EE2703 - Applied Programming Lab
Final Exam

Hemanth Ram - EE18B132

July 30, 2020

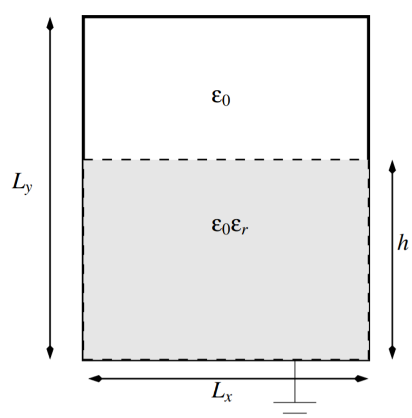
Overview

In this assignment, we try to calculate the potential, electric field, and the charge distribution of a capacitor partially filled with a dielectric fluid. We analyse the system for different amounts of the fluid. We compute the distributions using Laplace's equations, some continuity equations and the given boundary conditions. We then move on to verify the continuity of D_n at the boundary and also see if the system obeys Snell's law.

(a) Setting up Parameters and Conventions used

Since, the exact indexing and the parameters to be used have not been mentioned in the problem statement, I would like to list out all of them used in the program here. A $100x200$ mesh ($M = 200$ and $N = 100$) is used throughout and so, the step size is $0.1 cm/step$. The maximum number of iterations has been set to $Niter = 15000$ and an accuracy of $\delta = 10^{-3}$ is desired. The following diagram shows how the matrix used in the program maps to the tank in question.

```
  row-indexing
            [
  [0]         [  .. the first row is the top layer (1V) ..  ]
   .                              .
   .                              .
[k-1]         [  .. (k-1)th layer - just above boundary ..  ]
  [k]         [  ....    kth layer - the boundary    .....  ]
[k+1]         [  .. (k+1)th layer - just below boundary ..  ]
   .                              .
   .                              .
[M-1]         [  .....    bottom of the tank (0V)    .....  ]
            ]
  coloumn-indexing ->
              [  0 1 2 3     ................          N-1  ]
```



The index k in the matrix is calculated as :

$$k = \frac{L_y - h}{L_y} * M = (1 - \frac{h}{L_y}) * M$$

So, for example, for $h/L_y = 0.1$ and $h/L_y = 0.8$, $k_{0.1} = 180$ and $k_{0.8} = 40$ respectively. Also, whenever the center point of the mesh is in question, the point $(M/2, N/2)$ is considered. Since center point can also be taken as the point next to it, there may be slight variations from the original field at the center.

(b) Algorithm to determine h from the observed resonant frequency f

We know that the resonant frequency is given by $2\pi f = 1/\sqrt{LC}$. Also for the given setup, the capacitance C can be calculated as the series combination of

$$C_1 = \frac{A\epsilon_o}{L_y - h} \; and \; C_2 = \frac{A\epsilon_o\epsilon_r}{h}$$

which gives us :

$$\frac{1}{C} = \frac{1}{A\epsilon_o}(L_y - h + \frac{h}{\epsilon_r}) \; ... \; (1)$$

where,

$$A = Base\ area\ of\ the\ tank$$

$$h = Height\ till\ which\ fluid\ is\ filled$$

$$L_y = Total\ height\ of\ the\ tank$$

$$\epsilon_r = Relative\ permittivity\ of\ the\ fluid$$

So, given the above variables and the inductance of the inductor used in RLC circuit, the h required for the resonant frequency can be found out, using the expression mentioned above.

(c) Parallelizing the computation

The central part of the computaitons done involve updating very huge matrices using their old values. Using for-loops for doing this would lead python to a crawl and thus would slow down the process. To parallelize these operations, we use **vectorization**. We splice different parts of the matrices required for updation and perform the update with these, in a single line. For example the following for-loop :

```
for(i=1 ; i<Ny-1 ; i++ ){ // interior nodes
    for(j=1 ; j<Nx-1 ; j++ ){ // interior nodes
        phinew[i,j]=0.25*(phi[i,j-1]+phi[i,j+1]
                    +phi[i-1,j]+phi[i+1,j]);
        }
    }
```

can be converted to a single line in python by :

```
phi[1:-1,1:-1]=0.25*(phi[1:-1,0:-2]
    + phi[1:-1,2:]
    + top neighbours
    + bottom neighbours);
```

Once these updates are done, the boundary conditions have to be separately updated. In our case, the boundary condition is the continuity of D_n at the fluid's surface, for which the update equation is :

$$\phi_{k,n} = \frac{\epsilon_r \phi_{k-1,n} + \phi_{k+1,n}}{(1 + \varepsilon r)} \; ,$$

where k denotes the row corresponding to the boundary in the potential distribution matrix.

