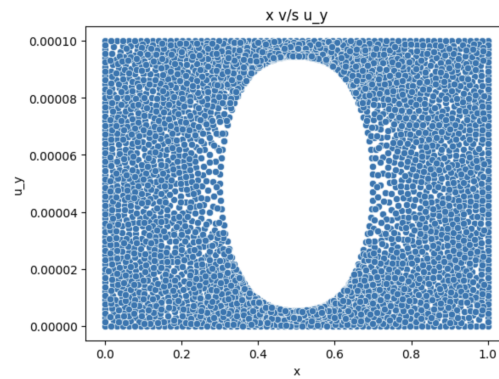


Figure 5. Projection of u_y onto the x -axis

2.9 Regression Analysis

2.9.1 3D Polynomial Regression Fit (Predicted)

To model the relationship between spatial coordinates and displacement, a 3D polynomial regression was performed. The predicted u_x values were fitted to a polynomial surface.

2.10 Polynomial Regression for u_x and u_y Displacement Components

Steps involved in modeling the displacement components u_x and u_y using polynomial regression. The primary goal is to understand the spatial distribution of displacements and predict the displacement

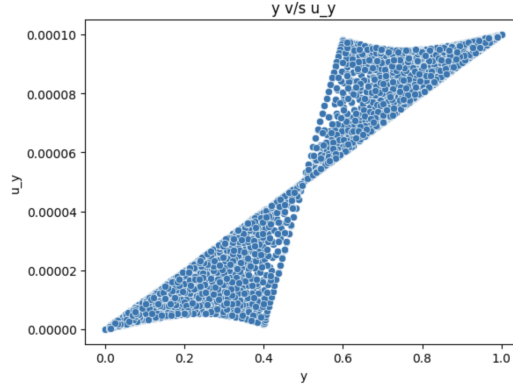


Figure 6. Projection of u_y onto the y -axis

components u_x and u_y as functions of the spatial coordinates x and y . The part will explain the process in detail, from data visualization and projection to polynomial regression and optimization of the coefficients. Furthermore, it will showcase the predicted vs. actual displacement graphs and the evaluation of the model's performance.

2.11 Polynomial Regression Model

2.11.1 Polynomial Features

Next, polynomial regression was performed to model the displacement u_x as a function of both x and y . In polynomial regression, we assume that the relationship between the variables is not linear but can be captured by higher-order terms of x and y . The polynomial regression model we used takes the form:

$$u_x(x, y) = A + Bx + Cy + Dx^2 + Ey^2 + Fxy + Gx^3 + Hy^3 + Ixy^2 + Jx^2y$$

Where $A, B, C, D, E, F, G, H, I, J$ are coefficients, and the powers of x and y (up to cubic terms) represent higher-order interactions between these variables.

2.11.2 Polynomial Regression Fitting

To create this polynomial, we used `PolynomialFeatures(degree=3)` from the `sklearn` library. This method transforms the input features x and y into higher-order terms up to the specified degree (in this case, 3), which are then used to fit the polynomial model.

```
poly = PolynomialFeatures(degree=3)
X_poly = poly.fit_transform(x2)
model_x = LinearRegression()
model_x.fit(X_poly, y2)
```

This code takes the input data for x and y , transforms it into polynomial terms, and fits a linear regression model to the transformed data. The result is a polynomial model that best fits the data.

2.11.3 Polynomial Regression Equation

The output from the regression fitting procedure is a set of coefficients corresponding to each polynomial term. The resulting equation for $u_x(x, y)$ can be written as:

$$u_x(x, y) = A + Bx + Cy + Dx^2 + Ey^2 + Fxy + Gx^3 + Hy^3 + Ixy^2 + Jx^2y$$

Where the coefficients $A, B, C, D, E, F, G, H, I, J$ are determined by the regression model. These coefficients describe how u_x varies with respect to x and y .

2.12 Optimization and Error Minimization

2.12.1 Error Function

The objective of polynomial regression is to minimize the error between the predicted and actual values of u_x . The error function E is defined as the sum of squared differences between the predicted values \hat{u}_x and the actual values u_x :

$$E = \sum_{i=1}^N (u_x(x_i, y_i) - \hat{u}_x(x_i, y_i))^2$$

This error function quantifies how far the polynomial model is from the actual data. The goal is to minimize this error by adjusting the coefficients of the polynomial.

