

EE2703: Final Exam

Akilesh Kannan (EE18B122)

July 31, 2020

1 Problem Description

The Laplace equation (1) is a special case of Poisson's equation applicable in a region where the charge concentration is zero inside. It provides for a simple condition for the electric potential function ϕ , thus making it mathematically easy to compute the potential distribution in the region. This method is widely used to perform numerical simulations of regions with given boundary conditions.

$$\nabla^2 \phi = 0 \quad (1)$$

In 2 dimensions, Equation (1) reduces to the PDE,

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (2)$$

For numerical simulation, it can be proved that [1] the potential at any point (i, j) can be written as

$$\phi_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}}{4} \quad (3)$$

Equation (3) is valid for those points which aren't at the interface between 2 dielectrics. For those points, the potentials can be written as

$$\phi_{interface} = \frac{\epsilon_{ra}\phi_a + \epsilon_{rb}\phi_b}{\epsilon_{ra} + \epsilon_{rb}} \quad (4)$$

where the subscript a denotes above the boundary and b , for below the boundary.

The main objective of the problem given is to analyse the dependence of a metal tank's capacitance on the height of fluid (a dielectric) in the tank, and how it affects the resonant frequency of an external RLC circuit, using the tank as it's capacitor. The tank's side and bottom walls are kept grounded, while the top is connected to the RLC circuit, and is assumed to be at 1V.

The angular resonant frequency of the RLC circuit is given by

$$\omega_o = \frac{1}{\sqrt{LC}} \quad (5)$$

It is required to solve Laplace's equation inside the tank (shown in 1) to find the potential distribution inside it, and from it, the capacitance seen by the circuit and the resonant frequency of the circuit.

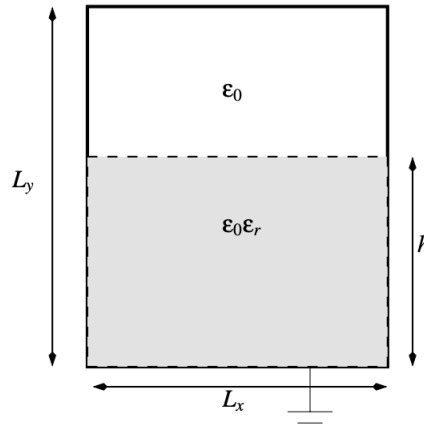


Figure 1: The tank

1.1 Notations and Conventions followed in program

1. After meshing the potential in the region into a $M \times N$ mesh of nodes, the boundary electric potential is assigned such that the top row of nodes is completely assigned to be 1V on the top plate, while the rest of the boundary is assigned to be having 0V (GND).
2. The origin is taken to be the bottom-left corner of the tank.

2 Formulating a method to estimate the height of the tank

Given the resonant angular frequency ω_o and the value of L , the capacitance of the tank is

$$C = \frac{1}{L\omega_o^2} \quad (6)$$

Now that we know the capacitance of the tank, we need a relation between the height of the fluid in the tank and its capacitance.

If the charge on the top plate (at 1V) is given by Q_{top} , then the capacitance is simply

$$C = \frac{Q}{V} = \frac{Q_{top}}{1V} = Q_{top} \quad (7)$$

Therefore, we need to find the total charge accumulated on the top surface of the tank.

Using Laplace's Equation (3) iteratively, we can find the potential at any point inside the tank. Using these values, we can find the Electric field (\vec{E}) by taking the negative gradient of the potential.

$$\vec{E} = \nabla\phi \quad (8)$$

We find the electric field at the center of a square enclosed by 4 mesh nodes $[(m, n), (m + 1, n), (m, n + 1) \text{ and } (m + 1, n + 1)]$ as shown below. Here, δ is the separation between 2 adjacent nodes.

$$E_{x,(m+0.5,n+0.5)} = -\frac{1}{2} \left(\frac{\phi_{m,n+1} - \phi_{m,n}}{\delta} + \frac{\phi_{m+1,n+1} - \phi_{m+1,n}}{\delta} \right) \quad (9)$$

$$E_{y,(m+0.5,n+0.5)} = -\frac{1}{2} \left(\frac{\phi_{m+1,n} - \phi_{m,n}}{\delta} + \frac{\phi_{m+1,n+1} - \phi_{m,n+1}}{\delta} \right) \quad (10)$$

At the boundaries (i.e., the walls of the tank), the \vec{E} is normal to it, as the walls are ideal conductors, and they are related to the surface charge density as,

$$\vec{E} = \frac{\sigma}{\epsilon_o \epsilon_r} \hat{n} \quad (11)$$

where, \hat{n} is the unit vector in the normal direction to the surface.

