Numerical simulation of electrical non-destructive testing of metals for flaw detection via the finite difference method

Jophy Lin, Sagarika Yagnyeshwaran, Rishith Chandra Kilaru, Srilekha Dantu, and Vijita Ayyangar

***Abstract*—The detection of internal flaws in a 3D-printed metal sample was simulated using a non-destructive testing (NDT) approach on a 2D grid. A finite difference method on a square grid of size 60 × 60 nodes was used to compute the electric potential across the material, and Successive Over-Relaxation (SOR) was applied to efficiently solve the linear equations. By iterating over the finite difference grid and updating the electric potential at each point until a set tolerance was reached, changes in voltage across the material became apparent. Regions with significant positive and negative voltage deviations indicated the presence and location of a crack near the center of the domain of the conductivity map. This method matters because it demonstrates how computational techniques can be used to detect defects in metal components without physically damaging them, which is useful in manufacturing and materials inspection.**

***Index Terms*—computational, 3D printing, Gauss-Seidel, finite difference, Python, conductivity, Laplace's equation, Poisson's equation, finite difference method, successive over-relaxation**

I. Introduction

3D PRINTING conductive materials, including metals and conductive polymers, allows for complex designs and rapid production, but can introduce internal defects. These include cracks and incomplete bonding between layers. These flaws may significantly degrade mechanical strength or electrical performance while remaining invisible to external inspection [1].

Common non-destructive testing (NDT) techniques, such as x-ray imaging or ultrasonic inspection, often require specialized equipment or a multitude of expenditures that may not be available in testing environments [2], [3]. As a result, simpler and more accessible methods for defect detection are in demand.

Electrical probing provides a potential alternative, as internal changes in conductivity could alter current paths and voltage distributions [4]. Because voltage can be measured at the surface of a metal, internal defects may be detected indirectly through imperfections in the observed voltage field.

Previous researchers examined voltage mapping to experimentally characterize the equipotentials and electric fields within a low-cost gel electrophoresis device [5]–[9]. However, this previous work did not attempt numerical simulation.

Author for correspondence: 126jlin@frhsd.com

Authors are with the Science & Engineering Magnet Program, Manalapan High School, 20 Church Lane, Englishtown, NJ 07726, USA

10.64804/z4c41213

In this work, the feasibility of electrical non-destructive testing is investigated using a finite-difference computational model of steady-state conduction [10]. By comparing voltage distributions in a uniform conductive sheet to those obtained in the presence of an internal low-conductivity defect, localized voltage perturbations are identified. These deviations from the baseline voltage field indicate the presence and approximate location of internal flaws.

The following is the hypothesis being tested in this experiment: if there is no flaw, there should be no measurable change in voltage:

$$H_0 : \Delta V = V_{flaw} - V_{clean} = 0 \tag{1}$$

In the case that the flaw exists, the voltage distribution changes, showing a measurable disruption in the surface of the material. This can be modeled by the following equation:

$$H_1 : \Delta V = V_{flaw} - V_{clean} \neq 0 \tag{2}$$

II. Methods and materials

Under steady-state conditions and in the absence of internal current, electrical conduction in a material with varying conductivity $\sigma(x, y)$ is governed by

$$\vec{\nabla} \cdot (\sigma \vec{\nabla} V) = 0 \tag{3}$$

where $V(x, y)$ is the electric potential. The material was assumed to be isotropic, and time-dependent factors were neglected.

The conductive domain was modeled as a two-dimensional square region on a regular uniform $60 \times 60\,\mathrm{node}$ finite-difference grid. This was to ensure that each node was relatively small compared to the simulated piece of metal, similar to how a real piece of metal has near-infinitely small particles. All voltages were evaluated at the grid nodes, with uniform spacing between each node for simplicity.

Fixed-potential (Dirichlet) boundary conditions were applied on the left and right boundaries of the domain, with voltages of $1.0\,\mathrm{V}$ and $0.0\,\mathrm{V}$, respectively, representing the source and sink electrodes. The top and bottom boundaries were treated as electrically insulating by enforcing zero normal voltage gradients. Together, these boundaries establish a controlled voltage field across the metal, allowing the low-conductivity regions to locally distort the voltage field.

