UNIVERSITY
OF WOLLONGONG
IN DUBAI

# Assignment Cover Sheet

| Student Name | Student number |
|---|---|
| Taha Yaseen Parker | 8243578 |

| | |
|---|---|
| **Subject code and name** | ECTE213 – Engineering Electromagnetics |
| **Lab Instructor** | Mr. Mahmoud Alkakuri |
| **Title of Assignment** | Lab 2 |
| **Date and time due** | 28 January 2025, 23.55 |
| **Lab Number** | 2 |

### Student declaration and acknowledgment

By submitting this assignment online, the submitting student declares on behalf of the team that:

1. All team members have read the subject outline for this subject, and this assessment item meets the requirements of the subject detailed therein.
2. This assessment is entirely our work, except where we have included fully documented references to the work of others. The material in this assessment item has yet to be submitted for assessment.
3. Acknowledgement of source information is by the guidelines or referencing style specified in the subject outline.
4. All team members know the late submission policy and penalty.
5. The submitting student undertakes to communicate all feedback with the other team members.

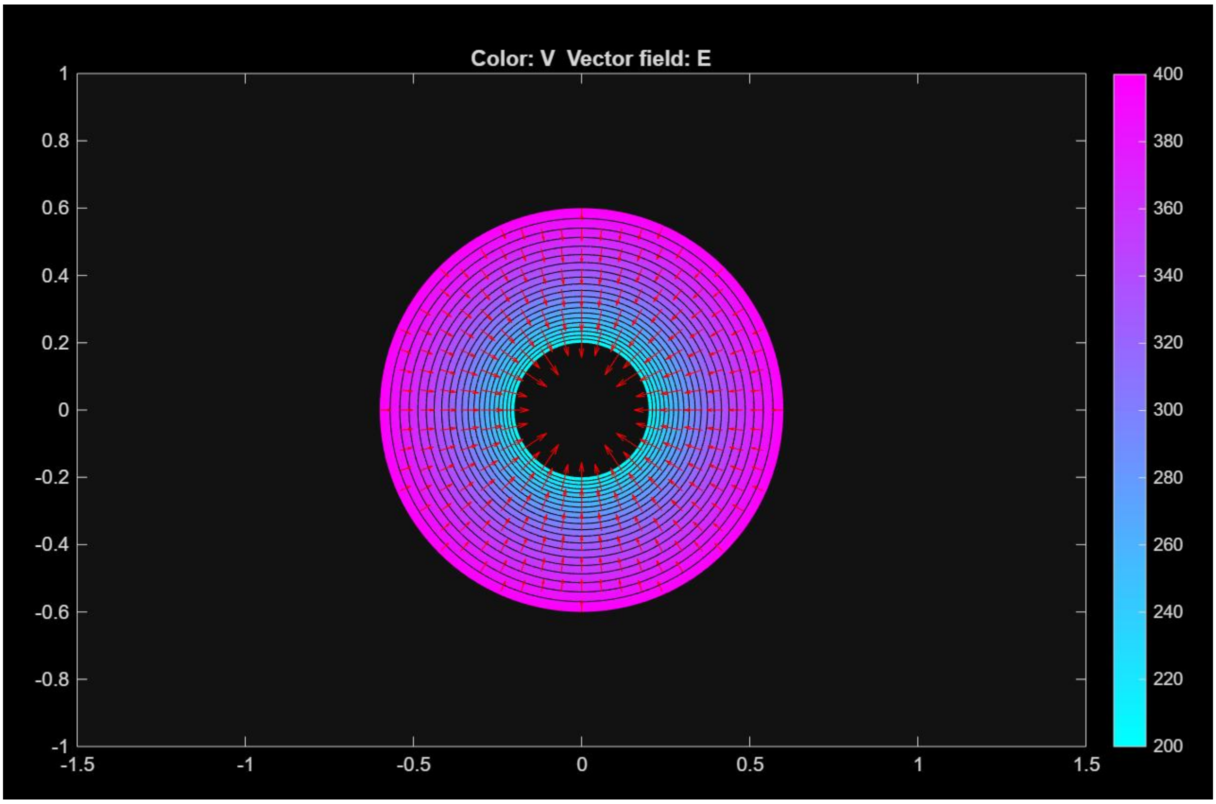
# Lab 2

## Exercise 2.3

Simulate the field in Example 2.2, letting the radius of the inner and outer circles to be 0.2 and 0.6, and their potentials to be 200 and 400. Assume $\epsilon = 1, \rho = 0$. Change the resolution of the meshing grids. Compare the results and computational time. Note that the two circles must have the same centre to simulate a coaxial cable.

```
function pdemodel
[pde_fig,ax]=pdeinit;
pdetool('appl_cb',5);
set(ax,'DataAspectRatio',[1 1 1]);
set(ax,'PlotBoxAspectRatio',[1.5 1 1]);
set(ax,'XLim',[-1.5 1.5]);
set(ax,'YLim',[-1 1]);
set(ax,'XTickMode','auto');
set(ax,'YTickMode','auto');

% Geometry description:
pdecirc(0,0,0.20000000000000001,'C1');
pdecirc(0,0,0.59999999999999998,'C2');
set(findobj(get(pde_fig,'Children'),'Tag','PDEEval'),'String','C2-C1')

% Boundary conditions:
pdetool('changemode',0)
pdesetbd(8,'dir',1,'1','400')
