ECSE: 543

Numerical Methods for Electrical Engineering

ECE Department

McGill University

Assignment – 2: Report

Presented to:

Dennis Giannacopoulos

Presented by:

Alok Patel (260954024)

Date: 16th November 2020

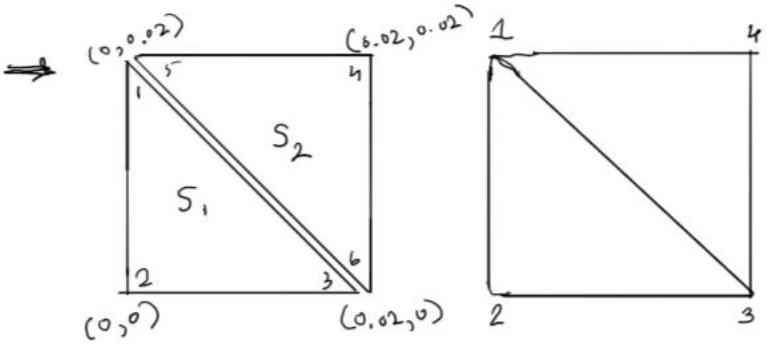
Index

Chapter-1: Abstract

In this report the detailed description of the assignment's solution is presented, with the plots, results and the methodology are presented. First, the local S-matrix for the two first-order triangular finite elements which is used to solve the equation for the electrostatic potential is formed and the Conjoint S matric is also formed. Second, the mesh is formed for the cross-section of an electrostatic problem. And using the SIMPLE2D code provided the potential at the nodes of the mesh is found. Also, using this node voltages, the capacitance per unit length is found. Lastly, the code for the conjugate gradient method to solve the matrix equation for the finite difference node spacing is formed and the potentials are calculated. And the comparison for the same is done with the Choleski Decomposition, and SOR method. At last, the strategy to calculate the capacitance per unit length of the system is discussed and it is calculated.

Chapter-2: Question-1

In this part of the assignment, the question asked to develop the S-matrix for the first order triangular finite elements used to solve for the electrostatic potential. Here in total three S matrix is found, two for local disjoint triangles (answer in green box), and one for the global (conjoint) triangle (answer in green box). Below provides the solution for the given problem.



$$S_{ij}^{(e)} = \int_{\Delta e} \nabla\alpha_i \cdot \nabla\varphi_j \, dS$$

→ For Δ^{le} 1 the S matrix is derived as below: $(A = \frac{1}{2} \times bh)$

$$\therefore S_1^{(e)} = \begin{bmatrix} S_{11}^{(e)} & S_{12}^{(e)} & S_{13}^{(e)} \\ S_{21}^{(e)} & S_{22}^{(e)} & S_{23}^{(e)} \\ S_{31}^{(e)} & S_{32}^{(e)} & S_{33}^{(e)} \end{bmatrix}$$

and $A = \frac{1}{2} \times 0.02 \times 0.02$

$$A = 2 \times 10^{-4}$$

where, $S_{11}^{(e)} = \frac{1}{4A}\left[(y_2 - y_3)^2 + (x_3 - x_2)^2\right] = \frac{1}{4 \times 2 \times 10^{-4}}\left[0 + (0.02)^2\right]$

$$\therefore S_{11}^{(e)} = 0.5$$

$$S_{22}^{(e)} = \frac{1}{4A}\left[(y_3 - y_1)^2 + (x_1 - x_3)^2\right] = \frac{1}{4 \times 2 \times 10^{-4}}\left[(0.02)^2 + (0.02)^2\right]$$

$$\therefore S_{22}^{(e)} = 1$$

$$S_{33}^{(e)} = \frac{1}{4A}\left[(y_1 - y_2)^2 + (x_2 - x_1)^2\right] = \frac{1}{4 \times 2 \times 10^{-4}}\left[(0.02)^2 + 0\right]$$

$$\therefore S_{33}^{(e)} = 0.5$$

$$S_{12}^{(e)} = \frac{1}{4A}\left[(y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3)\right]$$

$$= \frac{1}{4 \times 2 \times 10^{-4}}\left[0 + (0.02)(-0.02)\right]$$

$$\therefore S_{12}^{(e)} = -0.5 \quad ; \text{ also, } S_{21}^{(e)} = S_{12}^{(e)} = -0.5$$

⇒ Now, calculating other components,

$$S_{13}^{(e)} = \frac{1}{4A}\left[(y_2-y_3)(y_1-y_2) + (x_3-x_2)(x_2-x_1)\right]$$

$$= \frac{1}{4\times 2\times 10^{-4}}\left[0 + (0.02)(0)\right]$$

$$\therefore S_{13}^{(e)} = 0 \quad ; \text{ also, } S_{31}^{(e)} = S_{13}^{(e)} = 0$$

$$S_{23}^{(e)} = \frac{1}{4A}\left[(y_3-y_1)(y_1-y_2) + (x_1-x_3)(x_2-x_1)\right]$$

$$= \frac{1}{4\times 2\times 10^{-4}}\left[(-0.02)(0.02) + (-0.02)(0)\right]$$

$$\therefore S_{23}^{(e)} = -0.5 \quad ; \text{ also } S_{32}^{(e)} = S_{23}^{(e)} = -0.5$$

⇒ Thus, matrix 'S' of Δ'e 1 is:

$$S_1^{(e)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$$

⇒ Forming element of stiffness matrix for Δ'e-2.

$$\therefore S_2^{(e)} = \begin{bmatrix} S_{44}^{(e)} & S_{45}^{(e)} & S_{46}^{(e)} \\ S_{54}^{(e)} & S_{55}^{(e)} & S_{56}^{(e)} \\ S_{64}^{(e)} & S_{65}^{(e)} & S_{66}^{(e)} \end{bmatrix}$$

→ Getting elements of the matrix.

