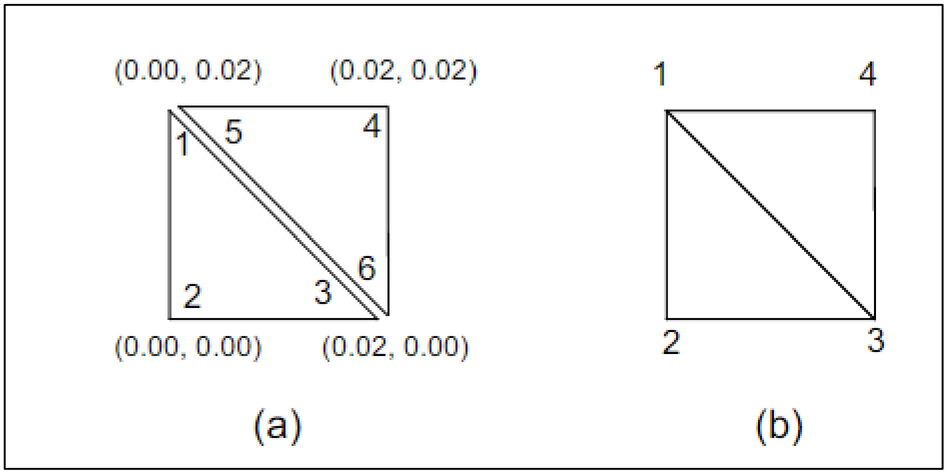
ECSE 543 ASSIGNMENT 2

RAZI MURSHED

260516333

Razi Murshed
ID - 26051633

QUESTION 1



For each of the triangles shown above we can find the energy as shown below –

Energy for 1 Triangle –

1. $W^{(e)} = \frac{1}{2}\int_{\Delta e} |\nabla U|^2 dS$

$$= \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} U_i U_j \int_{\Delta} \nabla \alpha_i \cdot \nabla \alpha_j \, dS$$

$$S = \int \nabla \alpha_i \cdot \nabla \alpha_j \, dS$$

General form of S –

$$S_{i,j} = \frac{1}{4A}\left[(y_j - y_{j+1})(y_{i+2} - y_i) + (x_{j+1} - x_j)(x_i - x_{i+2})\right]$$

Area of Triangle – $A = \frac{1}{2}bh$

$$A = \frac{1}{2}(0.02)^2 = 2\,cm^2$$

For triangle 1,2,3

Node 1: (0.00, 0.02)
Node 2: (0.00, 0.00)
Node 3: (0.02, 0.00)

Razi Murshed
ID - 26051633

All alphas -

$\alpha_1(x_1,y_1) = \frac{1}{2A}\left[(x_2y_3 - x_3y_2) + x(y_2-y_3) + y(x_3-x_2)\right]$

$\alpha_1(x_1,y_1) = 0.25(2y)$

$\alpha_2(x_2,y_2) = \frac{1}{2A}\left((x_3y_1 - x_1y_3) + x(y_3-y_1) + y(x_1-x_3)\right)$

$\alpha_2(x_2,y_2) = 0.25(4-2x-2y)$

$\alpha_3(x_3,y_3) = \frac{1}{2A}\left((x_1y_2 - x_2y_1) + x(y_1-y_2) + y(x_2-x_1)\right)$

$\alpha_3(x_3,y_3) = 0.25(-4+2x)$

$S_{i,j} = \int -\nabla\alpha_i \cdot \nabla\alpha_j \, ds$ — (I)

$S_{i,j} = \frac{1}{4A}\nabla(\alpha_i \cdot \alpha_j)$ — (II)

Therefore

$$S^{(1,2,3)} = \begin{bmatrix} 0.5 & -0.5 & 0 \\ -0.5 & 1 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$$

Razi Murshed
ID - 26051633

For Triangle 4, 5, 6

Node 4: (0.02, 0.02)
Node 5: (0.00, 0.02)
Node 6: (0.02, 0.00)