(d) Calculating the potential distribution in the given system

To do so, we use Laplace's equations and the continuity of D_n at the boundary to iteratively build the distribution. The update equation for the latter has been mentioned above. For Laplace's equations, we have :

$$\phi_{m,n} = \frac{\phi_{m-1,n} + \phi_{m+1,n} + \phi_{m,n+1} + \phi_{m,n-1}}{4},$$

Apart from these, we also should take care of the given boundary conditions, i.e the top part of the tank is maintained at 1V and the other sides are grounded. The following code snippet shows the mentioned updates :

```
# Laplace Update
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])


# D_n continuity Update
phi[k] = ( K*(phi[k+1]) + phi[k-1] ) / (K+1)

# Fortunately, we don't disturb values of the border of the tank
# So, we don't explicitly have to update them as 1V and 0V
```

Apart from the updates, we also keep track the error by making a copy of the previous iteration and take the maximum of the absolute error between them. This is done so that we could find the iteration when the desired accuracy is reached.

```
for _ in range(Niter):

    # Copy of previous iteration
    old_phi = phi.copy()

    ... update

    # Calculation of error
    errors[_] = (abs(old_phi-phi).max())
```

It is found that the errors vary exponentially with number of iterations after some 500 iterations. So, we approximately model the error as :

$$error = y = Ae^{bx}$$

We can obtain A and B by applying $LeastSquaresApproximation$ to the equation :

$$log(y) = log(A) + Bx$$

Since it is asked to extrapolate the error to infinity, the stopping condition would be :

$$\int_{N+0.5}^{\infty} Ae^{bk}dk = -\frac{A}{B}e^{B*(N+0.5)} < \delta$$

The following code snippet shows the above procedure :

```
# Creating matrix to perform LSA
M = ones((Niter-500,2))
M[:,0] = arange(501,Niter+1,1)

# Transposing the error to make it a column vector
c = c_[log(errors[500:])]

# Performing LSA with lstsq function
B,a1 = lstsq(M,c,rcond=None)[0]
A = exp(a1)
```

Putting all the things above, the following function returns the potential distribution and the required iterations to attain accuracy, given the size of the mesh (M x N), K (the dielectric constant of the fluid), dx (the step-size of the x and y axes), $hbyL$ (ratio of h and L_y), acc (the required accuracy) and $Niter$ (the maximum number of iterations).

```
def potential(M,N,K,dx,hbyL,acc,Niter):
    k = int(M*(1-hbyL))
    phi = zeros((M,N))
    phi[0] = np.ones((1,N))
    errors = ndarray((Niter,1))
    for _ in range(Niter):
        old_phi = phi.copy()
        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])
        phi[k] = ( K*(phi[k+1]) + phi[k-1] ) / (K+1)
        errors[_] = (abs(old_phi-phi).max())

    M = ones((Niter-500,2))
    M[:,0] = arange(501,Niter+1,1)
    c = c_[log(errors[500:])]
    B,a1 = lstsq(M,c,rcond=None)[0]
    A = exp(a1)

    for req_iter in range(500,Niter,1):
        er = (A*((exp(B*(req_iter+0.5)))))/(-B)
        if(er < acc):
            break

    return phi,req_iter
```

The output of the code above for different values of h with $Niter = 15000$ and $\delta = 10^{-3}$ were found to be :

```
Plotting Contours .. please wait ..
Required iterations for h/Ly =  0.1 => 14412
Required iterations for h/Ly =  0.2 => 14371
```

```
Required iterations for h/Ly =  0.3 => 14256
Required iterations for h/Ly =  0.4 => 14003
Required iterations for h/Ly =  0.5 => 13600
Required iterations for h/Ly =  0.6 => 13258
Required iterations for h/Ly =  0.7 => 13330
Required iterations for h/Ly =  0.8 => 13799
Required iterations for h/Ly =  0.9 => 14382
Done
```