pdesetbd(7,'dir',1,'1','400')
pdesetbd(6,'dir',1,'1','400')
pdesetbd(5,'dir',1,'1','400')
pdesetbd(4,'dir',1,'1','200')
pdesetbd(3,'dir',1,'1','200')
pdesetbd(2,'dir',1,'1','200')
pdesetbd(1,'dir',1,'1','200')

% Mesh generation:
setappdata(pde_fig,'Hgrad',1.3);
setappdata(pde_fig,'refinemethod','regular');
setappdata(pde_fig,'jiggle',char('on','mean',''));
setappdata(pde_fig,'MesherVersion','preR2013a');
pdetool('initmesh')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
```

```
pdetool('refine')
pdetool('refine')

% PDE coefficients:
pdeseteq(1,'1.0','0.0','0','1.0','0:10','0.0','0.0','[0 100]')
setappdata(pde_fig,'currparam',['1.0';'0  '])

% Solve parameters:
setappdata(pde_fig,'solveparam',char('0','5520','10','pdeadworst','0.5','lo
ngest','0','1E-4','','fixed','Inf'))

% Plotflags and user data strings:
setappdata(pde_fig,'plotflags',[1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 0 1]);
setappdata(pde_fig,'colstring','');
setappdata(pde_fig,'arrowstring','');
setappdata(pde_fig,'deformstring','');
setappdata(pde_fig,'heightstring','');

% Solve PDE:
pdetool('solve')
```



## Task 2.1

Use MATLAB to implement the finite difference method (FDM) to find the electric potential for the following source-free squared region using 4x4 grids.

*Hint:* You need to derive by hand expressions of the **A** and **B** in equation (2-7) for the problem here. The sizes of **A** and **B** are determined by the number of grid points with unknown potentials.

The expressions for A and B are as follows:

$4V_1 - 1V_2 - 1V_4 = 150$

$-1V_1 + 4V_2 - 1V_3 - 1V_5 = 100$

$-1V_2 + 4V_3 - 1V_6 = 130$

$-1V_1 + 4V_4 - 1V_5 - 1V_7 = 50$

$-1V_2 - 1V_4 + 4V_5 - 1V_6 - 1V_8 = 0$

$-1V_3 - 1V_5 + 4V_6 - 1V_9 = 30$

$-1V_4 + 4V_7 - 1V_8 = 60$

$-1V_5 - 1V_7 + 4V_8 - 1V_9 = 10$

$-1V_6 - 1V_8 + 4V_9 = 40$

Thus, we obtain the matrices A, B and X

$$A = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 150 \\ 100 \\ 130 \\ 50 \\ 0 \\ 30 \\ 60 \\ 10 \\ 40 \end{bmatrix} \qquad X = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \end{bmatrix}$$