All alphas —

$$\alpha_4(x_4, x_5) = \frac{1}{2A}\left[(x_6 x_5 - x_5 y_6) + x(y_5 - x_5) + y(x_6 - x_5)\right]$$

$$\alpha_4(x_4, y_4) = 0.25\,(4 + 2x + 2y)$$

$$\alpha_5(x_5, y_5) = \frac{1}{2A}\left((x_4 y_6 - x_6 y_4) + x(y_6 - y_4) + y(x_4 - x_5)\right)$$

$$\alpha_5(x_5, y_5) = 0.25\,(-4 - 2x)$$

$$\alpha_6(x_6, y_6) = \frac{1}{2A}\left((x_5 y_4 - x_4 y_5) + x(y_4 - y_5) + y(x_5 - x_4)\right)$$

$$\alpha_6(x_6, y_6) = 0.25\,(-4 - 2y)$$

Using (I) and (II)

$$S^{(4,5,6)} = \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

Razi Murshed
ID - 26051633

To find global Matrix S, we use

$$\boxed{S = C^T S_{com} C} \quad (11)$$

C is a disjoint matrix and S_{com} is combination of local S-matrices. Therefore S_{com}

$$S_{com} = \begin{bmatrix} S^{(1,2,3)} & 0 \\ 0 & S^{(4,5,6)} \end{bmatrix}$$

When combined, the two triangles form a rectangle of 4 nodes. Hence C is a 6 by 4 matrix such that

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$S_{com} = \begin{bmatrix} 0.5 & -0.5 & 0 & & & \\ -0.5 & 1 & -0.5 & & & \\ 0 & -0.5 & 0.5 & & & \\ & & & 1 & -0.5 & 0.5 \\ & & & -0.5 & 0.5 & 0 \\ & & & -0.5 & 0 & 0.5 \end{bmatrix}$$

Using (11)

$$S = \begin{bmatrix} 1 & -0.5 & 0 & -0.5 \\ 0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & -0.5 \\ 0.5 & 0 & -0.5 & 1 \end{bmatrix}$$

Razi Murshed
ID - 26051633

QUESTION 2

PART A

The mesh was created according to the following diagram –

It was specified in an input file called “meshGen.txt”. This was created by hand and then converted to a .dat file for input to the SIMPLE2D program in order to give us the output file, “Simple2DOutput.txt”.

PART B

From the file “Simple2DOutput.txt” we can see that the potential at 0.06,0.04 is 40.5265V.

PART C

We can find the capacitance by using the values found for energy in a capacitor and energy at a potential equation. After rearranging these equations, we come up with -

$$C = \frac{\varepsilon_0 |\nabla U|^2}{V^2}$$

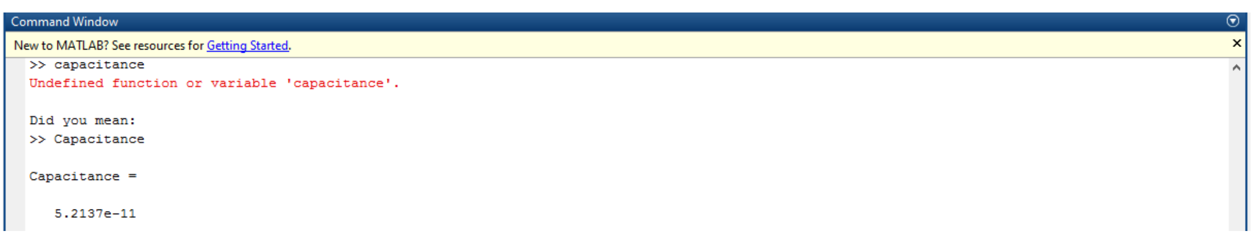
Then we find the energies of each element by using the equations as shown in question 1 and computing their sum. It is possible to find the energy of a square when given the potential at corners of the square. We find $|\nabla U|^2$ of the square to be –

$$|\nabla U|^2 = U^T S U$$

S is computed in question 1 and U is a vector of the 4 potentials at the corners of the square. If we assume these potentials at corner n to be Pn, the following equation can be derived –

$$|\nabla U|^2 = p1^2 - p1p2 - p2p4 + p2^2 - p2p3 + p3^2 - p3p4 + p4^2$$

Then we substitute $|\nabla U|^2$ in the capacitance equation as shown before and multiply the answer by 4 to find the potential of the Coaxial cable which comes out to be $5.21 \times 10^{-11}\, F$ -