$$S_{44}^{(e)} = \frac{1}{4A}\left[(y_5-y_6)^2 + (x_6-x_5)^2\right] = \frac{1}{4\times 2\times 10^{-4}}\left[(0.02)^2 + (0.02)^2\right]$$

$$\therefore S_{44}^{(e)} = 1$$

$$S_{55}^{(e)} = \frac{1}{4A}\left[(y_6-y_4)^2 + (x_6-x_5)^2\right] = \frac{1}{4\times 2\times 10^{-4}}\left[(0.02)^2 + (0)\right]$$

$$\therefore S_{55}^{(e)} = 0.5$$

$$S_{66}^{(e)} = \frac{1}{4A}\left[(y_4-y_5)^2 + (x_5-x_4)^2\right] = \frac{1}{4\times 2\times 10^{-4}}\left[(0)^2 + (0.02)^2\right]$$

$$\therefore S_{66}^{(e)} = 0.5$$

$$S_{45}^{(e)} = \frac{1}{4A}\left[(y_5 - y_6)(y_6 - y_4) + (x_6 - x_5)(x_4 - x_6)\right]$$

$$\therefore S_{45}^{(e)} = \frac{1}{4 \times 2 \times 10^{-4}}\left[(0.02)(-0.02) + (0.02)(0)\right]$$

$$\therefore \boxed{S_{45}^{(e)} = -0.5} \quad \& \text{ also, } \boxed{S_{54}^{(e)} = S_{45}^{(e)} = -0.5}$$

$$S_{56}^{(e)} = \frac{1}{4A}\left[(y_6 - y_4)(y_4 - y_5) + (x_4 - x_6)(x_5 - x_4)\right]$$

$$\therefore S_{56}^{(e)} = \frac{1}{4 \times 2 \times 10^{-4}}\left[(-0.02)(0) + 0\right]$$

$$\therefore \boxed{S_{56}^{(e)} = 0} \quad ; \text{ also, } \boxed{S_{65}^{(e)} = S_{56}^{(e)} = 0}$$

⇒ Thus, matrix 'S' of Δle 2 is:

$$S_2^{(e)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix} \quad (*)$$

⇒ Calculating 's' matrix for the conjoint triangles:

$$\therefore S_{con}^{(e)} = \begin{bmatrix} S_1^{(e)}{}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & S_2^{(e)}{}_{3\times3} \end{bmatrix}$$

⇒ Connecting Matrices:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} & 1 & 2 & 3 & 4 \\ 1 & 1 & & & \\ 2 & & 1 & & \\ 3 & & & 1 & \\ 4 & & & & 1 \\ 5 & 1 & & & \\ 6 & & & 1 & \end{bmatrix}_{6\times4} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$

Disjoint — Conjoint

$$V_{dis} = \boxed{C}\, V_{con}$$

⇒ Thus, conjoint matrix can be formed using,

$$\therefore S = C^T S_{con} C$$

→ After multiplication of Matricies the resultant S Matrix is;

$$\therefore S = \begin{bmatrix} S_{11}+S_{55} & S_{12} & S_{13}+S_{56} & S_{54} \\ S_{21} & S_{22} & S_{23} & 0 \\ S_{31}+S_{65} & S_{32} & S_{33}+S_{66} & S_{64} \\ S_{45} & 0 & S_{46} & S_{44} \end{bmatrix}_{4\times4}$$

⇒ Thus, conjoint matrix is,

$$\therefore S = \begin{bmatrix} 1 & -0.5 & 0 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix}_{4\times4}$$

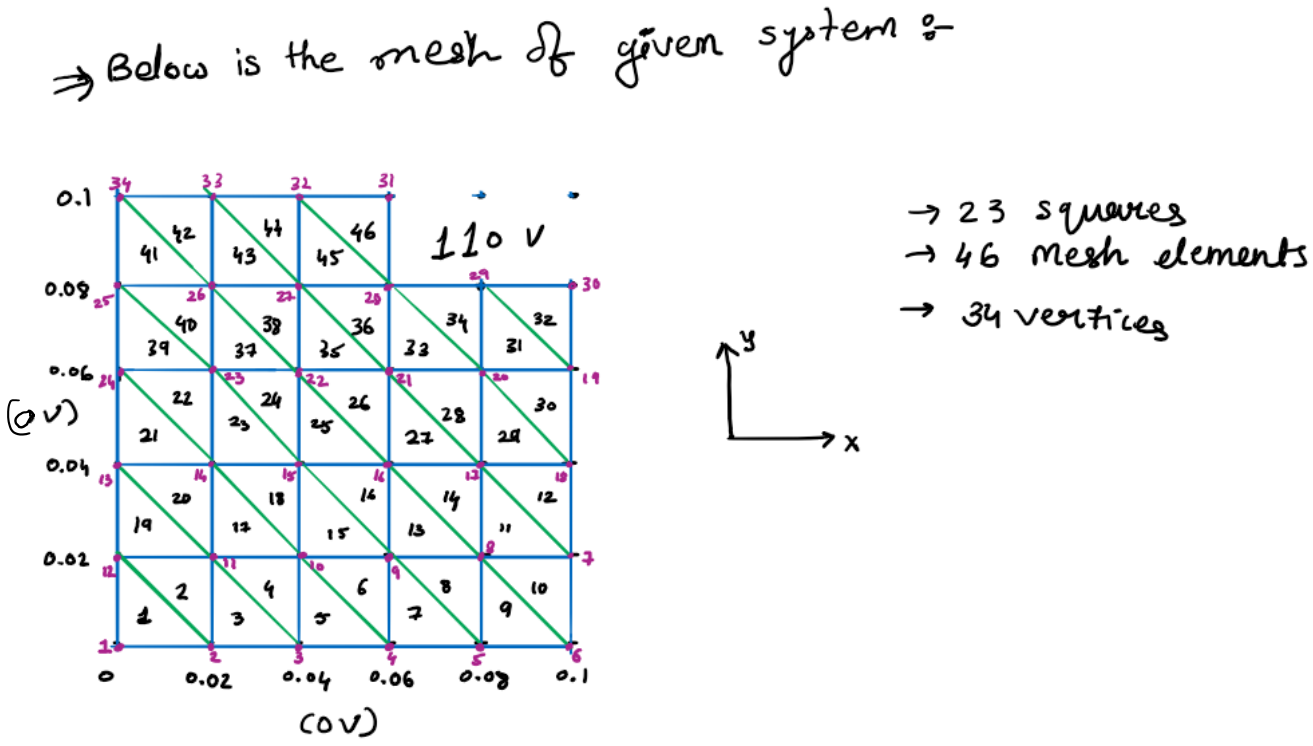
Ans

Chapter-3: Question-2

In this question, the mesh is to be created and the potential at the nodes is to be computed using the data input of the mesh to the SIMPLE2D_M.m file provided to us. The data input file is named as input_data.txt, which is organized as asked by the course instructor.