$X = A^{-1}B$

$$X = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \end{bmatrix} = \begin{bmatrix} 67.143 \\ 68.661 \\ 60 \\ 49.911 \\ 47.5 \\ 41.339 \\ 35 \\ 30.089 \\ 27.857 \end{bmatrix}$$

```
A = [
    4, -1, 0, -1, 0, 0, 0, 0, 0;
    -1, 4, -1, 0, -1, 0, 0, 0, 0;
    0, -1, 4, 0, 0, -1, 0, 0, 0;
    -1, 0, 0, 4, -1, 0, -1, 0, 0;
    0, -1, 0, -1, 4, -1, 0, -1, 0;
    0, 0, -1, 0, -1, 4, 0, 0, -1;
    0, 0, 0, -1, 0, 0, 4, -1, 0;
    0, 0, 0, 0, -1, 0, -1, 4, -1;
    0, 0, 0, 0, 0, -1, 0, -1, 4
];
B = [150; 100; 130; 50; 0; 30; 60; 10; 40];

x = inv(A) * B;

for i = 1:9
    fprintf('V%d = %f V\n', i, x(i));
end
```

## Output

```
>> task1
V1 = 67.142857 V
V2 = 68.660714 V
V3 = 60.000000 V
V4 = 49.910714 V
V5 = 47.500000 V
V6 = 41.339286 V
V7 = 35.000000 V
V8 = 30.089286 V
V9 = 27.857143 V
```

## Task 2.2

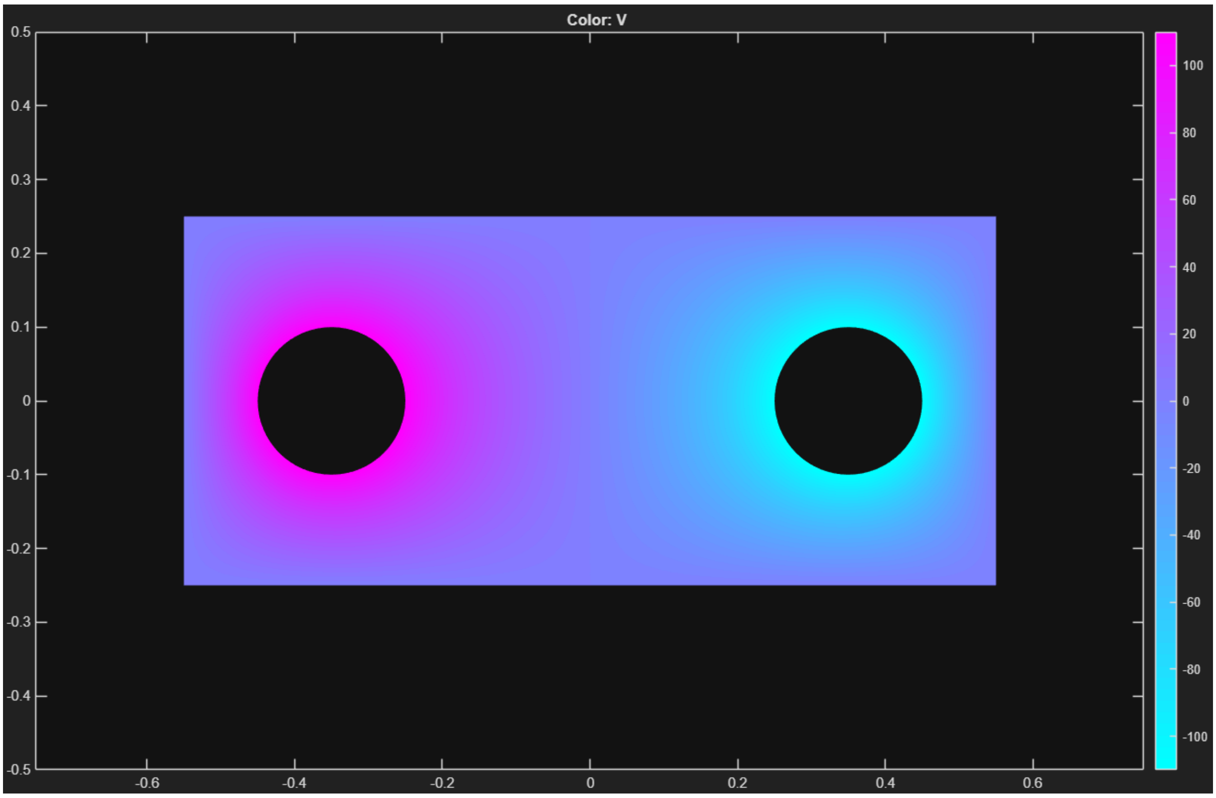
The figure below aims to simulate a two-wire transmission. Two solid metallic (ideal conductors) wires with the given sizes are contained in a rectangular region with zero charge density. The geometry is specified in Fig. 2.4.