QUESTION 3

The code for this question can be found in the Appendix in the “conjugateGradient.m” file. The Cholesky Decomposition code can be found under “choleskyDecompose.m”. The matrix A was constructed as shown below –

Razi Murshed
ID - 26051633

Editor - conjugateGradient.m | Variables - A

Capacitance | A | B

19x19 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-4	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	
2	1	-4	1	0	2	0	0	0	0	0	0	0	0	0	0	0	
3	0	1	-4	0	0	2	0	0	0	0	0	0	0	0	0	0	
4	1	0	0	-4	1	0	1	0	0	0	0	0	0	0	0	0	
5	0	1	0	1	-4	1	0	1	0	0	0	0	0	0	0	0	
6	0	0	1	0	1	-4	0	0	1	0	0	0	0	0	0	0	
7	0	0	0	1	0	0	-4	1	0	0	0	1	0	0	0	0	
8	0	0	0	0	1	0	1	-4	1	0	0	0	1	0	0	0	
9	0	0	0	0	0	1	0	1	-4	0	0	0	0	1	0	0	
10	0	0	0	0	0	0	0	0	0	-4	2	0	0	0	1	0	
11	0	0	0	0	0	0	0	0	0	1	-4	1	0	0	0	1	
12	0	0	0	0	0	0	1	0	0	0	1	-4	1	0	0	0	
13	0	0	0	0	0	0	0	1	0	0	0	1	-4	1	0	0	
14	0	0	0	0	0	0	0	0	1	0	0	0	1	-4	0	0	
15	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-4	2	
16	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	-4	
17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	
18	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	

The vector B was constructed as shown below –

Editor - conjugateGradient.m

Capacitance | A | B

19x1 double

	1
1	-110
2	0
3	0
4	-110
5	0
6	0
7	-110
8	0
9	0
10	-110
11	-110
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0

PART A

The matrix A was tested using the "choleskyDecompose.m" function. It failed seeing as A is not positive definite as the values it gave for the decomposed matrix were complex.

Razi Murshed
ID - 26051633

To solve this issue we can multiply A by A^T in order to get an SPD matrix. To preserve the integrity of the equation $Ax = B$ we have to multiply both sides by A^T to get $A^T Ax = A^T B$. This fix works as we can see the decomposed matrix A below –

4.582576	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1.74574	4.353433	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.218218	-1.75012	4.229592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-2.61861	-0.36096	-0.01426	3.465885	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.654654	-2.49393	-0.35643	-2.0748	2.986626	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.689111	-2.55201	0.349798	-2.16473	2.684013	0	0	0	0	0	0	0	0	0	0	0	0	0
0.436436	0.175012	0.0499	-1.96004	-0.63555	-0.25463	3.803408	0	0	0	0	0	0	0	0	0	0	0	0
0	0.459408	0.190094	0.625681	-1.83764	-0.75571	-2.16224	3.276894	0	0	0	0	0	0	0	0	0	0	0
0	0	0.472859	0.001945	0.727435	-1.94456	0.24909	-2.34529	2.983868	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4.582576	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.262922	0.173487	0.114411	-2.61861	3.468507	0	0	0	0	0	0	0	0
0	0	0	0.288527	0.200438	0.124057	-1.91289	-0.56595	-0.25335	0.436436	-1.79531	3.521924	0	0	0	0	0	0	0
0	0	0	0	0.334826	0.270047	0.599872	-1.79547	-0.69667	0	0.355622	-2.1316	3.328046	0	0	0	0	0	0
0	0	0	0	0	0.372576	0.024943	0.712715	-1.88017	0	0.024479	0.276115	-2.27338	3.093939	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-1.74574	-0.45306	-0.01461	0.039052	0.033583	4.212393	0	0	0	0
0	0	0	0	0	0	0	0	0	0.654654	-1.81222	-0.43704	-0.08627	-0.01005	-2.77298	2.720171	0	0	0
0	0	0	0	0	0	0.262922	0.173487	0.114411	0	0.544235	-1.81515	-0.54964	-0.21874	0.533866	-2.34406	3.027733	0	0
0	0	0	0	0	0	0	0.305167	0.239858	0	-0.02318	0.62235	-1.78788	-0.64717	0.021401	0.414895	-2.3454	3.028635	0
0	0	0	0	0	0	0	0	0.335135	0	-0.01105	0.018473	0.684121	-1.88092	0.007529	0.018026	0.33161	-2.41567	2.817009