3.1 Finite Element Mesh

The mesh is shown below, the element number is number with the black ink, the purple ink indicates the node number and triangle shape is taken for the mesh element. Look at the data file for the input data according to the number specification provided here.



The output of the MATLAB code for getting the potential at all the nodes using the SIMPLE2D_M.m file which is accessed using other MATLAB file named Q_2.m, also the capacitance value is also computed in the same file, is shown below. This code is provided in the Appendix-A.

Node	x	y	Voltage
1.0000	0	0	0
2.0000	0.0200	0	0
3.0000	0.0400	0	0
4.0000	0.0600	0	0
5.0000	0.0800	0	0
6.0000	0.1000	0	0
7.0000	0.1000	0.0200	23.2569
8.0000	0.0800	0.0200	22.2643
9.0000	0.0600	0.0200	19.1107
10.0000	0.0400	0.0200	13.6519

```
   11.0000    0.0200    0.0200    7.0186
   12.0000         0    0.0200         0
   13.0000         0    0.0400         0
   14.0000    0.0200    0.0400   14.4223
   15.0000    0.0400    0.0400   28.4785
   16.0000    0.0600    0.0400   40.5265
   17.0000    0.0800    0.0400   46.6897
   18.0000    0.1000    0.0400   48.4989
   19.0000    0.1000    0.0600   77.3592
   20.0000    0.0800    0.0600   75.4690
   21.0000    0.0600    0.0600   67.8272
   22.0000    0.0400    0.0600   45.3132
   23.0000    0.0200    0.0600   22.1921
   24.0000         0    0.0600         0
   25.0000         0    0.0800         0
   26.0000    0.0200    0.0800   29.0330
   27.0000    0.0400    0.0800   62.7550
   28.0000    0.0600    0.0800  110.0000
   29.0000    0.0800    0.0800  110.0000
   30.0000    0.1000    0.0800  110.0000
   31.0000    0.0600    0.1000  110.0000
   32.0000    0.0400    0.1000   66.6737
   33.0000    0.0200    0.1000   31.1849
   34.0000         0    0.1000         0

------------------------------------------------------------
The Potential at the point (0.06,0.04) is: 40.5265 V
------------------------------------------------------------
The capacitance per unit length of the system is: 5.2113e-11 F
------------------------------------------------------------
```

3.2 Capacitance Value

⇒ To compute the capacitance per length we can use the energy equation:

$$\mathcal{E} = \frac{1}{2} C V^2$$

→ To calculate the total energy of the systems, we use $\mathcal{E}_i = \frac{\epsilon_0}{2} U_i^T S_i U_i \quad ; \; i \in [0, 46]$ ↳ all elements.

→ Thus, $E_{Total} = \sum_{i=1}^{46} E_i = \frac{1}{2} C V^2 \quad ; V = 110 \text{ Volts}$

→ For two types of shapes we have two fixed 'S's:

(i) For "△" ↳ $S_a = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$

(ii) For "▽" ↳ $S_b = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$

By computing the capacitance value of the provided system, using the potential values computed and applying the equations provided to us, using the code written in the file Q_2.m file, the capacitance value is:

C = 52.113 pF

Chapter-4: Question-3

In this part of the question, the program to solve the problem, using the conjugate gradient method is shown. The Program for this is provided in Appendix-B, and the file name is *CG.m*.

4.1 Converting to PD

By creating the A matrix for the given system for the 19 unknowns (see Appendix-C) the A matrix is not P.D., as the diagonal elements are negative the square root, which is essential in the Choleski Decomposition, the A matrix is not P.D. Not only, the A matric in not PD, it is also not symmetric, because we apply the plane of symmetry at the border condition. See Appendix-C for more details. Thus, it is essential to convert it to P.D., this can be done by multiplying -1 both side to the Ax = b equation.

$$-Ax = -b$$

Or if we multiply both side to equation Ax = b with A^T then the new equation will have both the feature of Symmetry and PD.

$$A^T Ax = A^T b \;\; \rightarrow \;\; Sx = c$$

Thus, now the diagonal elements will be positive, and the non-diagonal elements will become negative. Now, using this we can solve this linear equation by Choleski decomposition, and Conjugate Gradient as well. One thing to point out that by just converting the A to PD, and not converting it to symmetry, we can directly solve through CG.

4.2 CD and CG

First, solving the Matrix equation by CD by solving Sx = c equation, the potential at the point (0.06, 0.04) is **40.5265 V**. And the Output x is as below:

```
The x matrix is :
    7.0186
   13.6519
   19.1107
   22.2643
   23.2569
   14.4223
   28.4785
   40.5265
   46.6897
   48.4989
   22.1921
   45.3132
   67.8272
   75.4690
   77.3592
   29.0330
   62.7550
   31.1849
   66.6737
```

Now solving the equation using the conjugate method by solving for the Sx = c equation is as shown below for the system of node number as shown in Appendix-C. The potential at the point (0.06, 0.04) is **40.5265 V**. But,

if we use the - Ax = - b and solve for the CG, then we can obtain 40.5262 V, which shows that there is some error in the potential if A is PD but not symmetric.

```
-----------------------------------------
  |  Node   | Voltage|
-----------------------------------------
     1       7.01855
     2       13.6519
     3       19.1107
     4       22.2643
     5       23.2569
     6       14.4223
     7       28.4785
     8       40.5265
     9       46.6897
    10       48.4989
    11       22.1921
    12       45.3132
    13       67.8272
    14        75.469
    15       77.3592
    16        29.033
    17        62.755
    18       31.1849
    19       66.6737
-----------------------------------------------------
The Potential at the point (0.06,0.04) is: 40.5265 V
=====================================================
```

4.3 Plot a graph

The question asks to plot the curve of two-norm and the infinity norm vs the number of iterations to converge. So, the code is written in the same *CG.m,* file using the formulas for calculating the values of the two-norm and infinity norm and storing in the vector and it is used to plot the curve. The plot of two-norm residual vs the Number of iterations is given in the Figure-1, and the plot of Infinite-norm residual vs Number of iterations is given in Figure-2.