Now, we have a simple relation between the charge on the surface and the electric field *normal* to it. Summing the charges at each point on the surface should give us the total charge Q on the surface. The calculation of capacitance of the tank can now be easily done with (7) and Q_{top} , which is obtained from (8)-(11).

Doing the above procedure for different heights of the fluid will give us a relationship between h , the height of the fluid, and C , the capacitance offered by the tank. We can then fit a curve to the $h - C$ values obtained to calculate h from C .

3 Solving Laplace's Equation in the Tank

As mentioned before, to solve Laplace's equation (1) numerically, we can iteratively calculate the potential at each point as the average of its neighbours. We stop the iterative calculation when we have reached a specified accuracy, in the error between successive iterations. The following code implements the iterative calculator.

```

1 def solve (M, N, step, k, accuracy, No)
2     '''
3     Function to solve Laplace's Equation
4     in the tank.
5
6     Assumes that top of tank is at 1V.
7
8     Input
9     -----
10    M: int
11        number of nodes along X-axis, including
12        boundary nodes
13    N: int
14        number of nodes along Y-axis, including
15        boundary nodes
16    step: float
17        distance between nodes (assumed same for
18        X- and Y- axes)
19    k: int
20        index corresponding to height h
21    accuracy: float
22        desired accuracy
23    No: int
24        maximum number of iterations
25
26    Output
27    -----
28    phi: 2d numpy array (MxN)
29        array of solved potentials
30    N: int
31        number of iterations carried out
32    err: 1d numpy array
33        error vector
34
35    '''
36    global Er
37
38    # initialise potentials to 0 everywhere, except at top plate
39    # potential at top = 1V
40    phi = np.zeros((N, M), dtype=float)
41    phi[-1, :] = 1.0
42
43    iteration=[]      # iteration number
44    error=[]          # error vector
45
46    # iteratively calculate potentials
47    for i in range(No):
48        # create copy of potentials
49        oldPhi = phi.copy()
50
51        # updating the potentials
52        phi[1:-1, 1:-1] = 0.25*(phi[1:-1, 0:-2]+phi[1:-1, 2:]+phi[0:-2, 1:-1]+phi[2:, 1:-1])
53        phi[k, 1:-1] = (Er*oldPhi[k-1, 1:-1] + oldPhi[k+1, 1:-1])*1.0/(1+Er)
54
55        # Applying Boundary Conditions
56        phi[0, :] = 0.0          # bottom edge
57        phi[:, -1] = 0.0         # right edge
58        phi[:, 0] = 0.0          # left edge

```

```

59     phi[-1, :] = 1.0           # top edge
60
61     # calculating error
62     currError = np.abs(phi-oldPhi).max()
63     error.append(currError)
64     iteration.append(i)
65
66     # stop if accuracy reached
67     if currError <= accuracy:
68         break
69
70     return phi, iteration[-1], error

```

A contour plot of the potential distribution (for $h = 0.5L_y$) is shown below.

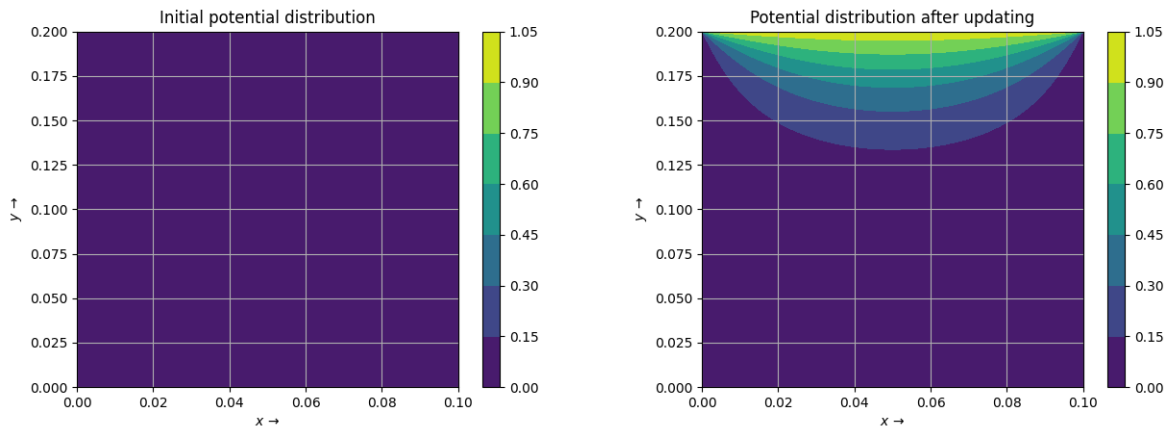


Figure 2: Potential Distribution - *left*: initialisation, *right*: iteratively calculated