To simulate defects in the metal, we modeled localized regions of lower electrical conductivity. Because a flawless piece

of metal that has no defects would be uniformly conductive, the lower conductivity represents the absence of metal, which is the gaps and cracks in a real piece of metal. The conductivity of the flawless material was set constant to 1.0, while the conductivity of the flawed regions was set constant to 1×10^{-9}, both in normalized units.

The governing equation was discretized using a finite-difference formulation that enforces current conservation at each grid node. Conductivity at cell interfaces was computed using the harmonic mean to accurately represent sharp conductivity contrasts. The harmonic mean was used because it models how current flows across materials with varying conductivities. Current is slowed down by the poorly conductive region, which is the crack, and so by taking the harmonic mean, the weight is shifted towards the smaller value. This method allows us to ensure that the current is conserved and prevents the overestimation of conductivity at the boundaries by clearly showing the defect.

Since we needed to find the voltage at all 3600 points, we would either use the Jacobi method or the Gauss-Seidel [10] method to solve the resulting system of linear equations. The Gauss-Seidel method updates the nodes individually based on the neighboring nodes, whereas the Jacobi method updates all nodes simultaneously using values from the previous iteration, making the Jacobi method computationally expensive. This is why the Gauss-Seidel method was implemented, where each calculation uses updated information rather than waiting for an entire iteration through the grid; the formula below was used to update the voltage of each node:

$$V_{i,j}^{(\mathrm{GS})} = \frac{\sigma_{i+\frac{1}{2},j}\,V_{i+1,j} + \sigma_{i-\frac{1}{2},j}\,V_{i-1,j} + \sigma_{i,j+\frac{1}{2}}\,V_{i,j+1} + \sigma_{i,j-\frac{1}{2}}}{\sigma_{i+\frac{1}{2},j} + \sigma_{i-\frac{1}{2},j} + \sigma_{i,j+\frac{1}{2}} + \sigma_{i,j-\frac{1}{2}}} \tag{4}$$

Successive-Over-Relaxation [10] (SOR) is the process of iteratively solving systems of linear equations by accelerating the convergence of the Gauss-Seidel Method. In order to further optimize and increase the speed of the program, we used the Successive-Over-Relaxation (SOR) technique until the maximum change in voltage between iterations fell below 1×10^{-6} V using the below formula:

$$V_{i,j}^{(n+1)} = (1-\omega)V_{i,j}^{(n)} + \omega V_{i,j}^{(\mathrm{GS})} \tag{5}$$

The electric field was computed from the converged voltage distribution using:

$$\vec{E} = -\vec{\nabla}V \tag{6}$$

To isolate the perturbations induced by the defects, the voltage difference field was calculated as such:

$$\Delta V = V_{flaw} - V_{clean} \tag{7}$$

The voltage difference field was then analyzed as the primary non-destructive testing signal. Numerical computations were performed in Python using the `numpy` and `matplotlib` libraries [11], [12]. A Jupyter notebook [13] with finite difference code is available at https://github.com/usernnamee/NDT-Testing-SOR and https://jupyter.snerds.org. Data and code are also available at https://github.com/devangel77b/427syagnyeshwaran-lab3.