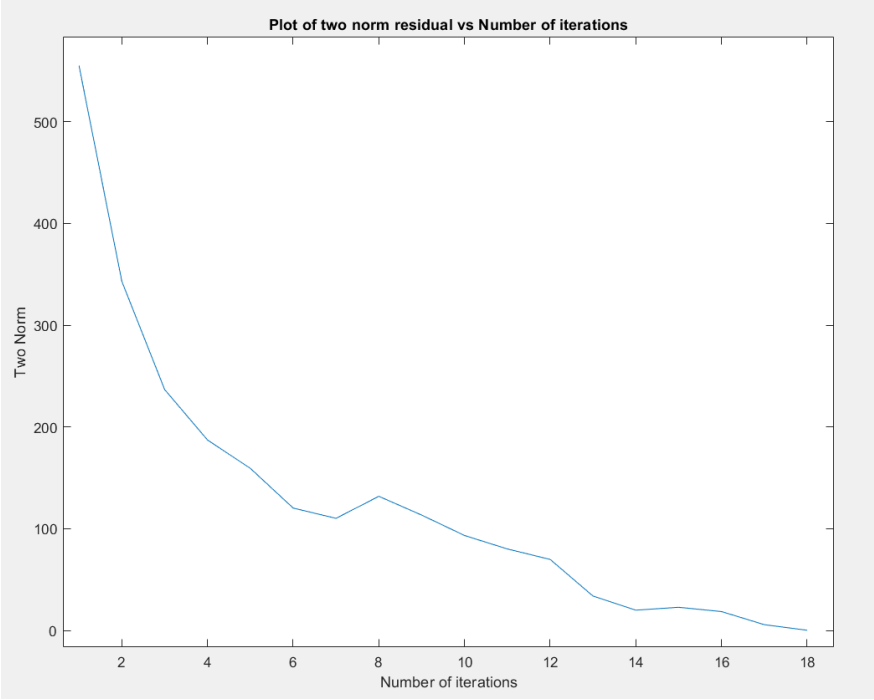


Figure-1: Plot of Two-norm vs Number of Iterations

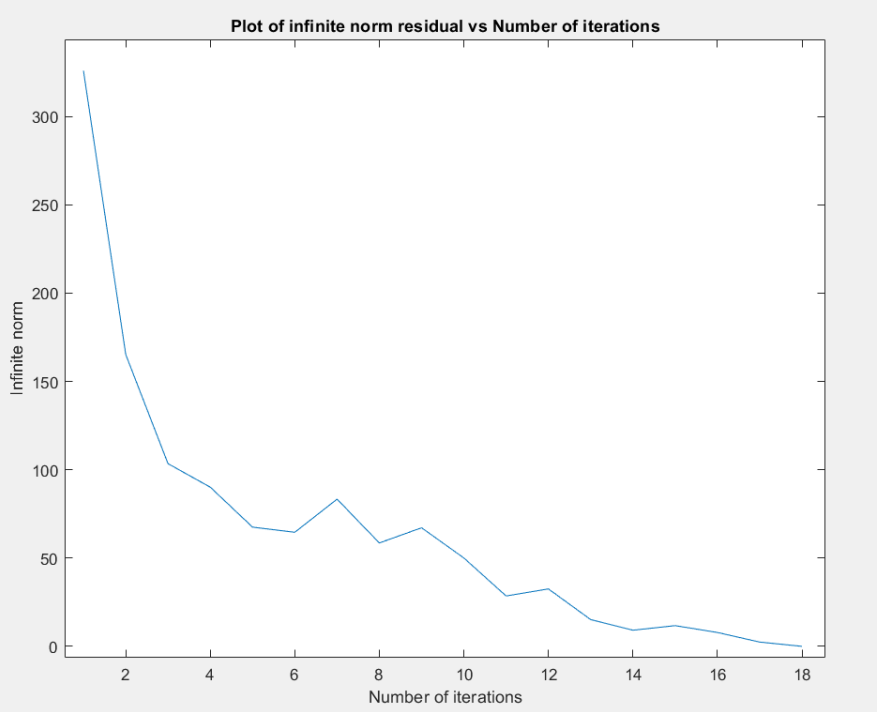


Figure-2: Plot of Infinity-norm vs Number of Iterations

4.4 Comparison

If we compare the three methods, then we are getting the same result by all the three with the accuracy of four decimal digits. If the number of decimal digits get increased, in my opinion there will be very minor difference in the answer after the four-decimal place. As for both the CG and SOR the node spacing is kept constant and both are the iterative process, and I also set the residual tolerance value of 10^{-5}, thus I think we are getting the same answer. And as the CD is highly efficient method for the small system, which indeed this system is, so I guess for CD we are getting the same result. Apparently, the FEM value is calculated by the MATLAB file, which is very efficient code to calculate the potential at the nodes. In conclusion, all the method provides the same results by four decimals.