(e) Charge accumulated in different parts of the tank

To calculate charge we use Gauss' law, for which we would require the electric fields distribution, which can be easily obtained from the potential distribution (*phi*) by differentiation.

$$\vec{E} = -\nabla\phi \text{ , } \textit{which can be vectorized as}$$

$$E_{x,(i,j)} = \frac{\phi_{i,j-1} - \phi_{i,j+1}}{\Delta x} \text{ , } E_{y,(i,j)} = \frac{\phi_{i-1,j} - \phi_{i+1,j}}{\Delta y}$$

Once we have found electric field at all points, we know that : $\int Eds = q/\epsilon$. For the top part, we have $ds = W.dx$, where W is the width of the tank (the unknown dimension of the tank). So, we now have

$$\frac{q}{W} = \epsilon_o \int Edx \approx \epsilon_o .\Delta x. \sum_{x=0}^{N-1} E_{y(1,x)}$$

Essentially, we take the summation of the electric field's normal components just inside the tank, to get the charge per unit width at the top. Similarly, we can get the charge on the portion of the wall touching the dielectric fluid as the sum of :

$$Charge\ at\ bottom\ surface \approx \epsilon_o\epsilon_r .\Delta x. \sum_{x=0}^{N-1} E_{y(-2,x)} \ \ and$$

$$Charge\ at\ walls\ inside\ fluid \approx 2 * \epsilon_o\epsilon_r .\Delta y. \sum_{y=0}^{M-1} E_{x(y,1)}$$

The second equation is multiplied by 2 to take into account the charge on both sides, rather than computing it separately, due to symmetry. The following code snippet shows the above computations :

```
# $phi$ for field calculation
phi = potential(M,N,2,1,hbyL,10**(-7),15000)

# initialising Ex, Ey arrays
Ex = zeros((M, N)); Ey = zeros((M, N))

# filling Ex, Ey arrays with fields
Ex[1:-1, 1:-1] = (phi[1:-1, 2:] - phi[1:-1, 0:-2])/delta
Ey[1:-1, 1:-1] = (phi[2:, 1:-1] - phi[0:-2, 1:-1])/delta
```

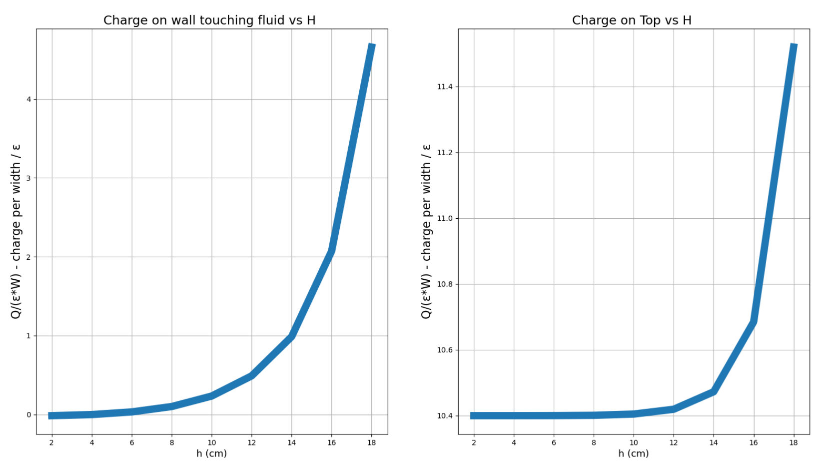
```
# Charge at top
q_top.append(-sum(Ey[1])*delta)

# Charge on tank inside fluid
q_fluid_bottom = 2*(sum(Ey[-2]))*delta
q_fluid_wall = 4*(sum(Ex[k:,1]))*delta
q_fluid.append(q_fluid_bottom+q_fluid_wall)
```