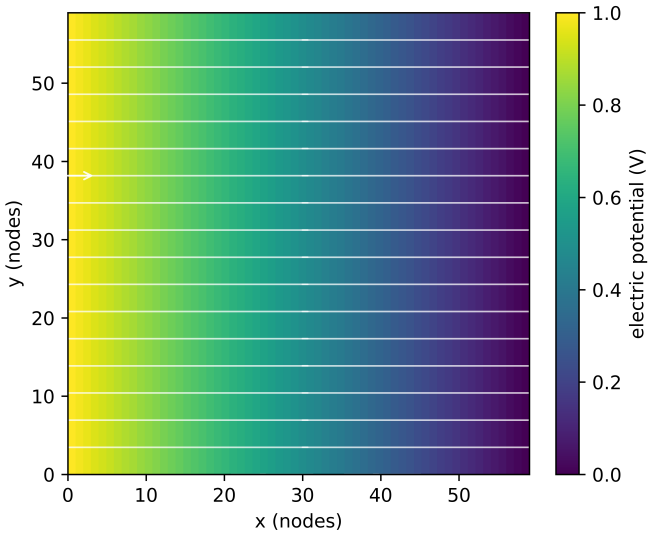


Fig. 1. Electric potential distribution for a uniform conductive sheet with no internal flaw.

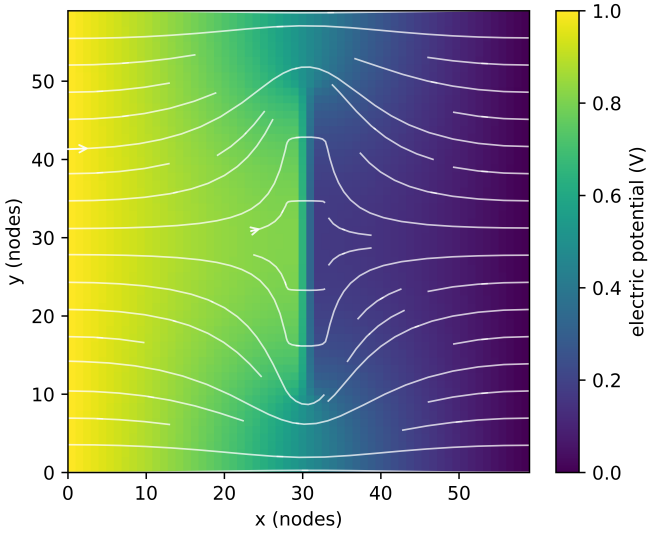


Fig. 2. Electric potential distribution for a conductive sheet containing a low-conductivity flaw.

III. Results

The electric potential is shown in Fig. 1 for a uniform 3D-printed metal with no internal flaw present.

The potential distribution after adding a flaw is shown in Fig. 2, with arrows representing the potential field lines that outline the shape of the flaw and defect. In this simulation, the flaw is a thin, vertical crack.

Fig. 3 shows the voltage difference, evidently highlighting the location at which there is an internal flaw. The blue voltage difference represents a location at which the voltage drops and the current is rerouted, whereas the red voltage difference represents a location at which the voltage is higher than expected, and the current is being impeded upstream.

The shape of the flaw inserted into the surface of the metallic material is depicted in Fig. 4.

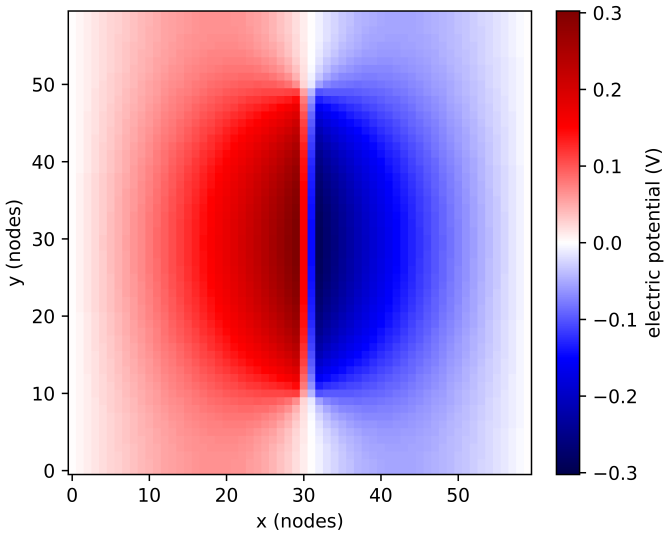


Fig. 3. Voltage difference (ΔV) between the defect-free and defective cases, serving as the electrical NDT signal.