CD	CG	FEM	SOR
40.5265 V	40.5265 V	40.5265 V	40.5265 V

4.5 Capacitance Value

To calculate the capacitance per length, using the potential values found from the Conjugate Gradient method, using the same equation used for the finite element mesh in Question-2 of the assignment, we can solve the capacitance value. Thus, using the same methodology and calculating the total energy and finding the capacitance value using the energy equation (Appendix-D). The potential at the nodes is set manually taken from the output of the CG. The capacitance per unit length is:

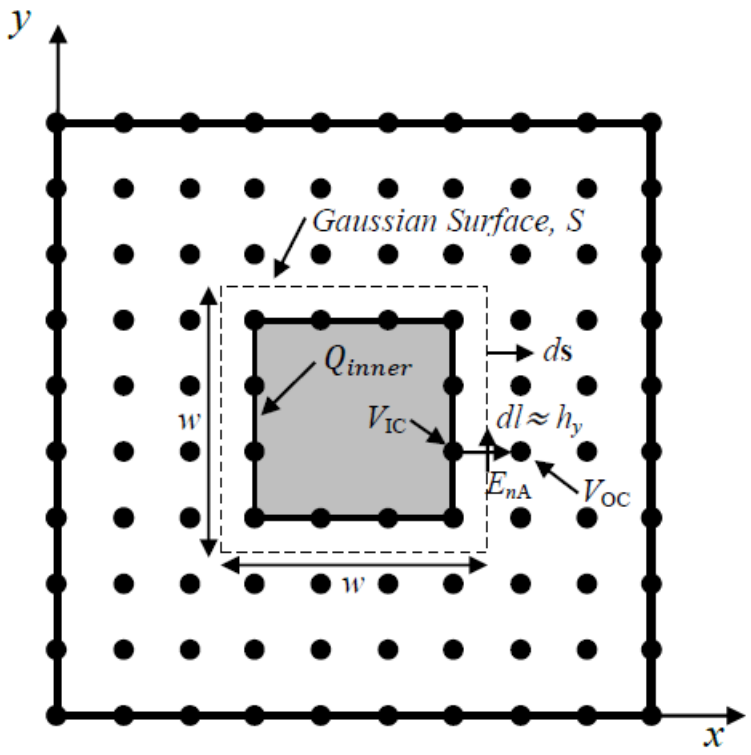
$$C = 5.2113 \times 10^{-11} = 52.113\, pF$$

The value is similar to what we get earlier, this is because the voltage at the nodes is very near to what obtained through the Finite Mesh Method. There is also another method which can be implemented to get the

capacitance per unit length, which is by estimating that the potential at all the points of the cable is uniformly distributed. Then calculating the inner charge value and then calculating the capacitance, by equation as below:

$$\frac{C}{L} = \frac{1}{V}\left(\frac{Q}{L}\right)$$

Now, to calculate the charge of the conductor the Gauss law can be use, and using the Electric Field which is normal to the counter is to be calculated to get the approx. charge. In MATLAB, we can approximate the integration with the summation, and this way the charge is found. Qinner and Voc is nothing but the Q and V provided in the equation.



Reference: https://www.waves.utoronto.ca/prof/svhum/ece221/labs/ECE221_Lab5.pdf

Chapter-5: Conclusion

In conclusion, the assignment helps to understand the FEM concepts, by questioning to build the S matrix for the conjoint and the two disjoint triangles. Then, for the potential problem the Finite Element is to be set and using the program provided the potential at the nodes is calculated. Then the code for the conjugate gradient method is written, and for the same problem the potential at the nodes is calculated using the A matrix formed for the problem. Here the A matrix was not PD and non-symmetric, so it is converted to PD and symmetric and then the problem is solved by Choleski Decomposition, and Conjugate Gradient method. Various experiments of the comparison were done and the graph of the residual is plotted for the understanding of the concept of the two norm and the infinity norm. Also, the capacitance of the system was found by the analogy of the total energy.

Chapter-6: Appendix

A. Question-2

```
%% ------------------ ESCE:543-Numerical Methods for Electrical Engineering -------
---------- %%
% Assignmnet-2: Q-2: (a),(b)&(c) - Solving Potential and the Capacitance
Calculations.
% The file which is used to read to data for the calculations, and it is named as
input_data.txt
% The function is called from the other file provided to us named as SIMPLE2D_M.m
% The output of this file is the potential at (0.06,0,04) viz. 16th node.
% Other output is the capacitance calculation per unit length between the conductor
and ground.
%% ----- Code Start(Calculating the potential) ----- %%
clear
clc
%Potential = SIMPLE2D_M('xy_data.txt','sequence_data.txt','boundary_data.txt');
Potential = SIMPLE2D_M('input_data.txt');
aa = 'The Potential at the point (0.06,0.04) is: ';
bb = Potential(16,:); cc = ' V';

disp('-----------------------------------------');
disp('  |  Node  |     x    |    y   | Voltage|')
disp('-----------------------------------------');
disp(Potential)
disp('-------------------------------------------------------------');
disp([aa num2str(bb(4)) cc])
disp('-------------------------------------------------------------');

%% ----- Calculating the capacitance per unit length ----- %%

fid = fopen('input_data.txt','r');
data = textscan(fid, '%f%f%f%f',  'CollectOutput', 1);
fclose(fid);
data = data{1};
%disp(data)
Sa = [1 -0.5 -0.5;
     -0.5 0.5 0;
     -0.5 0 0.5];
Sb = [0.5  -0.5 0;
     -0.5 1 -0.5;
      0 -0.5 0.5];
E = zeros(46,1);
U = zeros(3,1);
V = 110;
vert = 35;      % 35 because in data the vertex data is stored from that position.
Energy = 0;
eps = 8.85e-12;
for i=1:46
    if mod(i,2)~=0
        S = Sa;
        %disp('Odd')
    else
        S = Sb;
        %disp('Even')
    end
```

```
    vertex = data(vert,1:3);
    %disp(vertex)
    U(1,1) = Potential(vertex(1),4);
    U(2,1) = Potential(vertex(2),4);
    U(3,1) = Potential(vertex(3),4);
    %disp(U)
    %disp(S)
    E(i) = (U'*S*U);
    %disp(E(i))
    Energy = Energy + (E(i)/2);
    vert = vert+1;
end
Ener = 4*eps*Energy;          %4 times because converted 1/4th.
C = (2*Ener)/(V^2);
ee = ' F';
dd = 'The capacitance per unit length of the system is: ';
disp([dd num2str(C) ee])
disp('------------------------------------------------------------');

%% ----- Code End ----- %%
```