3.1 Parallelising the computation

Computing potential values for a 200x100 grid iteratively is an awfully slow process due to heavy computational complexity and the general slowness of the Python language due to its interpreted nature. This can be parallelised to make the updation faster. Numpy arrays are internally implemented in C, with a wrapper function to use it in Python.

Numpy arrays can also be "*vectorized*", i.e., optimized, pre-compiled code written in a low-level language (e.g. C) to perform mathematical operations over a sequence of data, due to the fact that they are homogeneous data types.

Thus, executing the operations in C helps optimize the code, as each row of the array is operated on in parallel.

One thing to keep in mind is that the row corresponding to the interface has to be done separately. For that, we first create a copy of the current potential array. Then update the potentials of each row in the array by averaging the potentials of its neighbours, including the interface row, update the interface row using Equation (4), with the values from the copied array, and then apply boundary conditions.

This has been done in lines 49 - 59 in the above code.

3.2 Error Estimation and Fitting

The accuracy for the solution is passed as the argument `accuracy` to the function `solve()`, and the maximum number of iterations to be performed, `No` is also passed to it.

At each iteration, the maximum error between the updated values and the values got in the previous iteration are compared, and if the difference is less than `accuracy`, then the solver stops iterating anymore. If not, it keeps continuing till it reaches `No`.

This method of iteratively solving Laplace's equation leads to an exponential relation between the errors and iterations (after a certain number of iterations) [1]. We can thus fit the error-iteration curve to an exponential of the form $y = Ae^{Bx}$.

Figure (3) shows the actual error and the error fitted using least-squares method.

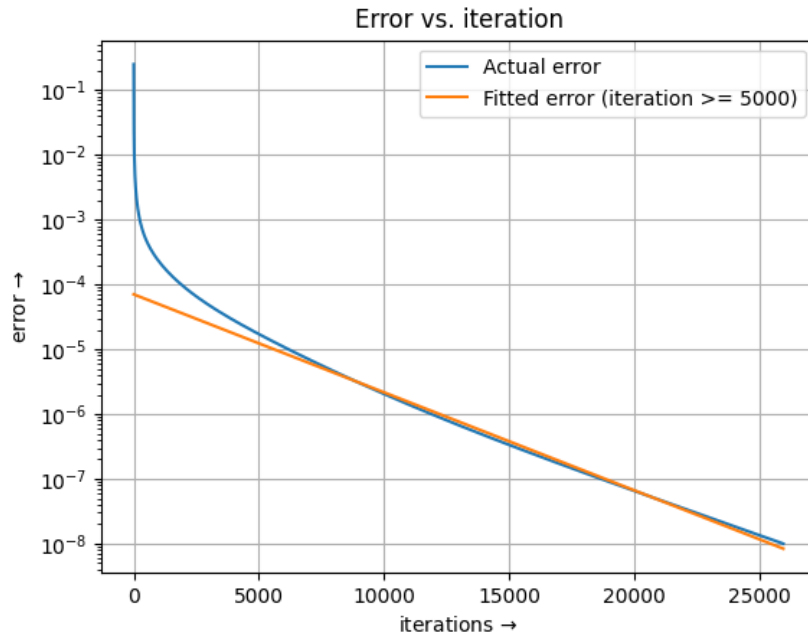


Figure 3: Exponential Fit of error with iteration

4 Calculation of \vec{E} from potential distribution

As described in Equations (8) - (11), we can find the electric field at the center of the mesh cells by averaging the gradients of the potential on it's sides. The following code snippet accomplishes the same

```

1 def findEField(phi, step, M, N):
2     '''
3     Calculates the x- and y- components of E-field at
4     each center point.
5
6     Input
7     -----
8     phi: 2d numpy array
9         potential array
10    step: float
11        distance between 2 points on the grid
12    M: int
13        nodes along x-axis
14    N: int
15        nodes along y-axis
16
17    Output
18    -----
19    Ex: 2d numpy array
20        X-components of E field
21    Ey: 2d numpy array
22        Y-components of E-field
23
24    '''
25    # Ex calculation
26    # * * * row i
27    # - - - --> center of mesh cells

