Modeling internal defects in conductive materials using electric potential measurements

Dia Avalur, Nitika Kishore, and Anika Tokala

***Abstract*—In this project, a numerical model was developed to study how electric potential behaves inside a conductive material and how internal defects can affect its electrical response. The material was represented as a two-dimensional grid, where each grid corresponds to an electric potential. Using basic principles of electrical conduction, the system was modeled under steady-state conditions and solved numerically using a finite-difference approach to Laplace's equation. Boundary conditions were applied by setting one side of the grid to a high potential while the other sides were grounded. After getting a steady-state solution, an internal defect was introduced by blocking a small region of the grid to simulate an insulating flaw. The resulting potential distributions were compared before and after the defect was added. The results showed that the presence of an internal defect caused noticeable changes in the potential pattern. These findings suggest that electrical measurements taken at the surface of a conductive object can give information about internal features, which supports the use of electrical methods or non-destructive testing.**

***Index Terms*—electric potential, Laplace's equation, finite-difference method, numerical modeling, non-destructive testing, equipotential lines, equipotentials, electric field visualizations, electric field mapping**

I. Introduction

ELECTRICAL conduction in a material is determined by the relationship between electric field, current flow, and electric potential, as described by Ohm's law [1]. Variations in electric potential create an electric field, and this drives current through a conductive medium [2]. Charge is conserved and does not accumulate within the material at steady state. This leads to an equation for the potential, known as Laplace's equation. To computationally study this behavior, the conductive region can be represented as a 2D grid with uniform spacing. Here, each grid point corresponds to the electric potential at that specific location. Boundary conditions are applied by fixing the potential on certain edges of the grid, such as holding one side at a high potential while keeping the other sides grounded [2]. Solving Laplace's equation on this grid would produce the steady state potential distribution and form the basis for analyzing how the internal features can change the system's electrical response.

Previous researchers examined conduction in a saline solution meant to simulate a low-cost electrophoresis system for biological studies [3]–[7]. These studies used measurements of voltage to map the equipotentials and electric fields within a test geometry, however, none considered the use of computational methods. Here, we consider the finite difference method, in which derivatives are approximated as differences between potential values at neighboring computational nodes, e.g.

$$\frac{dV}{dx} \approx \frac{V_{n+1} - V_n}{x_{n+1} - x_n}. \tag{1}$$

Application of this principle to 2D conduction equations is described below.

As 3D printers capable of metal and conductive materials become more available, low voltage DC electrical probing has potential advantages as a non-destructive testing (NDT) technique. Conventional techniques used in industry rely on x-rays from radioactive sources, pentrant dyes, or ultrasound equipment to accomplish NDT [8], [9], which may not be feasible for small-scale 3D printing setups. Methods based on conductivity have been applied to larger geophysical problems as well [10].

II. Methods and Materials

To model the electric potential inside the conductive region, Laplace's equation was solved using a finite-difference method on a two-dimensional grid [11]. The region was evenly divided into grid points, with each point representing the electric potential at that place. Boundary conditions were applied by fixing the potential along the edges of the grid, with one side held at a higher potential than the rest.

In a steady-state conductive material, no charge accumulates within the region because all charge is conserved [2], [12]–[14]. The continuity equation expresses this condition:

$$\vec{\nabla} \cdot \vec{J} = 0, \tag{2}$$

where $\vec{J}$ is the current density. Current density and electrical field are related through Ohm's law, expressed in differential form as [2]

$$\vec{J} = \sigma \vec{E}, \tag{3}$$

and the electric field is related to the electric potential by

$$\vec{E} = -\vec{\nabla}\phi. \tag{4}$$

These relationships combine, assuming the material has uniform conductivity, giving Laplace's equation [2]:

$$\vec{\nabla}^2 \phi = 0, \tag{5}$$

where the zero on the right hand side of (5) indicates the absence of internal sources. This equation controls the electric potential throughout the conductive region. In discretized form, (5) becomes [11]:

$$\frac{V_E - 2V_X + V_W + V_N - 2V_X + V_S}{h^2} = 0, \tag{6}$$

Author for correspondence: 426davalur@frhsd.com
Authors are with the Science & Engineering Magnet Program, Manalapan High School, 20 Church Lane, Englishtown, NJ 07726, USA