B. Conjugate Gradient

```
%% ------------------ ESCE:543-Numerical Methods for Electrical Engineering -------
---------- %%
% Assignmnet-2: Q-3: Conjugate Method to solve for the potential at points.

%% ----- Code Start ----- %%
clear
clc
%{
b = [0 0 0 0 0 0 0 0 0 0 0 0 -110 -110 -110 0 -110 0 -110]';
A = [-4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
     1 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
     0 1 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
     0 0 1 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
     0 0 0 2 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0;
     1 0 0 0 0 -4 1 0 0 0 1 0 0 0 0 0 0 0 0;
     0 1 0 0 0 1 -4 1 0 0 0 1 0 0 0 0 0 0 0;
     0 0 1 0 0 0 1 -4 1 0 0 0 1 0 0 0 0 0 0;
     0 0 0 1 0 0 0 1 -4 1 0 0 0 1 0 0 0 0 0;
     0 0 0 0 1 0 0 0 2 -4 0 0 0 0 1 0 0 0 0;
     0 0 0 0 0 1 0 0 0 0 -4 1 0 0 0 1 0 0 0;
     0 0 0 0 0 0 1 0 0 0 1 -4 1 0 0 0 1 0 0;
     0 0 0 0 0 0 0 1 0 0 0 1 -4 1 0 0 0 0 0;
     0 0 0 0 0 0 0 0 1 0 0 0 1 -4 1 0 0 0 0;
     0 0 0 0 0 0 0 0 0 1 0 0 0 2 -4 0 0 0 0;
     0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 -4 1 1 0;
     0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 -4 0 1;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 -4 1;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 -4];
 n = 19;
%}
%{
% Below is the data for the new b and A with A as PD.
 b = [0;0;0;0;0;0;0;0;0;0;0;0;110;110;110;0;110;0;110];
 A = [4,-1,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0;
        -1,4,-1,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0;
        0,-1,4,-1,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0;
```

```
        0,0,-1,4,-1,0,0,0,-1,0,0,0,0,0,0,0,0,0,0;
        0,0,0,-2,4,0,0,0,0,-1,0,0,0,0,0,0,0,0,0;
        -1,0,0,0,0,4,-1,0,0,0,-1,0,0,0,0,0,0,0,0;
        0,-1,0,0,0,-1,4,-1,0,0,0,-1,0,0,0,0,0,0,0;
        0,0,-1,0,0,0,-1,4,-1,0,0,0,-1,0,0,0,0,0,0;
        0,0,0,-1,0,0,0,-1,4,-1,0,0,0,-1,0,0,0,0,0;
        0,0,0,0,-1,0,0,0,-2,4,0,0,0,0,-1,0,0,0,0;
        0,0,0,0,0,0,-1,0,0,0,0,4,-1,0,0,0,-1,0,0,0;
        0,0,0,0,0,0,0,-1,0,0,0,-1,4,-1,0,0,0,-1,0,0;
        0,0,0,0,0,0,0,0,-1,0,0,0,-1,4,-1,0,0,0,0,0;
        0,0,0,0,0,0,0,0,0,-1,0,0,0,-1,4,-1,0,0,0,0;
        0,0,0,0,0,0,0,0,0,0,-1,0,0,0,-2,4,0,0,0,0;
        0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,4,-1,-1,0;
        0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,-1,4,0,-1;
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-2,0,4,-1;
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-2,-1,4];
n = 19;
%}

% New S and c are as below, by assigning value at A and b respectively:
 b = [0;0;0;0;0;0;0;-110;-110;-110;0;-220;330;110;330;-110;220;-110;330];
 A = [18,-8,1,0,0,-8,2,0,0,0,1,0,0,0,0,0,0,0,0;
     -8,19,-8,1,0,2,-8,2,0,0,0,1,0,0,0,0,0,0,0;
     1,-8,19,-8,1,0,2,-8,2,0,0,0,1,0,0,0,0,0,0;
     0,1,-8,22,-12,0,0,2,-8,3,0,0,0,1,0,0,0,0,0;
     0,0,1,-12,18,0,0,0,3,-8,0,0,0,0,1,0,0,0,0;
     -8,2,0,0,0,19,-8,1,0,0,-8,2,0,0,0,1,0,0,0;
     2,-8,2,0,0,-8,20,-8,1,0,2,-8,2,0,0,0,1,0,0;
     0,2,-8,2,0,1,-8,20,-8,1,0,2,-8,2,0,0,0,0,0;
     0,0,2,-8,3,0,1,-8,23,-12,0,0,2,-8,3,0,0,0,0;
     0,0,0,3,-8,0,0,1,-12,19,0,0,0,3,-8,0,0,0,0;
     1,0,0,0,0,-8,2,0,0,0,19,-8,1,0,0,-8,2,1,0;
     0,1,0,0,0,2,-8,2,0,0,-8,20,-8,1,0,2,-8,0,1;
     0,0,1,0,0,0,2,-8,2,0,1,-8,19,-8,1,0,1,0,0;
     0,0,0,1,0,0,0,2,-8,3,0,1,-8,22,-12,0,0,0,0;
     0,0,0,0,1,0,0,0,3,-8,0,0,1,-12,18,0,0,0,0;
     0,0,0,0,0,1,0,0,0,0,-8,2,0,0,0,22,-8,-12,3;
     0,0,0,0,0,0,1,0,0,0,2,-8,1,0,0,-8,22,3,-12;
     0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,-12,3,18,-8;
     0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,3,-12,-8,18];
n = 19;
%}
%% --- Long Method --- %%
%{
for i=1:n
    xk(i,1) = 0;
end
xk1 = xk;
rk = xk;
rk1 = xk;
var2 = xk;
var1 = xk;
var3 = xk;
var4 = xk;
var5 = xk;
pk1 = xk;
for i=1:n
    for j=1:n
        var1(i,1) = var1(i,1) + (A(i,j)*xk(j,1));
    end
```