```

```

28     # * * * * row i+1
29     #
30     negativeGradientX = (phi[:, :-1] - phi[:, 1:])*(1.0/step)
31     Ex = (negativeGradientX[:-1, :] + negativeGradientX[1:, :])*0.5
32
33     # Ey calculation
34     # * * *
35     #     -     --> center of mesh cells
36     # *     *
37     #     -     --> center of mesh cells
38     # *     *
39     # col i   col i+1
40     #
41     negativeGradientY = (phi[:, :-1] - phi[:, 1:])*(1.0/step)
42     Ey = (negativeGradientY[:, :-1] + negativeGradientY[:, 1:])*0.5
43
44     return Ex, Ey

```

Disclaimer: The plots presented in this section have been generated for a h/L_y ratio of 0.5, as asked in the Question Paper.

4.1 The normal \vec{D} continuity

The boundary conditions at the interface between air and fluid require the normal component of \vec{D} to be continuous across it:

$$D_{na} = D_{nb} \quad (12)$$

$$\epsilon_a E_{na} = \epsilon_b E_{nb} \quad (13)$$

Figure 4 shows the plot of \vec{D}_n above and below the interface. We can clearly see that the two are indeed equal.

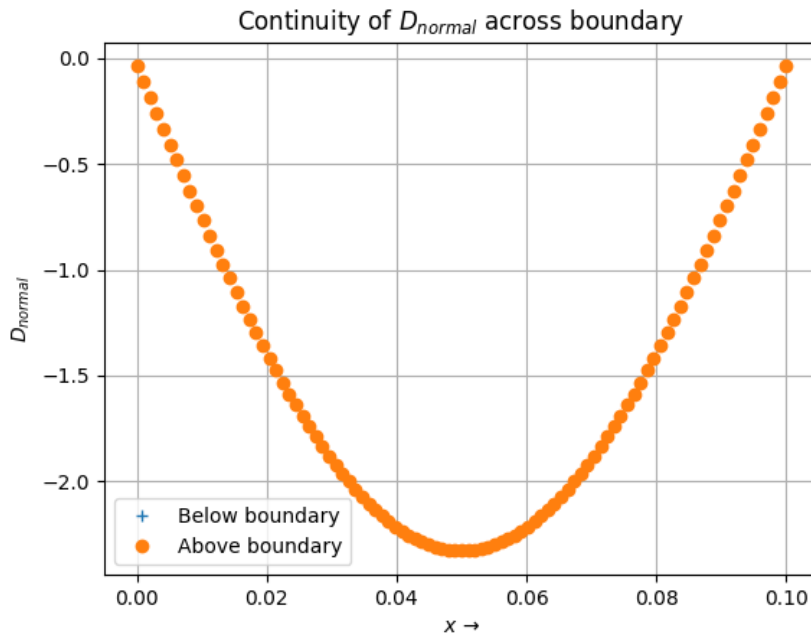


Figure 4: Continuity of \vec{D}_n across interface

4.2 The tangential \vec{E} continuity

The boundary conditions at the interface between air and fluid require the tangential component of \vec{E} to be continuous across it:

$$E_{ta} = E_{tb} \quad (14)$$

Figure 5 shows the plot of \vec{E}_t above and below the interface. We can clearly see that the two are indeed equal.