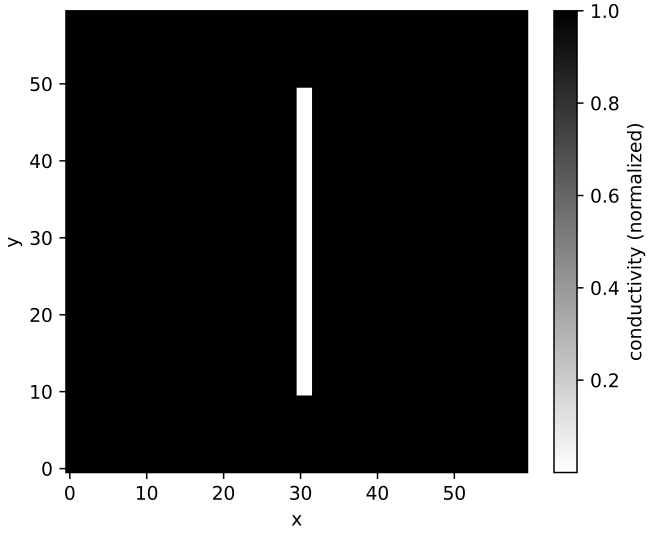


Fig. 4. Conductivity map of the simulated conductive sheet showing the location and geometry of the introduced flaw.

These results support the idea that electrical probing is a viable non-destructive testing method for conductive 3D-printed materials, as changes in voltage reflect hidden disruptions in conductivity without physically damaging the part to test for imperfections.

IV. Discussion

The behavior observed in this study can be explained by how electric current responds to changes in conductivity within a material.

In a conductive sheet with no defects, current distributes evenly and the voltage changes smoothly across the surface. When a flaw is present, the local conductivity is reduced, which increases resistance and forces current to reroute around the damaged region.

This rerouting causes uneven voltage changes that can be detected at the surface, even though the defect itself is internal. Because voltage depends directly on how current flows through the material, small disruptions in conductivity can produce noticeable differences in the measured potential.

Therefore, we reject the null hypothesis; flaws resulting in observable voltage differences than can be used as a diagnostic tool, subject to some conditions. This study is limited by the use of a simplified two-dimensional model and idealized material properties, which do not capture all aspects of real 3D-printed metal parts. Similarly, when using grids of different resolutions (30 by 30 and 120 by 120 in other tests), while keeping the size of the flaw relative to the grid constant, there was very little change in the observed pattern. The maximum voltage difference remained constant across different grid resolutions, which confirms grid convergence as it is independent of grid size. The spatial resolution and conductivity difference between the base material and the flaw will effect the minimum size of flaw that can be detected.

In practice, defects may have complex shapes, varying depths, and partial conductivity rather than being fully insulating. Future work could extend this approach to three-dimensional models, explore different defect sizes and conductivities, and experiment with various kinds of metals. Experimental validation using physical conductive materials or 3D-printed samples would further test the practicality of this method for real-world NDT applications.

In the future, we aim to make this process more precise by detecting the absence or presence of a flaw by testing the outer boundaries rather than interior. This is more accurate to real life since in a practical NDT situation, we only have access to the surface and not the interiors

Appendix

The following Python code was used to implement the finite-difference solver, apply boundary conditions, compute the electrical field, and generate Figs. 1 to 4. The code makes use of the `numpy` and `matplotlib` libraries [11], [12]. Relaxation or Gauss-Seidel is implemented around line 59, while successive over-relaxation is implemented around line 62 [10].