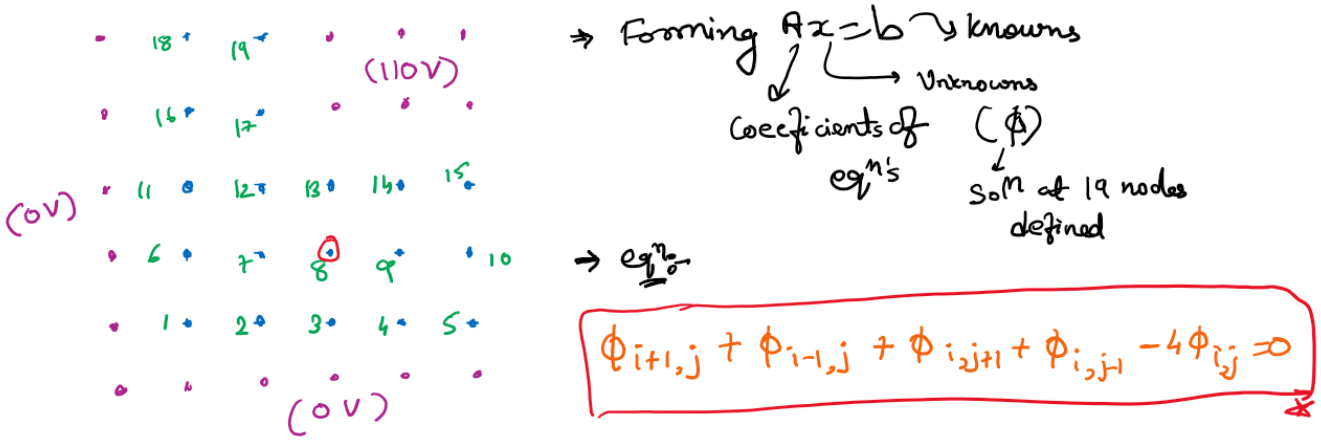
```
end
for i=1:n
    rk(i,1) = b(i,1) - var1(i,1);
end

pk = rk;
eps = 1e-5;
N = 0;
norm_inf = [];
iteration = [];
norm_2 = [];

%while true
for k=1:19
    num1=0;
    deno1=0;
    for i=1:n
        num1 = num1 + (pk(i,1)*rk(i,1));
    end
    for i=1:n
        for j=1:n
            var2(i,1) = var2(i,1) + (A(i,j)*pk(j,1));
        end
    end
    for i=1:n
        deno1 = deno1 + (pk(i,1)*var2(i,1));
    end
    alpha = num1/deno1;
    for i=1:n
        xk1(i,1) = xk(i,1) + (alpha*pk(i,1));
    end
    for i=1:n
        for j=1:n
            var3(i,1) = var3(i,1) + (A(i,j)*xk1(j,1));
        end
    end
    for i=1:n
        rk1(i,1) = b(i,1) - var3(i,1);
    end
    num2 = 0;
    deno2 = 0;
    for i=1:n
        for j=1:n
            var4(i,1) = var4(i,1) + (A(i,j)*rk1(j,1));
        end
    end
    for i=1:n
        num2 = num2 + (pk(i,1)*var4(i,1));
    end
    for i=1:n
        var5(i,1) = var5(i,1) + (A(i,j)*pk(j,1));
    end
    for i=1:n
        deno2 = deno2 + (pk(i,1)*var5(i,1));
    end
    beta = (-1)*(num2/deno2);
    for i=1:n
        pk1(i,1) = rk1(i,1) + (beta*pk(i,1));
    end
```

```
    limit = max(abs(rk1));
    two_norm = 0;
    for i=1:n
        two_norm = two_norm + (rk1(i,1)*rk1(i,1));
    end
    if two_norm<eps
        break
    end
    norm_inf = [norm_inf; limit];
    norm_2 = [norm_2; sqrt(two_norm)];

    rk = rk1;
    pk = pk1;
    xk = xk1;
    N = N + 1;
    iteration = [iteration; N];
end
%}
%% ----- Direct Method ----- %%

xk = zeros(n,1);
rk = b - A*xk;
pk = rk;
eps = 1e-5;
N = 0;
norm_inf = [];
iteration = [];
norm_2 = [];

while true

    num1 = pk'*rk;
    deno1 = pk'*A*pk;
    alpha = num1/deno1;
    xk1 = xk + alpha*pk;
    rk1 = b - A*xk1;
    num2 = pk'*A*rk1;
    deno2 = pk'*A*pk;
    beta = (-1)*(num2/deno2);
    pk1 = rk1 + beta*pk;

    limit = max(abs(rk1));
    two_norm = 0;
    for i=1:n
        two_norm = two_norm + (rk1(i,1)*rk1(i,1));
    end
    if two_norm<eps
        break
    end
    norm_inf = [norm_inf; limit];
    norm_2 = [norm_2; sqrt(two_norm)];

    rk = rk1;
    pk = pk1;
    xk = xk1;
    N = N + 1;
    iteration = [iteration; N];
end

%% --- Plot --- %%
```

```
f1 = figure;
f2 = figure;

figure(f1);
plot(iteration,norm_inf)
title('Plot of infinite norm residual vs Number of iterations')
xlabel('Number of iterations')
ylabel('Infinite norm')
hold on

figure(f2);
plot(iteration,norm_2)
title('Plot of two norm residual vs Number of iterations')
xlabel('Number of iterations')
ylabel('Two Norm')

aa = 'The Potential at the point (0.06,0.04) is: ';
bb = xk1; cc = ' V';
nn = [1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19];
s = ['  ';'  ';'  ';'  ';'  ';'  ';'  ';'  ';'  ';'  ';
     '  ';'  ';'  ';'  ';'  ';'  ';'  ';'  ';'  ';'  '];
disp('-----------------------------------------');
disp('  |  Node  | Voltage|')
disp('-----------------------------------------');
disp([s s num2str(nn) s s s num2str(xk1)])
disp('-------------------------------------------------------');
disp([aa num2str(xk1(8)) cc])
disp('-------------------------------------------------------');

%% --- Code Ends --- %%
```