10.64804/9wkhqz81

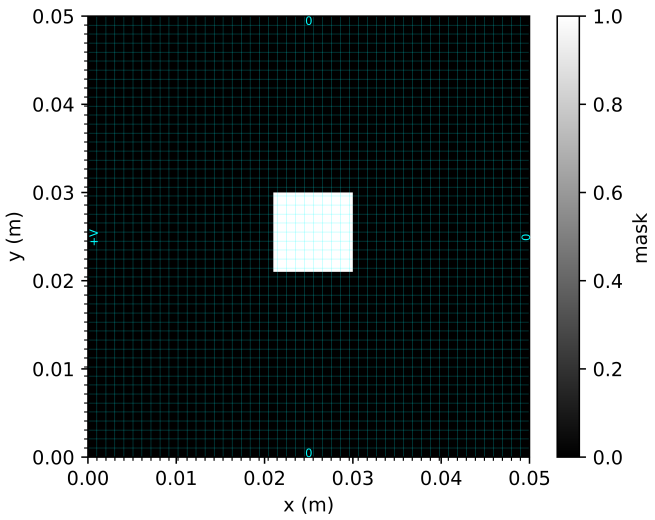


Fig. 1. Uniform regular square computational grid and pinned boundary conditions

$$4V_X = V_E + V_W + V_N + V_S, \tag{7}$$

where a given node point V_X is adjacent to neighbors north, south, east, and west on a regular grid of uniform spacing h. We iteratively calculated V by setting each interior point equal to the average of its four surrounding points. The process was repeated for 5000 iterations. As a convergence check during development, the mesh size was decreased and iterations increased tenfold with no change to results.

III. Computational Model

The conductive material was represented using a two-dimensional grid. The grid consisted of a 50×50 array, with the left boundary held at a high potential and the other boundaries grounded. Each point on the grid corresponds to the electric potential at that location, and the grid as a whole acts as a simplified model of the material. This allows the electric behavior of the system to be studied numerically instead of analytically.

Boundary conditions were applied by fixing the potential values along the edges of the grid. One side of the grid was set to a high potential to represent an applied voltage, while the remaining sides were set to ground. These fixed boundaries remained constant throughout the course of the simulation, while the interior grid points were free to update.

The potential values at the interior points were iteratively updated based on their surrounding neighbors until the grid itself reached a steady state. This steady-state solution represents how the electric potential would distribute itself throughout the material under the applied boundary conditions. By modifying the grid, such as introducing an internal blocked region to represent a defect, changes in the overall electrical response of the system can be observed. The computational grid and boundary conditions used in this model are shown in Fig. 1.

IV. Procedures

First, the basic physical principles controlling electrical conduction were identified. Ohm's law was used to relate current flow to the electric field, and the electric field was related to electric potential. Since charge does not accumulate in a steady-state system, these relationships combine and produce Laplace's equation, which describes how electric potential behaves inside a uniform conductive material.

To model this behavior computationally, the conductive material was represented using a Python program as a two-dimensional square grid. The grid was implemented as a 50×50 `numpy` array [15], where each element corresponds to the electric potential at a specific location in the material. Pinned boundary conditions were applied by directly assigning fixed potential values to the grid edges. One side of the grid was set to a high potential in order to represent an applied voltage, while the other sides were set to zero to represent ground. These boundary values were held constant throughout the simulation.

The interior points of the grid were solved using an iterative finite-difference method. In the beginning, the potential at every interior point was updated to equal the average of its four closest neighboring points. This update was performed using nested loops in the code, while skipping the boundary points so their values stayed fixed. The update process was repeated for several thousand iterations until the potential value changes became negligible, which indicated that a steady-state solution was reached.

Once the baseline solution was obtained, an internal defect was introduced into the model. This defect was represented by choosing a small square portion near the center of the grid and stopping those points from being updated during the iteration process. In the code, this was handled using a conditional statement that excluded the defect section from the averaging step. This region behaved as an insulating flaw within the material. The technique used here created a pinned boundary condition at the flaw with the nodes fixed at their initial values. Strictly speaking, this is not a no-flow boundary as might be expected for a nonconductive void; however, due to the location of the flaw at the centerline it behaves approximately the same way.

Lastly, the simulation results were visualized using `matplotlib` [16]. Heatmaps were generated to show the steady-state potential distribution for both the intact material and the material containing the defect. A difference plot was also created by subtracting the baseline potential grid from the defect grid. This showed how the defect changed the overall potential distribution.