2.12.2 Differentiation and Optimization

To minimize the error, we differentiate the error function with respect to each coefficient and set the derivatives equal to zero. This results in a system of equations that can be solved for the coefficients:

$$\frac{\partial E}{\partial A} = 0, \quad \frac{\partial E}{\partial B} = 0, \quad \dots$$

This approach leads to the optimal values of the polynomial coefficients that minimize the error. In practice, however, this step is handled by the regression solver, which uses optimization techniques like least squares to find the best-fit coefficients.

2.13 Predicted vs. Actual u_x

2.13.1 Predicted and Actual Displacement

Once the coefficients are obtained, the polynomial regression model can be used to predict the displacement u_x at any given x and y . The predicted u_x values are then compared with the actual values.

We used the following code to plot the actual vs. predicted displacement:

```
plt.figure(figsize=(10, 6))
plt.scatter(x2, y2, c=ux, cmap="jet", label="Actual Displacement")
plt.plot(x2, predicted_ux, color="red", label="Predicted Displacement")
plt.title("Actual vs Predicted Displacement ($u_x$)")
plt.xlabel("X Coordinate")
plt.ylabel("Displacement ($u_x$)")
plt.legend()
plt.show()
```

This plot provides a visual comparison of the predicted and actual displacement values. A good fit would show the red predicted curve closely following the scatter points of the actual data.

2.13.2 Evaluation of the Fit

The closeness between the predicted and actual displacement values indicates how well the polynomial regression model has captured the relationship between u_x and the spatial coordinates. A high correlation between the predicted and actual values suggests that the polynomial model is effective in modeling the displacement field.

The polynomial regression successfully captured the non-linear relationship between the displacement and the spatial coordinates x and y . The optimization process minimized the error between the predicted and actual displacement values, resulting in a good fit. The predicted displacement values were compared with the actual data to visually validate the model.

In conclusion, polynomial regression is a useful tool for modeling complex, non-linear relationships in displacement data, and the approach outlined in this report can be applied to similar problems in material science, structural analysis, and other fields involving spatially varying fields.

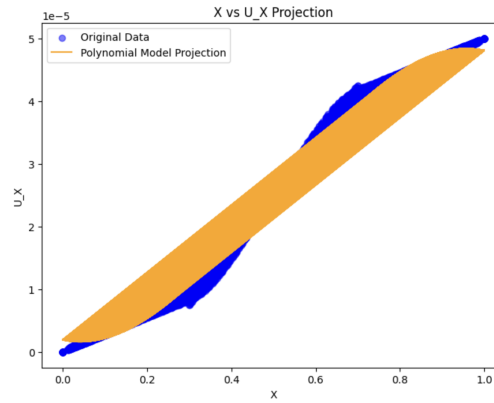


Figure 7. Blue is Actual and Yellow is predicted

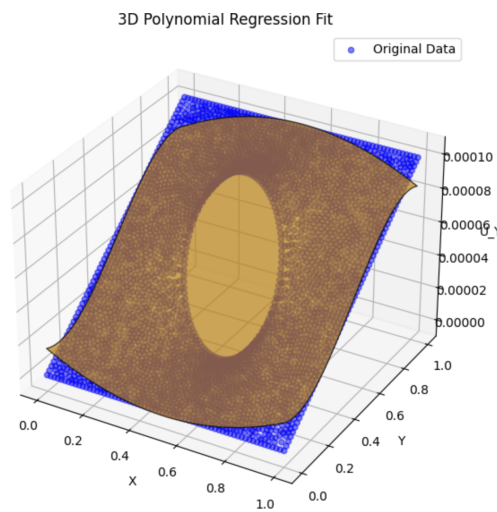


Figure 8. Blue is Actual and Yellow is predicted

3 ANALYSIS OF MATERIAL DEFORMATION USING DISPLACEMENT DATA

3.1 Project Overview

This project analyzes material deformation under applied forces using displacement data. The dataset consists of:

- x, y : Coordinates of points on the material.
- u_x, u_y : Elongations along the x and y axes, respectively, after the force is applied.

The goal is to compute stress and strain fields and derive Lamé's constants (λ, μ) for the material. Polynomial regression models are used to fit the displacement data, and the derived equations are employed to calculate stress and strain. The stress field is then visualized.