PART B

We proceed to solve the matrix equation by first using the cholesky decomposition method as was done in the previous assignment. Since the assignment was coded in MATLAB instead of Java this time, we rewrote the forward and back substitution parts as helper methods.

```
Command Window
New to MATLAB? See resources for Getting Started.

>> conjugateGradient

XCholesky =

   77.3592
   48.4989
   23.2569
   75.4690
   46.6897
   22.2643
   67.8272
   40.5265
   19.1107
   66.6737
   62.7550
   45.3132
   28.4785
   13.6519
   31.1849
   29.0330
   22.1921
   14.4223
    7.0186

>>
```

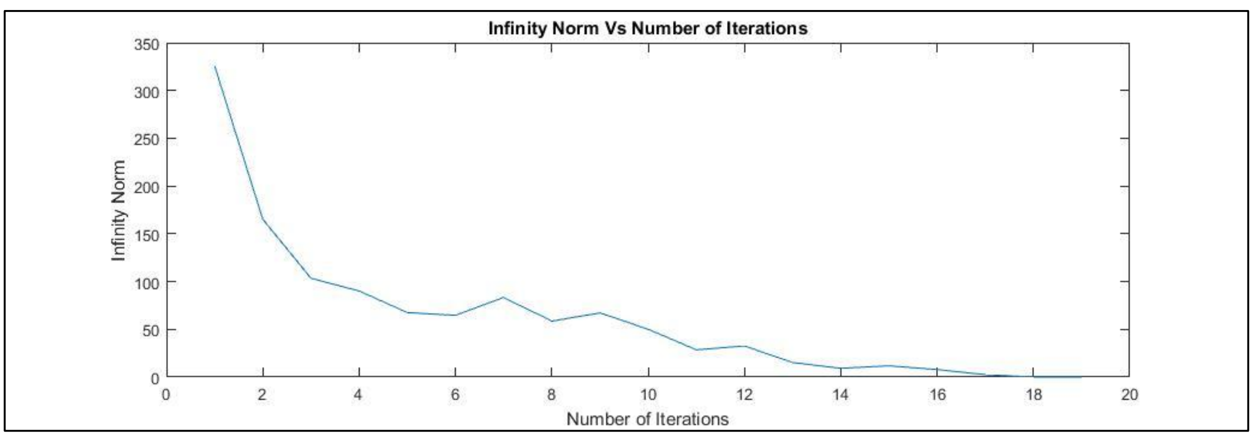
Then we move on to solve again but this time with the conjugate gradient method. This yields the following results –

Razi Murshed
ID - 26051633

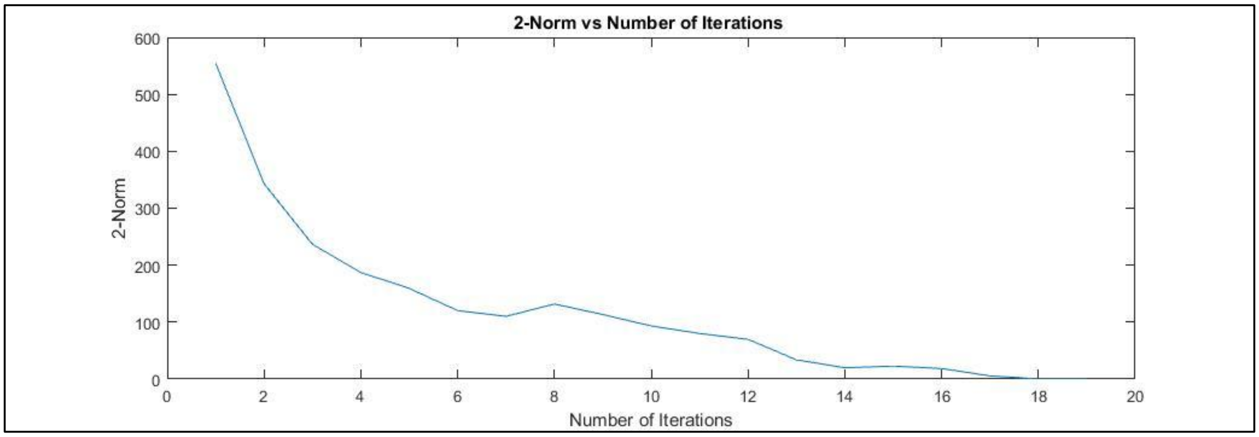
```
Command Window
New to MATLAB? See resources for Getting Started.
>> conjugateGradient
>> conjugateGradient
>> X

X =

   77.3592
   48.4989
   23.2569
   75.4690
   46.6897
   22.2643
   67.8272
   40.5265
   19.1107
   66.6737
   62.7550
   45.3132
   28.4785
   13.6519
   31.1849
   29.0330
   22.1921
   14.4223
    7.0186

fx >> 
```