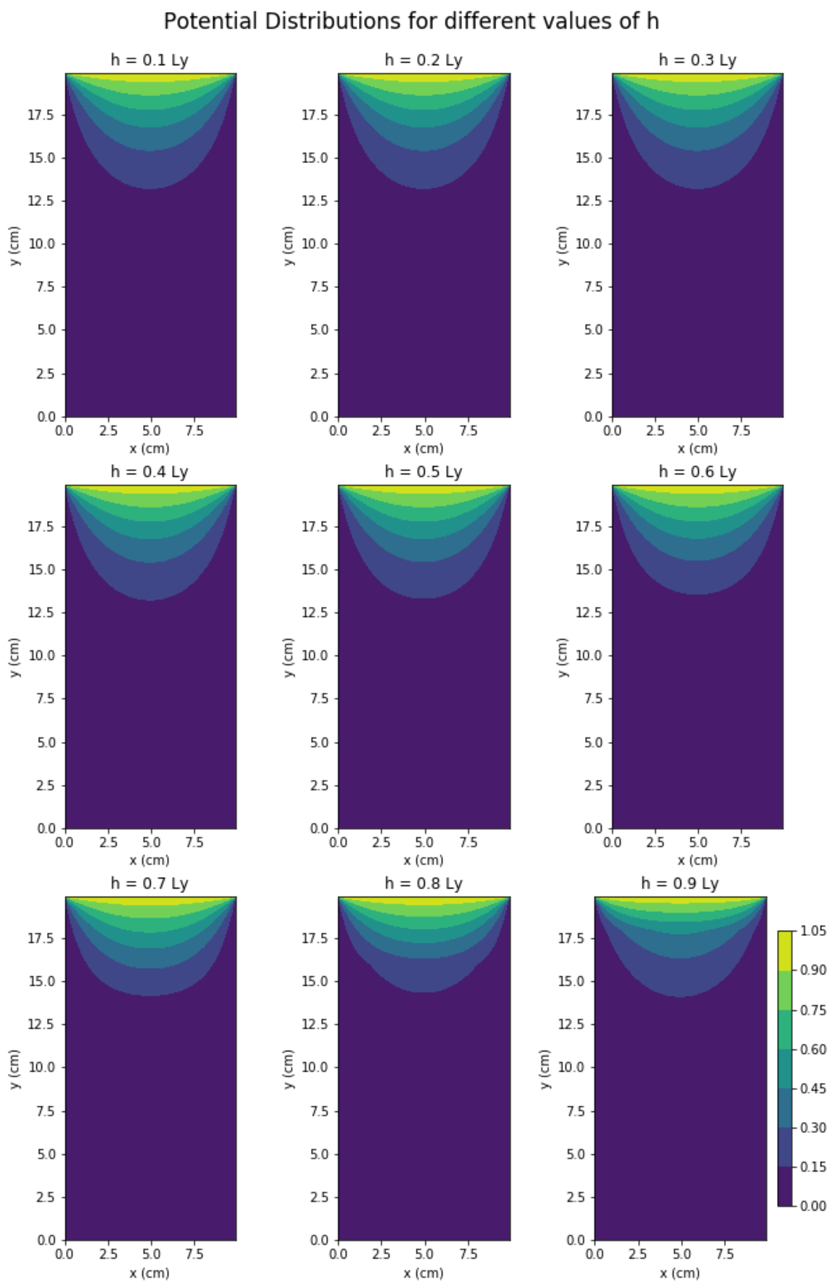
The charges were computed for different values of h/L_y. From equation (1), substituting $\epsilon_r = 2$, we get the following relation :

$$C \propto \frac{1}{L_y - h/2}$$

So, as h increases, C increases inversely and we expect a hyperbolic relation between C and h. Since the charge $q \propto C$ for a capacitor, we can tell that q would approximately follow the same relation with h as that of C. So we expect it not to be linear, rather expect it to be hyperbolic. The generated plots of q_{top} vs h/L_y and and q_{fluid} vs h/L_y are shown below.



As expected, both the plots are not linear, and increase rapidly after some h like the function mentioned in equation (1). Though, Q does not vary linearly with C ideally, it was a good approximation for us to arrive at this result. The potential distributions for different hs were also plotted while calculating the charges :



(f) $\vec{E}$ Calculation at center and continuity of D_n

The electric field can be calculated at all points using the method described above. After doing so, we take the E_x and E_y at the index $N/2$ in the $M/2^{th}$ row, which describes the boundary in the matrix, to get the electric field exactly at the center. By doing so, the electric field was found to be :

```
Ex at center :
-0.0003512403176645501 V / cm
Ey at center :
-0.03376457217756437 V / cm
```

Now, to prove the continuity of D_n at the boundary, we have to prove :

$$D_{n(air)} = D_{n(fluid)} \text{ .. } (at\ boundary)$$
$$\implies \epsilon_{air} E_{y(air)} = \epsilon_{fluid} E_{y(fluid)} \text{ .. } (2)$$