Data and code are available at https://github.com/devangel77b/426davalur-lab10.

V. Results

After running the simulation, the electric potential across the grid settled into a smooth pattern from the high-voltage side toward the sides that are grounded. This is what we expect from a solution to Laplace's equation with pinned boundary conditions, and it showed that the numerical method was

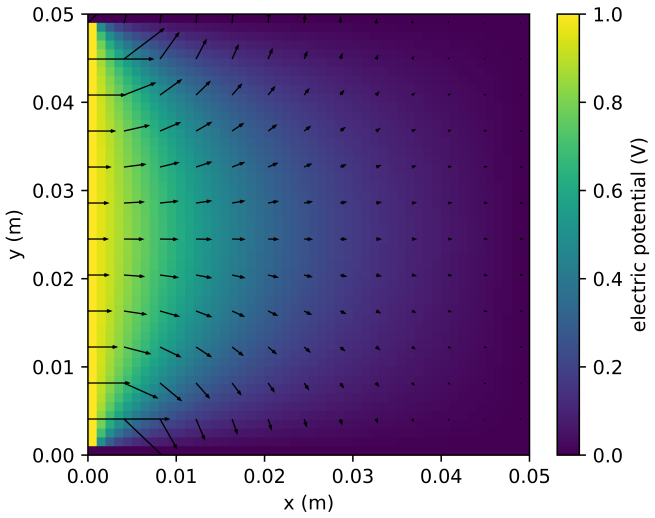


Fig. 2. Steady-state potential distribution (no defect) and electric field vectors

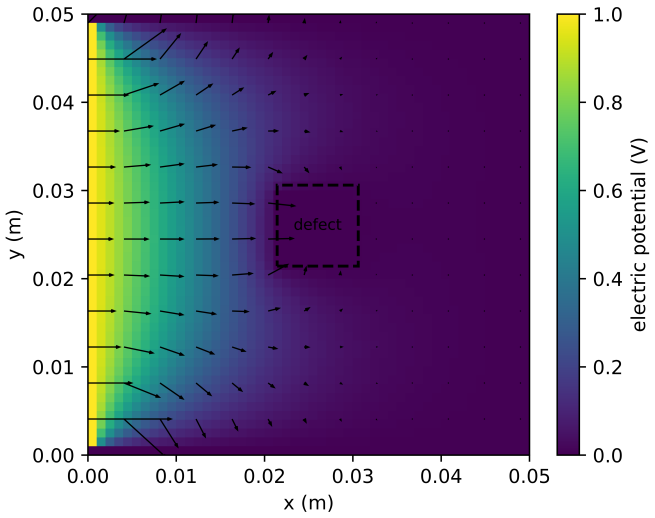


Fig. 3. Steady-state potential distribution and electric field vectors with internal defect

working correctly. The potential values stopped changing after multiple iterations, indicating that the system had reached a steady state. This electrical potential distribution obtained from the simulation is shown in Fig. 2.

When a defect was added by blocking a small portion of the grid, the potential pattern changed. The values around the defect were distorted, and this disturbance spread outward toward the grid edges. The resulting electric potential distribution with the defect present is shown in Fig. 3.

Comparing the boundary values before and after adding the defect showed noticeable differences in a region of about two- to three times the length of the flaw. This suggests that changes in the material can be detected by only looking at measurements taken along the surface.

VI. Discussion

The results of this simulation show that internal changes within a conductive material can influence the electrical behavior observed at its surface. Even though the defect was located in the grid, its presence still altered the potential near the boundaries. This is important because it means that information about the interior of a material can be inferred without directly accessing it, which is the main idea behind non-destructive testing [10], [17].

The distortion caused by the defect was not limited to the immediate area around it. Instead, the effect spread outward and affected the overall potential pattern to a distance of 2 to 3 times the length of the flaw. This suggests that electrical measurements taken at the surface are sensitive to internal features. Even when those features are not visible or are relatively small.

Even though this model is simplified, it captures the behavior of steady-state electrical conduction. The fact that a basic grid-based simulation is able to show these effects support the idea that electrical probing can be a useful tool for detecting internal mistakes. More detailed models could likely improve accuracy, but the trends observed are consistent with the expected physical behavior.