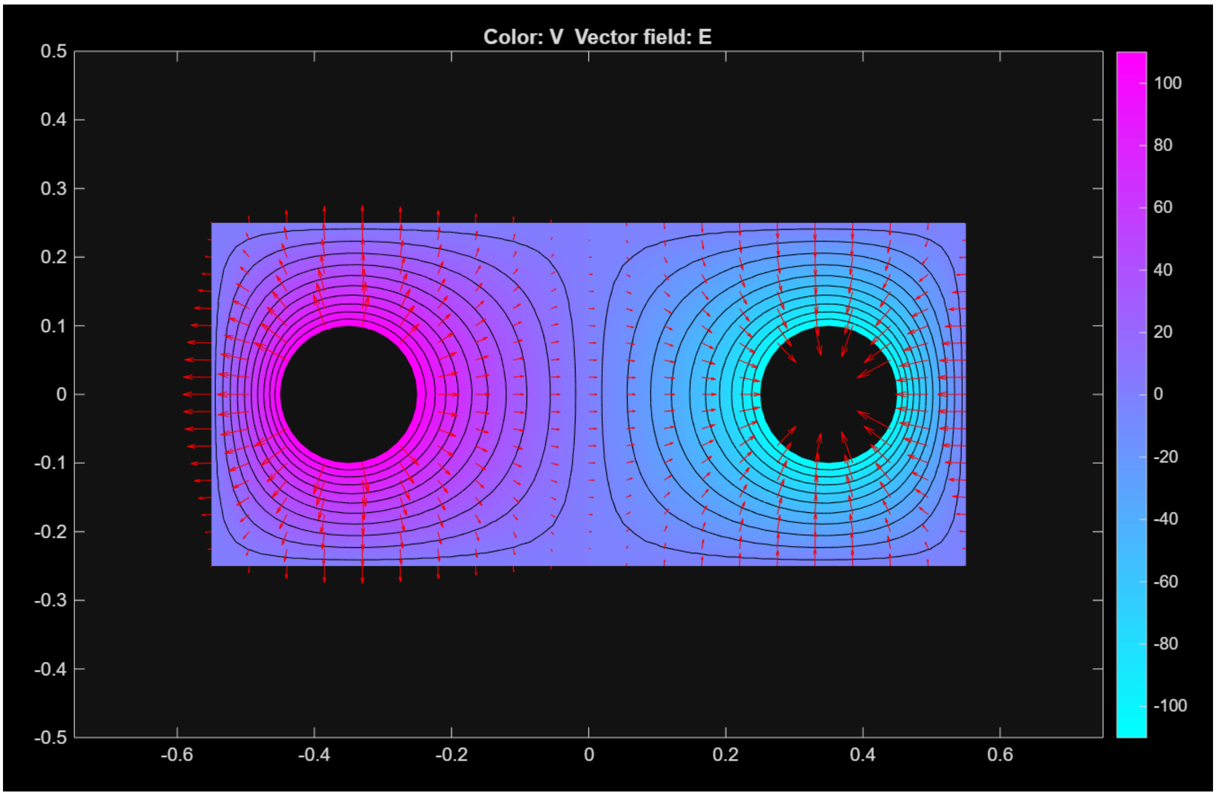
Use the PDE Toolbox to solve for the electric field shown above. Assume $\epsilon = 1, \rho = 0$