PART C

The following graphs display inf-norm and two norm for the conjugate program –



Razi Murshed
ID - 26051633



PART D

The results are summarized in the following table –

Solving Method	Result(V)
Cholesky Decomposition	40.5265
Conjugate Gradient	40.5265
Simple 2D program	40.5265
IterateSOR	40.5265

As we can see the values are all equal given that every method approximates a solution to Ax=b very closely. The true solution remains same regardless of the method used given an identical mesh.

PART E

We must find the potential at every node and then we can use the Simple 2D program and in conjunction with the methods used in question 2 we can find the capacitance per unit length.

APPENDIX

MESHGEN.TXT

1 0.04 0.0

2 0.06 0.0

3 0.08 0.0

4 0.1 0.0

5 0.0 0.02

6 0.02 0.02

7 0.04 0.02

8 0.06 0.02

Razi Murshed
ID - 26051633

9 0.08 0.02

10 0.1 0.02

11 0.0 0.04

12 0.02 0.04

13 0.04 0.04

14 0.06 0.04

15 0.08 0.04

16 0.1 0.04

17 0.0 0.06

18 0.02 0.06

19 0.04 0.06

20 0.06 0.06

21 0.08 0.06

22 0.1 0.06

23 0.0 0.08

24 0.02 0.08

25 0.04 0.08

26 0.06 0.08

27 0.08 0.08

28 0.1 0.08

29 0.0 0.1

30 0.02 0.1

31 0.04 0.1

32 0.06 0.1

33 0.08 0.1

34 0.1 0.1

1 2 7 0.000

2 8 7 0.000

Razi Murshed
ID - 26051633

2 3 8 0.000

3 9 8 0.000

3 4 9 0.000

4 10 9 0.000

5 6 11 0.000

6 12 11 0.000

6 7 12 0.000

7 13 12 0.000

7 8 13 0.000

8 14 13 0.000

8 9 14 0.000

9 15 14 0.000

9 10 15 0.000

10 16 15 0.000

11 12 17 0.000

12 18 17 0.000

12 13 18 0.000

13 19 18 0.000

13 14 19 0.000

14 20 19 0.000

14 15 20 0.000

15 21 20 0.000

15 16 21 0.000

16 22 21 0.000

17 18 23 0.000

18 24 23 0.000

18 19 24 0.000

19 25 24 0.000

19 20 25 0.000

Razi Murshed
ID - 26051633

20 26 25 0.000

20 21 26 0.000

21 27 26 0.000

21 22 27 0.000

22 28 27 0.000

23 24 29 0.000

24 30 29 0.000

24 25 30 0.000

25 31 30 0.000

25 26 31 0.000

26 32 31 0.000

26 27 32 0.000

27 33 32 0.000

27 28 33 0.000

28 34 33 0.000

1 110.0

5 110.0

6 110.0

7 110.0

29 0.000

30 0.000

31 0.000

32 0.000

33 0.000

34 0.000

10 0.000

16 0.000

22 0.000

Razi Murshed
ID - 26051633

28 0.000

4 0.000

SIMPLE2DOUTPUT.TXT

1.0000	0.0400	0	110.0000
2.0000	0.0600	0	66.6737
3.0000	0.0800	0	31.1849
4.0000	0.1000	0	0
5.0000	0	0.0200	110.0000
6.0000	0.0200	0.0200	110.0000
7.0000	0.0400	0.0200	110.0000
8.0000	0.0600	0.0200	62.7550
9.0000	0.0800	0.0200	29.0330
10.0000	0.1000	0.0200	0
11.0000	0	0.0400	77.3592