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  Nx, Ny = 60, 60
5  iters = 8000
6  tol = 1e-6
7
8  def harmonic_mean(a, b, eps=1e-12):
9      return (2.0 * a * b) / (a + b + eps)
10
11 def compute_electric_field(V, hx=1.0, hy=1.0):
12     dVdi, dVdj = np.gradient(V, hx, hy, edge_order=2)
13     Ex = -dVdj
14     Ey = -dVdi
15     return Ex, Ey
16
17 def solve_fd_conductance(with_flaw=False, omega=1.8):
18     sigma = np.ones((Nx, Ny), dtype=float)
19     if with_flaw:
20         sigma[10:50, 30:32] = 1e-9
21
22     V = np.zeros((Nx, Ny), dtype=float)
23     for j in range(Ny):
24         V[:, j] = 1.0 + (0.0 - 1.0) * (j / (Ny - 1))
25
```

```
    fixed = np.zeros((Nx, Ny), dtype=bool)
    fixed[:, 0] = True
    fixed[:, -1] = True

    def apply_bc(V):
        V[:, 0] = 1.0
        V[:, -1] = 0.0
        V[0, :] = V[1, :]
        V[-1, :] = V[-2, :]

    apply_bc(V)

    for _ in range(iters):
        max_delta = 0.0
        for i in range(1, Nx - 1):
            for j in range(1, Ny - 1):
                if fixed[i, j]:
                    continue

                ip, im = i + 1, i - 1
                jp, jm = j + 1, j - 1

                sxp = harmonic_mean(sigma[i, j], sigma[ip, j])
                sxm = harmonic_mean(sigma[i, j], sigma[im, j])
                syp = harmonic_mean(sigma[i, j], sigma[i, jp])
                sym = harmonic_mean(sigma[i, j], sigma[i, jm])

                A = sxp + sxm + syp + sym
                if A < 1e-20:
                    continue

                # The next line implements relaxation or
                # lazy Gauss-Seidel
                V_ideal = (sxp * V[ip, j] + sxm * V[im, j] +
                           syp * V[i, jp] + sym * V[i, jm]) / A

                # The next line implements successive over-relaxation
                V_old = V[i, j]
                V[i, j] = (1.0 - omega) * V_old + omega * V_ideal

                delta = abs(V[i, j] - V_old)
                if delta > max_delta:
                    max_delta = delta

        apply_bc(V)
        if max_delta < tol:
            break

    return V, sigma

def add_arrowheads(ax, stream, color="white", lw=1.0):
    segs = stream.lines.get_segments()
    if len(segs) == 0:
        return

    k = 12
    for idx in range(0, len(segs), k):
        seg = segs[idx]
        (x0, y0), (x1, y1) = seg[0], seg[1]
        ax.annotate(
            "",
            xy=(x1, y1),
            xytext=(x0, y0),
            arrowprops=dict(arrowstyle="->", color=color,
                            linewidth=lw, shrinkA=0,
                            shrinkB=0),
        )

# Run
V_clean, sigma_clean = solve_fd_conductance(with_flaw=False)
V_flaw, sigma_flaw = solve_fd_conductance(with_flaw=True)

V_diff = V_flaw - V_clean

Ex_clean, Ey_clean = compute_electric_field(V_clean)
Ex_flaw, Ey_flaw = compute_electric_field(V_flaw)

# PLOTTING (four figures)

x = np.arange(Nx)
y = np.arange(Ny)
X, Y = np.meshgrid(x, y)

# No flaw (V + E streamlines)
fig,ax = plt.subplots(figsize=(5,4), constrained_layout=True, dpi=1200)
im0 = ax.imshow(V_clean, origin="lower", cmap="viridis", vmin=0, vmax=1, extent=[0,Nx-1,0,Ny-1])
s0 = ax.streamplot(
    X, Y, Ex_clean, Ey_clean,
    density=0.6,
    linewidth=0.9,
    color=(1, 1, 1, 0.75),
    arrowstyle='-'
)
add_arrowheads(ax, s0, color=(1, 1, 1, 0.9), lw=1.0)
ax.set_xlabel("x (nodes)")
ax.set_ylabel("y (nodes)")
cb0=plt.colorbar(im0, ax=ax)
cb0.set_label("electric potential (V)")
fig.savefig("potential-noflaw.png",dpi=1200)
plt.close(fig)