A. Limitations

This model uses many simplifying assumptions that limit how closely it represents a real physical system. The simulation is two-dimensional, while real conductive objects are three-dimensional, which means that some effects present in real materials are not captured here. The conductivity of the material is also assumed to be uniform, even though real materials can have variations that affect current flow. Also, the defect is represented as a completely blocked region, which is a simplified way of modeling an internal flaw. Lastly, the grid resolution is limited, so very detailed or small figures may not be accurately represented. These limitations are illustrated in Fig. 4, which shows that the effect of the defect appears as a broad change in the potential rather than a defined feature.

B. Detectability limits of internal defects

An important question is how defect size and material properties affect whether a flaw can be detected from surface measurements. In this model, the defect was big enough to clearly distort the potential, but smaller defects would produce weaker, more localized changes that may not be visible at the boundaries.

Material conductivity also plays a role. In highly conductive materials, current can redistribute more easily around a defect, reducing its effect on the surface potential. Also, this model does not include noise or measurement limitations, so it likely overestimates how detectable a defect would be in practice.

Overall, while defects do influence surface measurements, their detectability depends on both their size and the material properties, and smaller defects may not always be observable.

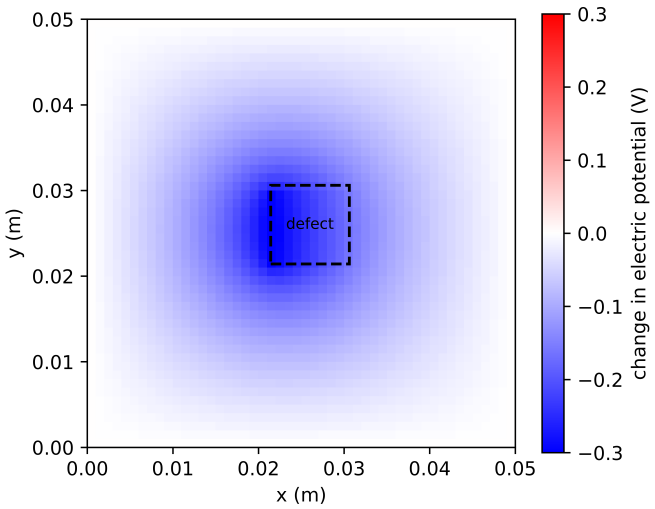


Fig. 4. Difference in electric potential due to an internal defect

C. Conclusion

This project showed that a simple numerical model can still reveal meaningful information about how electricity behaves in a conductive material. By modeling the material as a grid and solving for the steady-state potential, it became clear that internal changes, like an insulating defect, can affect the electrical behavior seen at the surface. Even with a basic setup, the defect caused noticeable distortions in the potential pattern, especially near the boundaries. This supports the idea that electrical measurements taken at the surface can be used to learn about what is happening inside a material. Overall, this model provides a clear example of how numerical methods can be used to explore non- destructive testing concepts in a straightforward and effective way.

APPENDIX

This code is available at https://github.com/devangel77b/426davalur-lab10. Relaxation is implemented at approximately line 60. Line 58 implements pinned nodes for a centerline defect.

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle



# ----------------------------
# Settings
# ----------------------------
N = 50 # grid dimensions N x N
ITERATIONS = 5000 # number of relaxation iterations
V_HIGH = 1.0 # high potential boundary value
V_GROUND = 0.0 # grounded boundary value
# Defect settings (squre block in the center)
DEFECT_HALF_WIDTH = 4 # defect will be (2*half_width+1) by (2*half_width+1)
# Physical size of square domain (meters)
Lx = 0.05
Ly = 0.05



# ----------------------------
# Helper: apply boundary conditions
# ----------------------------
def apply_boundary_conditions(V: np.ndarray) -> None:
    """
    Left boundary = high potential
    All other boundaries = grounded
    """
    V[:, 0] = V_HIGH # left = high
    V[:, -1] = V_GROUND # right = ground
    V[0, :] = V_GROUND # top = ground
    V[-1, :] = V_GROUND # bottom = ground