12.0000	0.0200	0.0400	75.4690
13.0000	0.0400	0.0400	67.8272
14.0000	0.0600	0.0400	45.3132
15.0000	0.0800	0.0400	22.1921
16.0000	0.1000	0.0400	0
17.0000	0	0.0600	48.4989
18.0000	0.0200	0.0600	46.6897
19.0000	0.0400	0.0600	40.5265
20.0000	0.0600	0.0600	28.4785
21.0000	0.0800	0.0600	14.4223
22.0000	0.1000	0.0600	0
23.0000	0	0.0800	23.2569

Razi Murshed
ID - 26051633

24.0000	0.0200	0.0800	22.2643
25.0000	0.0400	0.0800	19.1107
26.0000	0.0600	0.0800	13.6519
27.0000	0.0800	0.0800	7.0186
28.0000	0.1000	0.0800	0
29.0000	0	0.1000	0
30.0000	0.0200	0.1000	0
31.0000	0.0400	0.1000	0
32.0000	0.0600	0.1000	0
33.0000	0.0800	0.1000	0
34.0000	0.1000	0.1000	0

CHOLESKYDECOMPOSE.M

```
function L = choleskyDecompose(M);
% Cholesky Decompose a Symmetric Positive Definite m by m matrix
n = length( M );
L = zeros( n, n );
for i=1:n
   L(i, i) = sqrt(M(i, i) - L(i, :)*L(i, :)');
   for j=(i + 1):n
      L(j, i) = (M(j, i) - L(i,:)*L(j ,:)')/L(i, i);
   end
end
end
```

FORWARDSUBSTITUTION.M

```
function x=forwardSubstitution(L,b,n)

x=zeros(n,1);
for j=1:n
    if (L(j,j)==0) error('Matrix is singular!'); end;
    x(j)=b(j)/L(j,j);
    b(j+1:n)=b(j+1:n)-L(j+1:n,j)*x(j);
end
```

BACKSUBSTITUTION.M

Razi Murshed
ID - 26051633

```
function x=backSubstitution(U,b,n)

x=zeros(n,1);
for j=n:-1:1
    if (U(j,j)==0) error('Matrix is singular!'); end;
    x(j)=b(j)/U(j,j);
    b(1:j-1)=b(1:j-1)-U(1:j-1,j)*x(j);
end
```

Q2CAPACITANCE.M

```
nodesWide = 6;
nodesHigh = 6;

mesh = zeros(nodesWide, nodesHigh);
input = zeros(4, 34);
% read mesh values
file = fopen('Simple2DOutput.txt', 'r');
input = fscanf(file, '%f', size(input));
input = input';
fclose(file);


for x = 1: size(input, 1)
    xNode = cast((input(x,2) / 0.02), 'int8');
    yNode = cast((input(x,3) / 0.02), 'int8');
    mesh(yNode + 1,xNode + 1) = input(x,4);
end

% % voltage is 10 volts on inner conductor
mesh(1,1) = 110.0;
mesh(1,2) = 110.0;

totalEnergy = 0.0;
for y  = 1 : (nodesHigh - 1);
    for x = 1: (nodesWide - 1);
        u1 = mesh(y+1,x);
        u2 = mesh(y,x);
        u3 = mesh(y,x+1);
        u4 = mesh(y+1,x+1);

        totalEnergy = totalEnergy + (u1*u1 - u1*u2);
        totalEnergy = totalEnergy + (-u1*u4 + u2*u2);
        totalEnergy = totalEnergy + (-u2*u3 + u3*u3);
        totalEnergy = totalEnergy + (-u3*u4 + u4*u4);
    end
end

%
epsilon = 8.854187817620e-12;
voltageSquared = 12100;
Capacitance = totalEnergy*(epsilon * 4 / voltageSquared);
%
```