Here, $E_{y(air)}$ is the electric field just above the boundary, which can be taken from the row just above the boundary row in the electric field distribution matrix. Similarly, for $E_{y(fluid)}$, we take the row just below it. and for any point in the boundary equation (2) must be satisfied. In other words,

$$\frac{E_{y(k+1,i)}}{E_{y(k-1,i)}} = \frac{\epsilon_{air}}{\epsilon_{fluid}} \ \ \forall \ \ 0 \leqslant i < N \ \ ,$$

where k refers to the index of the row representing the boundary in the matrix. So, by taking the $(k-1)^{th}$ row and $(k+1)^{th}$ row of E_y and performing element-wise division, we got the following array as output :

```
E_fluid/E_air at all points in boundary:
[0.48782275 0.48782328 0.48782416 0.4878254  0.48782698 0.48782889
 0.48783113 0.4878337  0.48783657 0.48783975 0.4878432  0.48784693
 0.48785091 0.48785514 0.48785958 0.48786423 0.48786906 0.48787406
 0.4878792  0.48788447 0.48788984 0.48789529 0.48790079 0.48790634
 0.4879119  0.48791745 0.48792297 0.48792844 0.48793383 0.48793912
 0.4879443  0.48794934 0.48795423 0.48795893 0.48796344 0.48796773
 0.4879718  0.48797561 0.48797916 0.48798244 0.48798542 0.48798811
 0.48799048 0.48799253 0.48799426 0.48799564 0.48799669 0.48799738
 0.48799773 0.48799773 0.48799738 0.48799669 0.48799564 0.48799426
 0.48799253 0.48799048 0.48798811 0.48798542 0.48798244 0.48797916
 0.48797561 0.4879718  0.48796773 0.48796344 0.48795893 0.48795423
 0.48794934 0.4879443  0.48793912 0.48793383 0.48792844 0.48792297
 0.48791745 0.4879119  0.48790634 0.48790079 0.48789529 0.48788984
 0.48788447 0.4878792  0.48787406 0.48786906 0.48786423 0.48785958
 0.48785514 0.48785091 0.48784693 0.4878432  0.48783975 0.48783657
 0.4878337  0.48783113 0.48782889 0.48782698 0.4878254  0.48782416
 0.48782328 0.48782275]

Average value of the ratio is
0.48791123630262306
```

Since $\epsilon_{fluid} = 2$, as expected, the value of $\frac{E_{y(k+1,i)}}{E_{y(k-1,i)}}$ at all points of the boundary is close to $\frac{\epsilon_{air}}{\epsilon_{fluid}}$ (0.5 in our case). Hence, the continuity of D_n has been verified.

(g) Is Snell's Law agreed here ?

The change in angle at the boundary can be found by taking difference of the angle of the electric field just above the boundary and below the boundary. The angle is found taking $tan^{-1}(E_x/E_y)$. So,

$$\Delta\theta = tan^{-1}(E_{x(k+1,n/2)}/E_{y(k+1,n/2)}) - tan^{-1}(E_{x(k-1,n/2)}/E_{y(k-1,n/2)}) \ ,$$

where $(k, n/2)$ refers to the center of the mesh, for $h/L_Y = 0.5$. By doing so, the following output was obtained :

```
Incident Angle :
0.471 degrees
Transmitted Angle :
0.879 degrees
Change in Angle :
0.408 degrees
```

To check Snell's Law, we have to check if $\frac{sin(i)}{sin(r)} = \frac{1}{\sqrt{\epsilon_{fluid}}}$ From the obtained i and r, $\frac{sin(i)}{sin(r)}$ comes out to be approximately 0.53. But $\frac{1}{\sqrt{\epsilon_{fluid}}} \approx 0.7$. So, Snell's law is not obeyed here. It is because Snell's law is valid only for propagating electromagnetic waves. The electric field in analysis is static, so there is no varying magnetic field which could cause propagation. Another reason might be that the field is almost normally incident at the boundary, as it evident from the very small angles found out.

Conclusion

We successfully calculated the potential distributions for different amounts of fluid in the tank and also found the electric field and charge accumulations. We verified the continuity of D_n at the boundary and found that, at the boundary, the electric field didn't obey Snell's law, as it is static in nature, and almost normally incident.