# ----------------------------
# Solver: finite difference relaxation
# ----------------------------
def solve_laplace(N: int, iterations: int, defect_mask: np.ndarray | None = None) -> np.ndarray:
    """
    Solves Laplace's equation on an N x N grid using iterative relaxation.
    If defect_mask is provided, True cells are treated as blocked and not updated.
    """
    V = np.zeros((N, N), dtype=float)
    apply_boundary_conditions(V)
    for _ in range(iterations):
        # update interior points
        for i in range(1, N - 1):
            for j in range(1, N - 1):
                if defect_mask is not None and defect_mask[i, j]:
                    continue
                V[i, j] = 0.25 * (
                    V[i + 1, j] +
                    V[i - 1, j] +
                    V[i, j + 1] +
                    V[i, j - 1]
                )
        # reapply boundary conditions each iteration just to be safe
        apply_boundary_conditions(V)
    return V









# ----------------------------
# Make defect mask
# ----------------------------
def make_center_defect_mask(N: int, half_width: int) -> np.ndarray:
    """
    Creates a square defect mask in the center.
    True = defect region
    """
    mask = np.zeros((N, N), dtype=bool)
    c = N // 2
    mask[
        c - half_width:c + half_width + 1,
        c - half_width:c + half_width + 1
    ] = True
    return mask









# ----------------------------
# Compute electric field
# ----------------------------
def compute_electric_field(V: np.ndarray, Lx: float, Ly: float):
    """
    Computes E = -grad(V)
    """
    dx = Lx / (V.shape[1] - 1)
    dy = Ly / (V.shape[0] - 1)
    dVdy, dVdx = np.gradient(V, dy, dx)
```

```
    Ex = -dVdx
    Ey = -dVdy
    return Ex, Ey




# ----------------------------
# Figure 1: discretized grid, boundary conditions, and mask
# ----------------------------
# mask
def save_mask(
        mask: np.ndarray,
        filename : str,
        cmap: str = "gray") -> None:
    x = np.linspace(0, Lx, mask.shape[1])
    y = np.linspace(0, Ly, mask.shape[0])
    X, Y = np.meshgrid(x,y)
    plt.figure(figsize=(5,4))
    ax = plt.gca()
    im = ax.imshow(
        mask,
        cmap=cmap,
        origin="lower",
        extent=[0, Lx, 0, Ly],
        aspect="equal",
        vmin=0,
        vmax=1
    )
    plt.text(0.025, 0.049, "0",
    ↪  ha="center",color=(0,1,1),fontsize=6)
    plt.text(0.025, 0, "0",
    ↪  ha="center",color=(0,1,1),fontsize=6)
    plt.text(0.049, 0.025, "0", rotation=90,
    ↪  va="center",color=(0,1,1),fontsize=6)
    plt.text(0, 0.025, "+V", rotation=90,
    ↪  va="center",color=(0,1,1),fontsize=6)
    ax.set_xticks(x,minor=True)
    ax.set_yticks(y,minor=True)

    ↪  ax.grid(which='minor',color=(0,1,1),linestyle='-',linewidth=0.1)
    cbar = plt.colorbar(im, ax=ax)
    cbar.set_label("mask")
    ax.set_xlabel("x (m)")
    ax.set_ylabel("y (m)")
    plt.tight_layout()
    plt.savefig(filename, dpi=1200, bbox_inches="tight")
    plt.close()




# ----------------------------
# Heatmap + vector field
# ----------------------------
def save_heatmap_with_field(
        V: np.ndarray,
        filename: str,
        cmap: str = "viridis",
        defect_mask: np.ndarray | None = None,
        annotate_defect: bool = False
) -> None:
    Ex, Ey = compute_electric_field(V, Lx, Ly)
    # coordinate grid in meters
    x = np.linspace(0, Lx, V.shape[1])
    y = np.linspace(0, Ly, V.shape[0])
    X, Y = np.meshgrid(x, y)
    plt.figure(figsize=(5, 4))
    ax = plt.gca()
    im = ax.imshow(
        V,
        cmap=cmap,
        origin="lower",
        extent=[0, Lx, 0, Ly],
        aspect="equal",
        vmin=0,
        vmax=1
    )
    cbar = plt.colorbar(im, ax=ax)
    cbar.set_label("electric potential (V)")

    # Subsample arrows so plot isn't overcrowded
    step = 4
    Xq = X[::step, ::step]
    Yq = Y[::step, ::step]
    Exq = Ex[::step, ::step]
    Eyq = Ey[::step, ::step]