# With flaw (V + E streamlines)
fig,ax = plt.subplots(figsize=(5,4), constrained_layout=True, dpi=1200)
im1 = ax.imshow(V_flaw, origin="lower", cmap="viridis", vmin=0, vmax=1,extent=[0,Nx-1,0,Ny-1])
s1 = ax.streamplot(
    X, Y, Ex_flaw, Ey_flaw,
    density=0.6,
    linewidth=0.9,
    color=(1, 1, 1, 0.75),
    arrowstyle='-'
)

add_arrowheads(ax, s1, color=(1, 1, 1, 0.9), lw=1.0)
ax.set_xlabel("x (nodes)")
ax.set_ylabel("y (nodes)")
cb1=plt.colorbar(im1, ax=ax)
cb1.set_label("electric potential (V)")
fig.savefig("potential-flaw.png",dpi=1200)
plt.close(fig)

# Difference signal
fig,ax = plt.subplots(figsize=(5,4), constrained_layout=True, dpi=1200)
dvmax = np.max(np.abs(V_diff))
im2 = ax.imshow(V_diff, origin="lower", cmap="seismic", vmin=-dvmax, vmax=dvmax)
ax.set_xlabel("x (nodes)")
ax.set_ylabel("y (nodes)")
cb2=plt.colorbar(im2, ax=ax)
cb2.set_label("electric potential (V)")
fig.savefig("potential-difference.png",dpi=1200)
plt.close(fig)

# Conductivity map
fig,ax = plt.subplots(figsize=(5,4), constrained_layout=True, dpi=1200)
im3 = ax.imshow(sigma_flaw, origin="lower", cmap="gray_r")
ax.set_xlabel("x")
ax.set_ylabel("y")
cb3=plt.colorbar(im3, ax=ax)
cb3.set_label("conductivity (normalized)")
fig.savefig("conductivity.png",dpi=1200)
plt.close(fig)
```

ACKNOWLEDGEMENT

The team thanks all group members for their collaborative contributions to this work, including code development, simulation design, and manuscript writing. This work was made possible through close teamwork and collaboration among all group members throughout the research and writing process.

Additionally, the team acknowledges the Science and Engineering Magnet Program for providing access to the Jupyter Notebook computing environment used to develop, run, and test the simulations. Lastly, the team would like to thank all peers who have provided constructive feedback and review, helping improve the quality of this work.

REFERENCES

[1] N. Dalal, Y. Gu, D. R. Hines, A. Dasgupta, and S. Das, "Cracks in the 3d-printed conductive traces of silver nanoparticle ink," *Journal of Micromechanics and Microengineering*, vol. 29, p. 097001, 2019.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

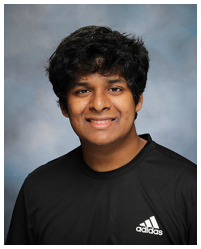
[11] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser,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

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

[13] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter Notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87–90.

Jophy Lin is a senior in the Science and Engineering Magnet Program at Manalapan High School. She was also a member of Hackathon Leadership Team. She is a member of the Research Club and enjoys labubus, hanging out in Harvard Square, projective geometry, RNA folding, and streaming foreign language dramas.

Sagarika Yagnyeshwaran is a senior in the Science and Engineering Magnet Program at Manalapan High School. She was also an intern at Commvault in Tinton Falls, NJ. She enjoys heads up displays, Trivia Club, Hackathon Leadership Team, concert chorus, and streaming foreign language dramas.



Rishith Chandra Kilaru is a senior in the Science and Engineering Magnet Program at Manalapan High School. He was also an intern at WIT Sports in New York, and a member of Hackathon Leadership Team. When he is not building augmented reality video games, he enjoys karaoke, ramen, and writing, and is famous for BONELAB modding.

Srilekha Dantu is a senior in the Science and Engineering Magnet Program at Manalapan High School. She enjoys working on AP Physics C E&M and on senior projects such as a low-cost intravenous weight monitoring device and a Spiderman grapple gun.

Vijita Ayyangar is a senior in the Science and Engineering Magnet Program at Manalapan High School. She is an intern at I-House Architecture in Belmar, NJ and a member of the Drama Club. In her free time, she enjoys working on AP Physics C E&M.