MARY OJI 101036761

FINITE DIFFERENCE METHOD

QUESTION 1

Solution to simple 2D Laplace case

```
clear all
W = 2;
L = 3;
V0 = 1;

dx = 0.2; % x mesh spacing
dy = 0.2; % y mesh spacing
nx = L/dx; % Number of points along x
ny = W/dy; % Number of points along y
```

The finite difference can be implemented using a matrix $GF = F$. V = voltage at discrete points F = force G = relation between voltages at different points G is computed using; $\frac{V_{x-1,y} - 2V_{x,y} + V_{x+1,y}}{(\Delta x)^2} + \frac{V_{x,y-1} - 2V_{x,y} + V_{x,y+1}}{(\Delta y)^2} = 0$ $

```
% Coefficients are calculated below.

c1 = -2*(1/dx^2 + 1/dy^2);
c2 = 1/(dx^2);
c3 = 1/(dy^2);
```

The mapCoordinate function takes an x,y coordinate and converts it to an index in the V array.

```
G = zeros(nx*ny,nx*ny);

for x=2:(nx-1)
    for y=2:(ny-1)
        i = coordinate(x,y,nx);
        G(i,i) = c1;
        G(i,coordinate(x-1,y,nx)) = c2;
        G(i,coordinate(x+1,y,nx)) = c2;
        G(i,coordinate(x,y-1,nx)) = c3;
        G(i,coordinate(x,y+1,nx)) = c3;
    end
end
```

The F matrix is generated with boundary set to V0, where x = 0 and x = L.

```
F = zeros(nx*ny,1);

for y=1:ny
    i = coordinate(1,y,nx);
    G(i,i) = 1;
```

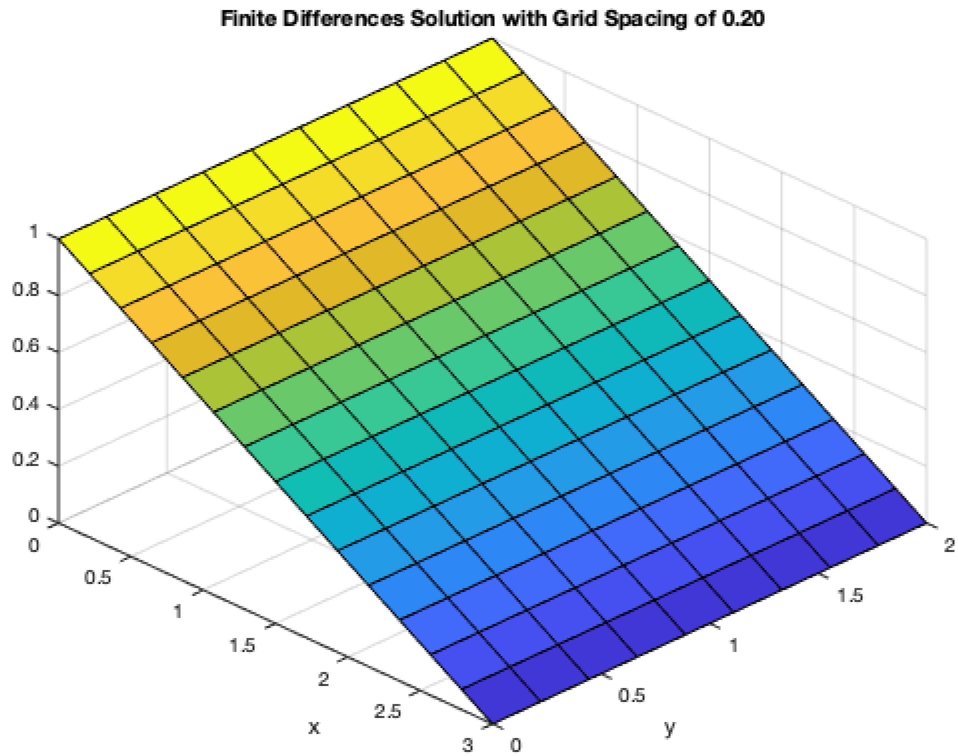
```
    F(i) = V0;

    i = coordinate(nx,y,nx);
    G(i,i) = 1;
end
```

Setting up boundary conditions for analytical solution where V=0 at the corners.

```
for x=2:(nx-1)
    i = coordinate(x,1,nx);
    G(i,i) = 1;
    G(i,coordinate(x,2,nx)) = -1;

    i = coordinate(x,ny,nx);
    G(i,i) = 1;
    G(i,coordinate(x,ny-1,nx)) = -1;
end
```