    # If defect exists, do not draw arrows inside it
    if defect_mask is not None:
        maskq = defect_mask[::step, ::step]
        Exq = np.where(maskq, np.nan, Exq)
        Eyq = np.where(maskq, np.nan, Eyq)
    ax.quiver(
        Xq, Yq, Exq, Eyq,
        color="black",
        #scale=10,
        width=0.003,
        headwidth=3,
        headlength=4
    )

    # Draw defect box if present
    if defect_mask is not None:
        rows, cols = np.where(defect_mask)
        i_min, i_max = rows.min(), rows.max()
        j_min, j_max = cols.min(), cols.max()

        dx = Lx / (V.shape[1] - 1)
        dy = Ly / (V.shape[0] - 1)
        x0 = j_min * dx
        y0 = i_min * dy
        width = (j_max - j_min + 1) * dx
        height = (i_max - i_min + 1) * dy
        rect = Rectangle(
            (x0, y0), width, height,
            linewidth=1.5,
            edgecolor="black",
            facecolor="none",
            linestyle="--"
        )
        ax.add_patch(rect)

        if annotate_defect:
            ax.text(
                x0 + width / 2,
                y0 + height / 2,
                "defect",
                color="black",
                fontsize=8,
                ha="center",
                va="center"
            )

    ax.set_xlabel("x (m)")
    ax.set_ylabel("y (m)")
    plt.tight_layout()
    plt.savefig(filename, dpi=1200, bbox_inches="tight")
    plt.close()




# ----------------------------
# Difference map
# ----------------------------
def save_difference_map(
        deltaV: np.ndarray,
        filename: str,
        defect_mask: np.ndarray | None = None,
        annotate_defect: bool = False) -> None:
    plt.figure(figsize=(5, 4))
    ax = plt.gca()
    im = ax.imshow(
        deltaV,
        cmap="bwr",
        origin="lower",
        extent=[0, Lx, 0, Ly],
        aspect="equal"
    )

        # Draw defect box if present
    if defect_mask is not None:
        rows, cols = np.where(defect_mask)
        i_min, i_max = rows.min(), rows.max()
        j_min, j_max = cols.min(), cols.max()

        dx = Lx / (deltaV.shape[1] - 1)
        dy = Ly / (deltaV.shape[0] - 1)
        x0 = j_min * dx
        y0 = i_min * dy
```

```
        width = (j_max - j_min + 1) * dx
        height = (i_max - i_min + 1) * dy
        rect = Rectangle(
            (x0, y0), width, height,
            linewidth=1.5,
            edgecolor="black",
            facecolor="none",
            linestyle="--"
        )
        ax.add_patch(rect)

        if annotate_defect:
            ax.text(
                x0 + width / 2,
                y0 + height / 2,
                "defect",
                color="black",
                fontsize=8,
                ha="center",
                va="center"
            )

    im.set_clim(-0.3,0.3)
    cbar = plt.colorbar(im, ax=ax)
    cbar.set_label("change in electric potential (V)")
    ax.set_xlabel("x (m)")
    ax.set_ylabel("y (m)")
    plt.tight_layout()
    plt.savefig(filename, dpi=1200, bbox_inches="tight")
    plt.close()




# ----------------------------
# Main
# ----------------------------
def main():
    # Figure 1
    #save_figure1_grid_setup("Figure1_GridSetup.png", N)

    # No defect
    V_base = solve_laplace(N, ITERATIONS)
    save_heatmap_with_field(
        V_base,
        "Figure2_BaselinePotential.png",
        cmap="viridis"
    )

    # With defect
    defect_mask = make_center_defect_mask(N,
    ↪  DEFECT_HALF_WIDTH)
    V_defect = solve_laplace(N, ITERATIONS,
    ↪  defect_mask=defect_mask)
    save_heatmap_with_field(
        V_defect,
        "Figure3_DefectPotential.png",
        cmap="viridis",
        defect_mask=defect_mask,
        annotate_defect=True
    )

    # Difference map
    deltaV = V_defect - V_base
    save_difference_map(
        deltaV,
        "Figure4_PotentialDifference.png",
        defect_mask=defect_mask,
        annotate_defect=True
    )

    # defect mask
    save_mask(defect_mask,"Figure1_mask.png")

    print("Done. Generated Figure1{Figure4 PNG files.")