C. Node numbering of the unknown

The potential at the point (0.06,0.04) point is at the node 8 in this plot. The equation is applied at all the nodes, shown in the figure below, and at nodes 5, 10, 15, 18 and 19 we applied the plane of symmetry. And thus matrix A and b is formed using the set of equations, which is provided below. As the A is non-symmetric and not PD, it is converted to Sx = c form and thus S and c is as provided below.



```
b = [0 0 0 0 0 0 0 0 0 0 0 0 -110 -110 -110 0 -110 0 -110]';

A = [-4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
     1 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
     0 1 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
     0 0 1 -4 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
     0 0 0 2 -4 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
     1 0 0 0 0 -4 1 0 0 0 1 0 0 0 0 0 0 0 0;
     0 1 0 0 0 1 -4 1 0 0 0 1 0 0 0 0 0 0 0;
     0 0 1 0 0 0 1 -4 1 0 0 0 1 0 0 0 0 0 0;
     0 0 0 1 0 0 0 1 -4 1 0 0 0 1 0 0 0 0 0;
     0 0 0 0 1 0 0 0 2 -4 0 0 0 0 1 0 0 0 0;
     0 0 0 0 0 1 0 0 0 0 -4 1 0 0 0 1 0 0 0;
     0 0 0 0 0 0 1 0 0 0 1 -4 1 0 0 0 1 0 0;
     0 0 0 0 0 0 0 1 0 0 0 1 -4 1 0 0 0 0 0;
     0 0 0 0 0 0 0 0 1 0 0 0 1 -4 1 0 0 0 0;
     0 0 0 0 0 0 0 0 0 1 0 0 0 2 -4 0 0 0 0;
     0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 -4 1 1 0;
     0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 -4 0 1;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 -4 1;
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 -4];

S = [18,-8,1,0,0,-8,2,0,0,0,1,0,0,0,0,0,0,0,0;
     -8,19,-8,1,0,2,-8,2,0,0,0,1,0,0,0,0,0,0,0;
     1,-8,19,-8,1,0,2,-8,2,0,0,0,1,0,0,0,0,0,0;
     0,1,-8,22,-12,0,0,2,-8,3,0,0,0,1,0,0,0,0,0;
     0,0,1,-12,18,0,0,0,3,-8,0,0,0,0,1,0,0,0,0;
     -8,2,0,0,0,19,-8,1,0,0,-8,2,0,0,0,1,0,0,0;
     2,-8,2,0,0,-8,20,-8,1,0,2,-8,2,0,0,0,1,0,0;
     0,2,-8,2,0,1,-8,20,-8,1,0,2,-8,2,0,0,0,0,0;
     0,0,2,-8,3,0,1,-8,23,-12,0,0,2,-8,3,0,0,0,0;
     0,0,0,3,-8,0,0,1,-12,19,0,0,0,3,-8,0,0,0,0;
     1,0,0,0,0,-8,2,0,0,0,19,-8,1,0,0,-8,2,1,0;
     0,1,0,0,0,2,-8,2,0,0,-8,20,-8,1,0,2,-8,0,1;
     0,0,1,0,0,0,2,-8,2,0,1,-8,19,-8,1,0,1,0,0;
     0,0,0,1,0,0,0,2,-8,3,0,1,-8,22,-12,0,0,0,0;
     0,0,0,0,1,0,0,0,3,-8,0,0,1,-12,18,0,0,0,0;
     0,0,0,0,0,1,0,0,0,0,-8,2,0,0,0,22,-8,-12,3;
     0,0,0,0,0,0,1,0,0,0,2,-8,1,0,0,-8,22,3,-12;
     0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,-12,3,18,-8;
     0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,3,-12,-8,18];

 c = [0;0;0;0;0;0;0;-110;-110;-110;0;-220;330;110;330;-110;220;-110;330];
```

D. Capacitance Calculation by CG

```
%% --- Capacitance Calculations --- %%

Potential = [0;0;0;0;0;0;23.2568674801173;22.2643057533745;
    19.1106843503746;13.6519290102836;7.01855435376675;0;0;
    14.4222883912879;28.4784773616273;40.5265026047826;46.6896712215120;
    48.4988583816628;77.3592236491626;75.4690180911891;67.8271775336894;
    45.3131894026422;22.1921218710458;0;0;29.0330096687591;62.7549808782839;
    110;110;110;110;66.6737244214342;31.1849359428342;0];

vertex = [1,2,12;2,11,12;2,3,11;3,10,11;3,4,10;4,9,10;4,5,9;5,8,9;5,6,8;
    6,7,8;8,7,17;7,18,17;9,8,16;8,17,16;10,9,15;9,16,15;11,10,14;10,15,14;
    12,11,13;11,14,13;13,14,24;14,23,24;14,15,23;15,22,23;15,16,22;16,21,22;
    16,17,21;17,20,21;17,18,20;18,19,20;20,19,29;19,30,29;21,20,28;20,29,28;
    22,21,27;21,28,27;23,22,26;22,27,26;24,23,25;23,26,25;25,26,34;26,33,34;
```

```
    26,27,33;27,32,33;27,28,32;28,31,32];
Sa = [1 -0.5 -0.5;
     -0.5 0.5 0;
     -0.5 0 0.5];
Sb = [0.5 -0.5 0;
     -0.5 1 -0.5;
      0 -0.5 0.5];
E = zeros(46,1);
U = zeros(3,1);
V = 110;
Energy = 0;
eps = 8.85e-12;

for i=1:46
    if mod(i,2)~=0
        S = Sa;
    else
        S = Sb;
    end
    verte = vertex(i,:);
    U(1,1) = Potential(verte(1));
    U(2,1) = Potential(verte(2));
    U(3,1) = Potential(verte(3));
    E(i) = (U'*S*U);
    Energy = Energy + (E(i)/2);
end
Ener = 4*eps*Energy;
C = (2*Ener)/(V^2);
disp('The capacitance is: ')
disp(C)

%% --- Code Ends --- %%
```