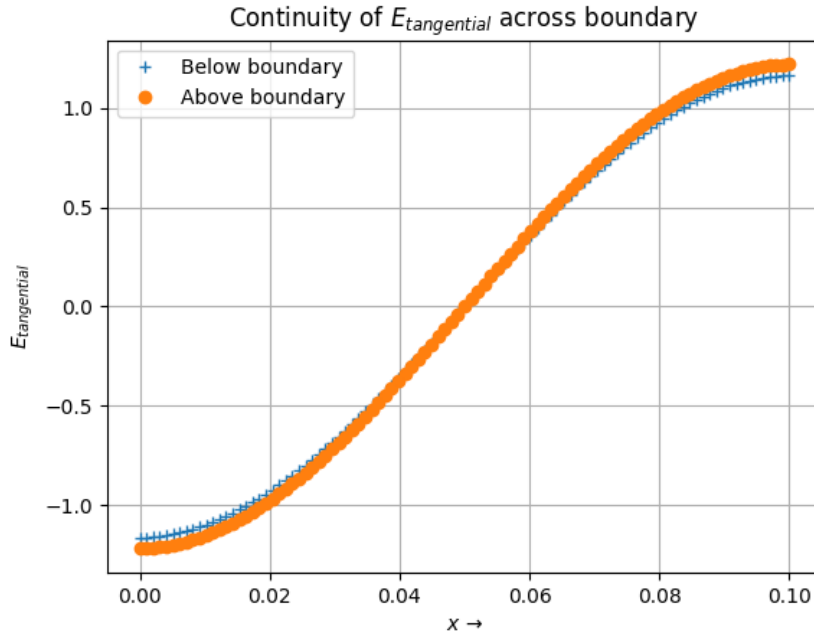


Figure 5: Continuity of \vec{E}_t across interface

4.3 Direction of \vec{E} at interface

Let us take θ_a and θ_b as the angles made by \vec{E} with the y-axis (normal to interface), above and below the interface respectively.

From boundary conditions imposed by Maxwell's Equations, we get,

$$E_a^{\parallel} = E_b^{\parallel} \quad (15)$$

$$\epsilon_a E_a^{\perp} = \epsilon_b E_b^{\perp} \quad (16)$$

or, in another form,

$$\frac{E_a^{\parallel}}{\epsilon_a E_a^{\perp}} = \frac{E_b^{\parallel}}{\epsilon_b E_b^{\perp}} \quad (17)$$

$$\implies \frac{\tan \theta_a}{\epsilon_a} = \frac{\tan \theta_b}{\epsilon_b} \quad (18)$$

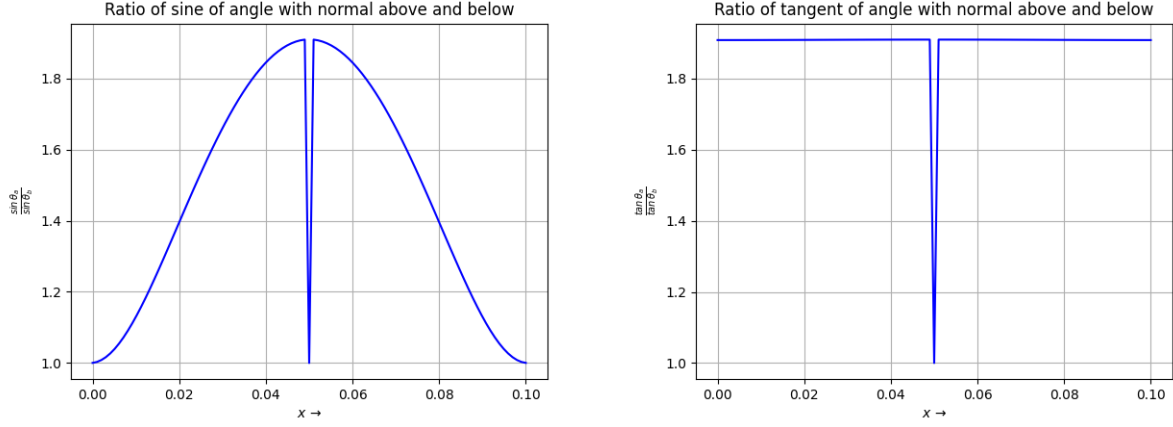
$$\implies \frac{\tan \theta_a}{\tan \theta_b} = \frac{\epsilon_a}{\epsilon_b} \quad (19)$$

i.e., the ratio of the tangents of the angles is a constant.

Thus, Snell's Law, which states that "the ratio of the sines of the angles of incidence and refraction of a wave are constant when it passes between two given media"[2], will not be valid.

This is expected because, Snell's law holds good only for *Electromagnetic Waves*, as given in the above statement, and not for *Electrostatic* distributions.

Given below are plots that confirm these inferences.