if __name__ == "__main__":
    main()
```

ACKNOWLEDGMENT

We thank the Manalapan High School Science & Engineering program for their support. We thank several anonymous reviewers for helpful feedback. DA developed the computational model, implemented the numerical simulation, and performed the analysis of the results. NK helped with developing the numerical approach, helped test and verify the simulation results, and contributed to editing and revising the written sections. AT supported the project by reviewing the model setup, helping with interpretation of the results, and providing feedback during the writing process.

REFERENCES

[1] R. D. Knight, *Physics for Scientists and Engineers: a strategic approach*, 4th ed. Pearson, 2017.

[2] J. D. Jackson, *Classical Electrodynamics*, 3rd ed. John Wiley & Sons, 1999.

[3] S. Ayyagari, V. Collemi, K. Shah, and K. Tomazic, "Computational mapping analysis of equipotential and electric field lines in gel electrophoresis rig," *Journal of Science & Engineering*, vol. 1, pp. 37–40, 2024.

[4] S. Baru, V. Choudhary, P. Thaker, N. Martin, and D. Ahmad, "Demonstrating a method to create a low-cost electrophoresis rig solution," *Journal of Science & Engineering*, vol. 1, pp. 42–44, 2024.

[5] R. Cohen, S. Musuku, J. Hammer, E. Handique, N. Patel, D. Gandhi, and N. Gershteyn, "Visualizing electric potential: mapping equipotential lines in a conductive water tray," *Journal of Science & Engineering*, vol. 1, pp. 45–47, 2024.

[6] R. Edwards, C. Karabin, C. Li, K. Patel, and J. Schatz, "Electric field mapping for cost-effective gel electrophoresis applications," *Journal of Science & Engineering*, vol. 1, pp. 49–52, 2024.

[7] A. Kumar, S. Perkins, N. Muthukumar, H. Villaseñor, and A. Khanna, "Mapping electric potential and electric field distribution in saltwater and investigating the effect of distance from source," *Journal of Science & Engineering*, vol. 1, pp. 53–55, 2024.

[8] J. R. Davis, *Non-Destructive Evaluation of Materials*. Materials Park, OH: ASM International, 2004.

[9] P. Cawley, "Non-destructive testing—current capabilities and future directions," *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*, vol. 215, pp. 213–223, 2001.

[10] M. H. Loke, J. E. Chambers, D. F. Rucker, O. Kuras, and P. B. Wilkinson, "Recent developments in the direct-current geoelectrical imaging method," *Journal of Applied Geophysics*, vol. 95, pp. 135–156, 2013.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: the art of scientific computing*, 2nd ed. Cambridge University Press, 1992.

[12] R. A. Pelcovits and J. Farkas, *Barron's AP Physics C Premium*. Fort Lauderdale, FL: Kaplan North America, 2024.

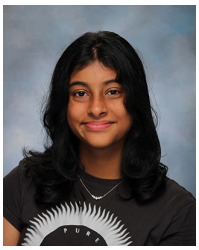
[13] P. A. Tipler and G. Mosca, *Physics for Scientists and Engineers*, 5th ed. New York: W H Freeman and Company, 2004.

[14] W. Moebs, S. J. Ling, and J. Sanny, *University Physics*. Houston, TX: OpenStax, 2016, vol. 1.

[15] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[16] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, pp. 90–95, 2007.

[17] A. Hauptmann, M. Ikehata, H. Itou, and S. Siltanen, "Revealing cracks inside conductive bodies by electric surface measurements," *Inverse Problems*, vol. 35, p. 025004, 2018.



Dia Avalur is a senior in the Science and Engineering Magnet Program at Manalapan High School. Her main area of research is the intersection between engineering and medicine, including the development of alginate patches for drug delivery, early detection of neurological conditions from machine learning, and studies of segmentation in medical imaging.

Nitika Kishore is a senior in the Science and Engineering Magnet Program at Manalapan High School. Her main area of research is the wonderful world of lasers. When she is not studying AP Physics C E&M she enjoys developing low-cost intravenous monitoring devices and breathtaking laser light shows that turn it up to 11.

Anika Tokala is a senior in the Science and Engineering Magnet Program at Manalapan High School. Her main area of research is accessibility technology. When she is not studying AP Physics C E&M she enjoys literature and developing screen readers to assist blind and visually impaired students with math.