Razi Murshed
ID - 26051633

CONJUGATEGRADIENT.M

```
CABLE_HEIGHT = 0.1;
CABLE_WIDTH = 0.1;
CORE_HEIGHT = 0.02;
CORE_WIDTH = 0.04;
CORE_POTENTIAL = 110;
MIN_RESIDUAL = 0.0001;

height  = 0.02;

nodesWide = int8(CABLE_WIDTH/height) + 1;
nodesHigh = int8(CABLE_HEIGHT/height) + 1;
nodeNumber = 19;

mesh = zeros(nodesWide,nodesHigh);

[rows, columns] = size(mesh);
for i = 1:rows
 for j = 1:columns
    if(( j <= int16(CORE_WIDTH/height)+1)&&(i <=
int16(CORE_HEIGHT/height)+1))
       mesh(i,j) = CORE_POTENTIAL;
    else
       mesh(i,j) = 0;
    end
 end
end

%Constructing A and B matrices
B = zeros([nodeNumber 1]);
B(1) = -110;
B(4) = -110;
B(7) = -110;
B(10) = -110;
B(11) = -110;

A = zeros(nodeNumber) - [[4 -1 0 -2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [- 1 4 -1 0 -2 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
     [0 -1 4 0 0 -2 0 0 0 0 0 0 0 0 0 0 0 0]
     [- 1 0 0 4 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0]
     [0 -1 0 -1 4 -1 0 -1 0 0 0 0 0 0 0 0 0 0]
     [0 0 -1 0 -1 4 0 0 -1 0 0 0 0 0 0 0 0 0]
     [0 0 0 -1 0 0 4 -1 0 0 0 -1 0 0 0 0 0 0]
     [0 0 0 0 -1 0 -1 4 -1 0 0 0 -1 0 0 0 0 0]
     [0 0 0 0 0 -1 0 -1 4 0 0 0 0 -1 0 0 0 0]
     [0 0 0 0 0 0 0 0 0 4 -2 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 -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 0 0 0 0 -1]
     [0 0 0 0 0 0 0 0 0 -1 0 0 0 0 4 -2 0 0 0]
```

Razi Murshed
ID - 26051633

```
     [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 -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 -1 4]
     ];
 %Cannot do this since it gives compleX values
 %decomposedA = choleskYDecompose(A*A');

%Solving using Cholesky
%Solution is to multiplY bY transpose
newA = A' * A;
U = choleskyDecompose(newA);
B = A'*B;
Ut = U';
%Solve the equation
Y = zeros([size(B) 1]);
X = zeros([size(B) 1]);
% Acuiring D vector
n = length(U);
% Now use a vector y to solve 'Ly=b'
Y=forwardSubstitution(U,B,n);
X=backSubstitution(Ut,Y, n);
XCholesky = X;

%Solving using conjugate gradient
X = zeros([nodeNumber 1]);
R = B-(newA*X);
P = R;
infNormVec = [];
twoNormVec = [];
iterations = [];
results = zeros(2, nodeNumber);
for i = 1:nodeNumber
    temp = (P'*R);
    temp1 = (P'*newA*P);
    alpha = temp/temp1;
    X = X + alpha*P;
    R = B - newA*X;
    temp = (P'*newA*R);
    beta = ((-1)*temp)/(temp1);
    P = R + beta*P;

   results(1,1) = X(1);
   results(2,1)= R(1);
    for y  = 1 : nodeNumber - 1
        results(1,y+1) = X(y+1);
        results(2,y+1)= R(y+1);
    end
    infNorm = 0;
    twoNorm = 0;
    for y  = 1 : nodeNumber
        val = abs(R(y,1));
        if val > infNorm
            infNorm = val;
        end
        twoNorm = twoNorm + (R(y,1))^2;
```

Razi Murshed
ID - 26051633

```
    end
    twoNorm = sqrt(twoNorm);
    infNormVec = [infNormVec, infNorm];
    twoNormVec = [twoNormVec, twoNorm];
    iterations = [iterations,i];
end

figure
plot(iterations, infNormVec);
figure
plot(iterations, twoNormVec);
```