Figure 6: Charge variation with h - left: Q_{top} , right: Q_{fluid}

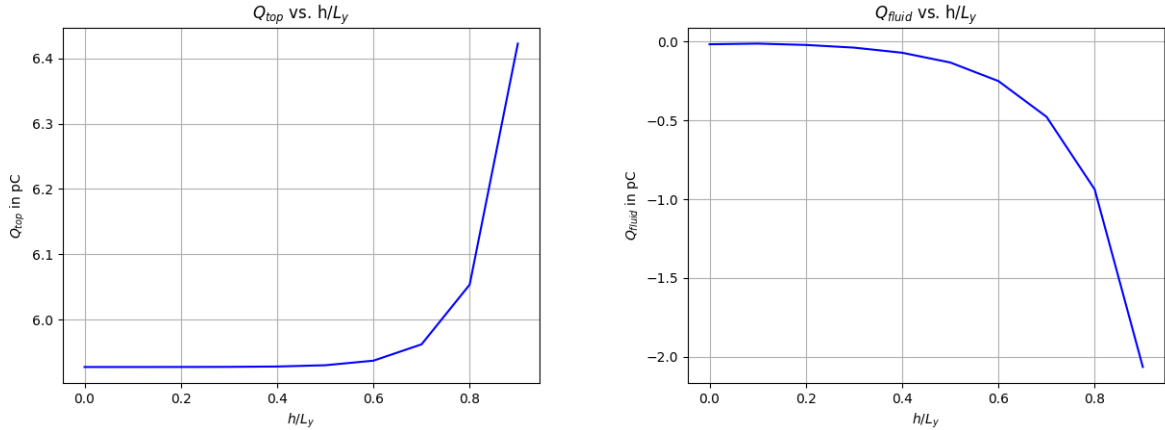
5 Dependence of Q_{top} on fluid height h

With the above calculated \vec{E} , we can find the total charge on the top plate Q_{top} for different values of h , going from $h = 0.1L_y$ to $h = 0.9L_y$, as described by Equation (11).

Similarly we can calculate Q_{fluid} by summing up the charges on the bottom plate, and on the side-plates, only along the regions submerged in the fluid.

$$Q_{fluid} = Q_{bottom} + Q_{left,sub} + Q_{right,sub} \quad (20)$$

We get the following plots, that describe the relation between Q_{top} vs. h/L_y and Q_{fluid} vs. h/L_y .

Figure 7: Charge variation with h - left: Q_{top} , right: Q_{fluid}

Qualitatively, as the height of the dielectric increases, the capacitance must increase as, the tank can store more charge in the dielectric compared to air. This is indeed observed. in the plot. But, the plots also describe that the rate of increase isn't linear, but looks like an exponential.

One can do the following thought experiment to understand why the increase isn't expected to be linear. Consider an increase of $0.1L_y$ in the height of the fluid. This means that the height of the air column inside decreases by $0.1L_y$. The strength of the electric field vector \vec{E} , is much more stronger near the top plate due to the large drop of 1V in a small region. So, naturally we expect that the change in the charge on the surface will be more for the same $0.1L_y$ change in the height of the fluid, if that change takes place near the top. So, **we don't expect the variation of the charge on the top surface Q_{top} to be linearly varying with h .**

6 Estimation of height using resonant frequency

Now that we have got an approximate relationship between h and Q_{top} ($= C$), we can now find the height h , given the capacitance C , which is calculated using Equation (5).

We can use two methods to find h from C :

1. We can use a simple linear interpolation to find an approximate analytical relation, that is piece-wise linear, between h and C , which we can then use to find h , given C .
2. From the plot of Q_{top} vs. h , we can see that it looks approximately of the form of $Q_{top} = \frac{1}{ah+b}$, which is expected because the tank can be approximately modelled as a series combination of an air capacitor of width $L_y - h$ and another fluid-filled capacitor of width h - which results in a net capacitance of $\frac{\epsilon_o \epsilon_r A}{h(1-\epsilon_r) + \epsilon_r L_y}$, where A is the area of cross-section. Thus, we can try to fit a rectangular hyperbola $y = \frac{1}{ax+b}$ to the observed plot, and get a relation between C and h .

7 Conclusion

A numerical solution to an analytically complex problem of finding the potential distribution inside the tank has been obtained, to a high degree of accuracy. Though it is not exact, this serves to help us analyse how the charge distribution and in turn, potential distribution changes with the height of the fluid, without using complex math and more importantly, efficiently.

Bibliography

- [1] Assignment 5 - EE2703 (Jan-May 2020).
- [2] Snell's law. <https://g.co/kgs/Vjb6FG>.