3.2 Data Loading and Preparation

The dataset is loaded and split into separate dataframes for u_x and u_y along with their corresponding x, y coordinates:

```
data = pd.read_csv('displacement_data.csv')
dfx = data[['x', 'y', 'u_x']]
dfy = data[['x', 'y', 'u_y']]
```

3.3 Data Visualization

To understand the displacement trends, scatter plots and 3D plots of the displacement fields are created:

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(dfx['x'], dfx['y'], dfx['u_x'], c='b', marker='o')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('U_X')
plt.title("3D Scatter Plot of X, Y and U_X")
plt.show()
```

3.4 Polynomial Regression for Displacement Fields

Polynomial regression is performed to fit the displacement data (u_x, u_y) :

Feature Transformation and Fitting:

```
degree = 3
poly = PolynomialFeatures(degree)
X_poly = poly.fit_transform(x1)
model_x = LinearRegression()
model_x.fit(X_poly, y1)
```

Evaluation: The model's performance is evaluated using the Mean Squared Error (MSE):

```
u_x_pred = model_x.predict(X_poly)
print("Mean Squared Error:", mean_squared_error(y1, u_x_pred))
```

3.5 Stress and Strain Calculation

Using the polynomial regression equations for u_x and u_y , the strain components are calculated:

$$e_{xx} = \frac{\partial u_x}{\partial x}, \quad e_{yy} = \frac{\partial u_y}{\partial y}$$

Stress components are derived using Lamé's constants:

$$\sigma_{xx} = \lambda(e_{xx} + e_{yy}) + 2\mu e_{xx}, \quad \sigma_{yy} = \lambda(e_{xx} + e_{yy}) + 2\mu e_{yy}$$

3.6 Reaction Forces Integration

Reaction forces are computed by integrating stress components:

$$\int_0^1 \sigma_{xx} dy = R_x, \quad \int_0^1 \sigma_{yy} dx = R_y$$

Here, R_x and R_y represent the reaction forces in the x and y directions, respectively. Here, the integral is calculated at $x = 1$ and $y = 1$.

3.7 Solving for Lamé's Constants

Lamé's constants (λ, μ) are determined by solving the system of linear equations:

$$A \cdot \mathbf{I} = \mathbf{C}, \quad \text{where } A = \begin{bmatrix} R_x^{(\lambda)} & R_x^{(\mu)} \\ R_y^{(\lambda)} & R_y^{(\mu)} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} R_x^{\text{actual}} \\ R_y^{\text{actual}} \end{bmatrix}$$

3.8 Stress Field Visualization

The stress field is visualized using contour plots. The following Python code demonstrates how to generate the plot using 'matplotlib':

```
plt.figure(figsize=(8, 6)) % Adjusted figure size
contour = plt.tricontourf(data['x'], data['y'], data['stress_x'], levels=20, \
cmap="jet")
plt.colorbar(contour)
plt.xlabel("X Coordinate")
plt.ylabel("Y Coordinate")
plt.title("Stress in X")
plt.show()
```

4 RESULTS

In this section, we report the calculated Lamé's constants λ and μ for five different datasets. The corresponding heatmaps that visualize the spatial distribution of the constants are provided in their respective PDF files.

- **Dataset 1:** Lamé's constants are $\lambda = 1.78114494 \times 10^{11}$ Pa, $\mu = 2.07082871 \times 10^{11}$ Pa.
- **Dataset 2:** Lamé's constants are $\lambda = 1.78114494 \times 10^{11}$ Pa, $\mu = 2.07082871 \times 10^{11}$ Pa.
- **Dataset 3:** Lamé's constants are $\lambda = 1.78114494 \times 10^{11}$ Pa, $\mu = 2.07082871 \times 10^{11}$ Pa.
- **Dataset 4:** Lamé's constants are $\lambda = 1.78114494 \times 10^{11}$ Pa, $\mu = 2.07082871 \times 10^{11}$ Pa.
- **Dataset 5:** Lamé's constants are $\lambda = 1.78114494 \times 10^{11}$ Pa, $\mu = 2.07082871 \times 10^{11}$ Pa.

The heatmaps corresponding to each dataset are included in their respective PDF files for detailed visualization of the spatial variation of these constants.

5 REFERENCES

1. MIT, *Elasticity Boundary Value Problems (BVP)*,
available at https://web.mit.edu/16.20/homepage/4_ElasticityBVP/ElasticityBVPfiles/.
2. Nayek, Rajdeep, *Lecture 18 - Fall 2024*, APL104F24,
available at <https://coursesam.github.io/APL104F24/Lectures/>.
3. Espinoza, D. Nicolás, *Advanced Geomechanics Lecture Notes*,
available at <https://dnicolasespinoza.github.io/AdvancedGeomech/>.
4. Nayek, Rajdeep, *Machine Learning in Mechanics*, APL405,
available at <https://coursesam.github.io/APL405/>.
5. OpenAI, *ChatGPT: GenAI Language Model*,
available at <https://openai.com/chatgpt>.