Solution from matrices

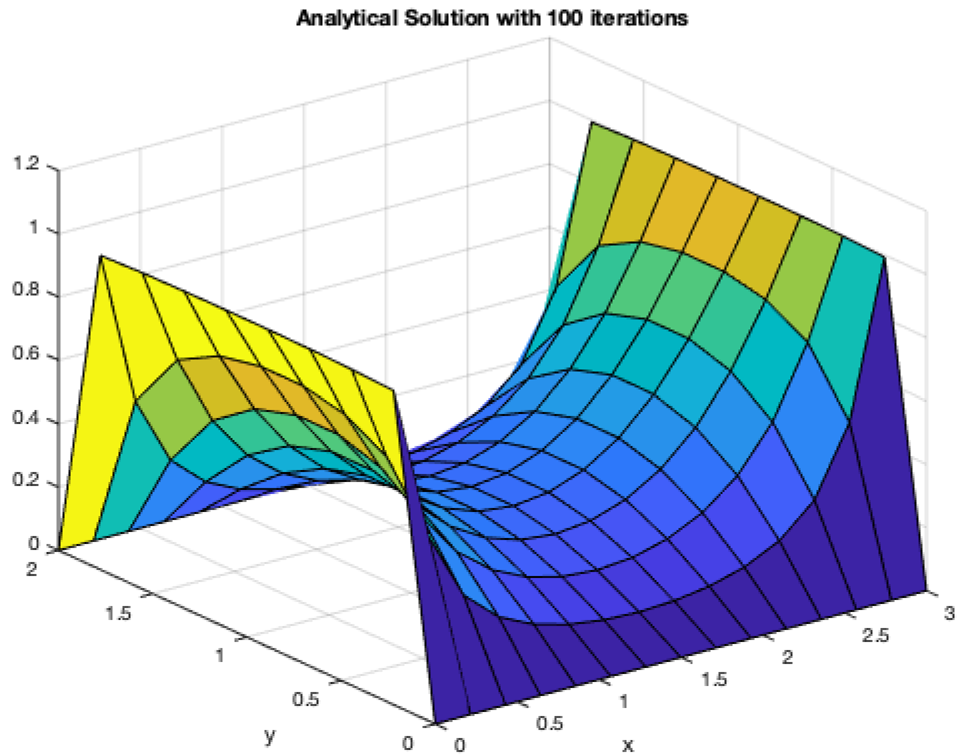
```
sol = G\F;
sol = reshape(sol,[],ny)';

figure(1);
surf(linspace(0,L,nx),linspace(0,W,ny),sol);
xlabel('x');
ylabel('y');
title(sprintf('Finite Differences Solution with Grid Spacing of %.2f',
 dx));
set(gca, 'View', [45 45])
```



Analytical solution for comparison with Finite Difference solution.

```
analyticalSol = zeros(ny, nx);
x1 = repmat(linspace(-L/2,L/2,nx),ny,1);
y1 = repmat(linspace(0,W,ny),nx,1)';
iter = 100;
avgError = zeros(iter,1);


for i=1:iter
    n = 2*i - 1;
    analyticalSol = analyticalSol + 1./n.*cosh(n.*pi.*x1./W) ...
        ./cosh(n.*pi.*(L./2)./W).*sin(n.*pi.*y1./W);

    avgError(i) = mean(mean(abs(analyticalSol.*4.*V0./pi - sol)));
end

analyticalSol = analyticalSol.*4.*V0./pi;

figure(2);
surf(linspace(0,L,nx),linspace(0,W,ny),analyticalSol);
xlabel('x');
ylabel('y');
title(sprintf('Analytical Solution with %d iterations', iter));

%figure(3);
```

```
%plot(1:i,avgError);
%xlabel('Iteration');
%ylabel('Average Error (V)');
%title('Convergence of Analytical Solution');
%grid on;
```



MESHING AND COMMENTS ON NUMERICAL VS ANALYTICAL

- The analytical solution is easier to implement aside from finding the expression for the analytical solution.
- For Finite Difference, there is more to the implementation because it requires the generation of the G and F matrices.
- For complex problems, FD will be preferred as it is more flexible than finding the analytical solution. Finding the analytical solution for complex problems will be difficult.