a. Set the problem in the PDE Toolbox.
   *Hint:* The center of the rectangular (R1) can be set to (0,0), the width of R1 is 0.55*2=1.1; the center of the left circle (E1) is (-0.35,0). Set the region formula to be R1-E1-E2.
b. Visualize the potential field and the electric field.

```
function pdemodel
[pde_fig, ax] = pdeinit;
pdetool('appl_cb', 5);
set(ax, 'DataAspectRatio', [1 1 1]);
set(ax, 'PlotBoxAspectRatio', [1.5 1 2]);
set(ax, 'XLim', [-0.75 0.75]);
set(ax, 'YLim', [-0.5 0.5]);
set(ax, 'XTickMode', 'auto');
set(ax, 'YTickMode', 'auto');

% Geometry description
pderect([-0.55 0.55 0.25 -0.25], 'R1');
pdecirc(-0.35, 0, 0.1, 'E1');
pdecirc(0.35, 0, 0.1, 'E2');
set(findobj(get(pde_fig, 'Children'), 'Tag', 'PDEEval'), 'String', 'R1-
E1-E2')

% Boundary conditions
pdetool('changemode', 0)
pdesetbd(12, 'dir', 1, '1', '-110')
pdesetbd(11, 'dir', 1, '1', '-110')
pdesetbd(10, 'dir', 1, '1', '-110')
pdesetbd(9, 'dir', 1, '1', '-110')
pdesetbd(8, 'dir', 1, '1', '110')
pdesetbd(7, 'dir', 1, '1', '110')
pdesetbd(6, 'dir', 1, '1', '110')
pdesetbd(5, 'dir', 1, '1', '110')
pdesetbd(4, 'dir', 1, '1', '0')
pdesetbd(3, 'dir', 1, '1', '0')
pdesetbd(2, 'dir', 1, '1', '0')
pdesetbd(1, 'dir', 1, '1', '0')
```

```
% Mesh generation
setappdata(pde_fig, 'Hgrad', 1.3);
setappdata(pde_fig, 'refinemethod', 'regular');
setappdata(pde_fig, 'jiggle', char('on', 'mean', ''));
setappdata(pde_fig, 'MesherVersion', 'preR2013a');
pdetool('initmesh')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')
pdetool('refine')

% PDE coefficients
pdeseteq(1, '1.0', '0.0', '0', '1.0', '0:10', '0.0', '0.0', '[0 100]')
setappdata(pde_fig, 'currparam', ['1.0'; '0  '])

% Solve parameters
setappdata(pde_fig, 'solveparam', char('0', '15360', '10', 'pdeadworst',
'0.5', 'longest', '0', '1E-4', '', 'fixed', 'Inf'))

% Plotflags and user data strings
setappdata(pde_fig, 'plotflags', [1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 0 1]);
setappdata(pde_fig, 'colstring', '');
setappdata(pde_fig, 'arrowstring', '');
setappdata(pde_fig, 'deformstring', '');
setappdata(pde_fig, 'heightstring', '');

% Solve PDE
pdetool('solve')
```


Color: V
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
100
80
60
40
20
0
-20
-40
-60
-80
-100


Color: V Vector field: E
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
100
80
60
40
20
0
-20
-40
-60
-80
-100

xxxxxxx