QUESTION 2

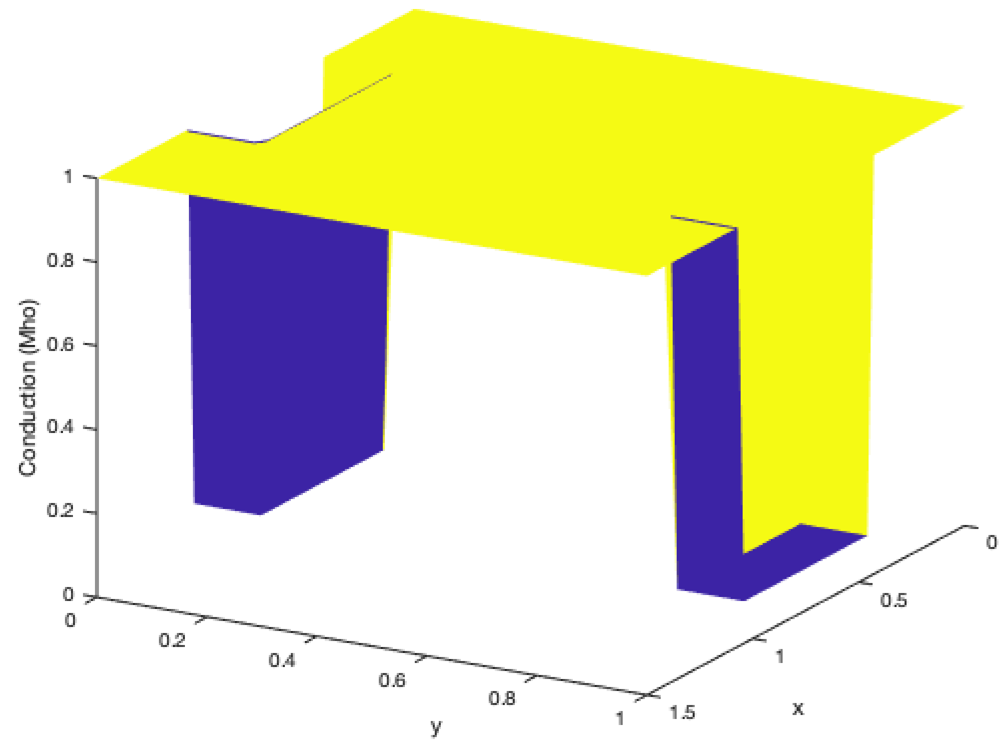
Solving current flow in a rectangular region using Finite Diffrenece Method.

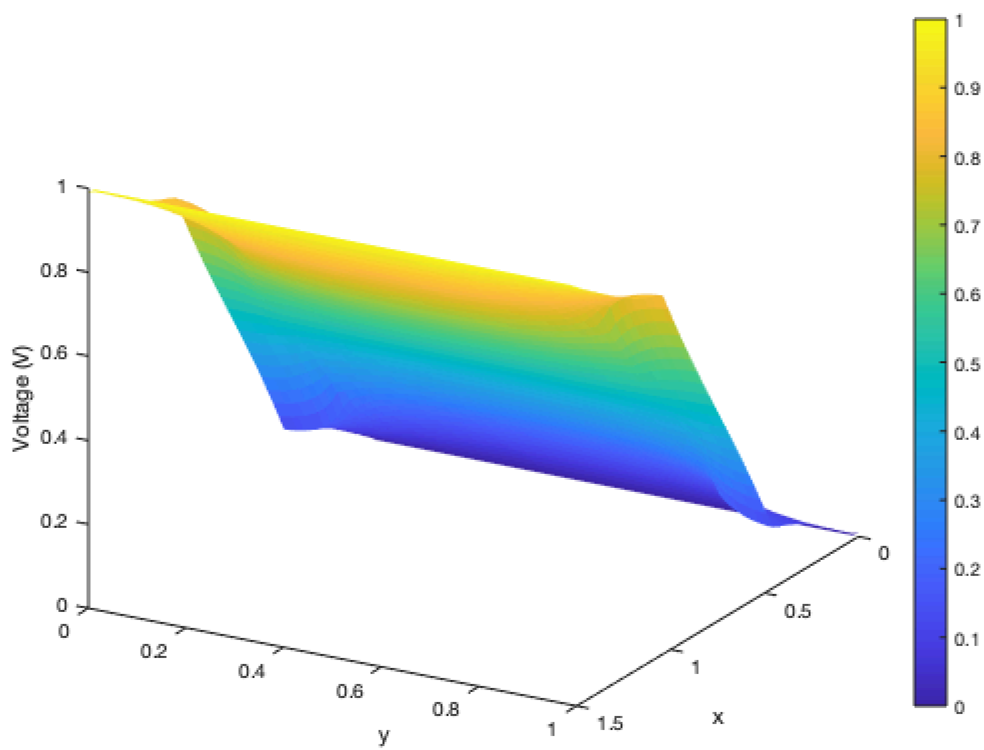
PART A

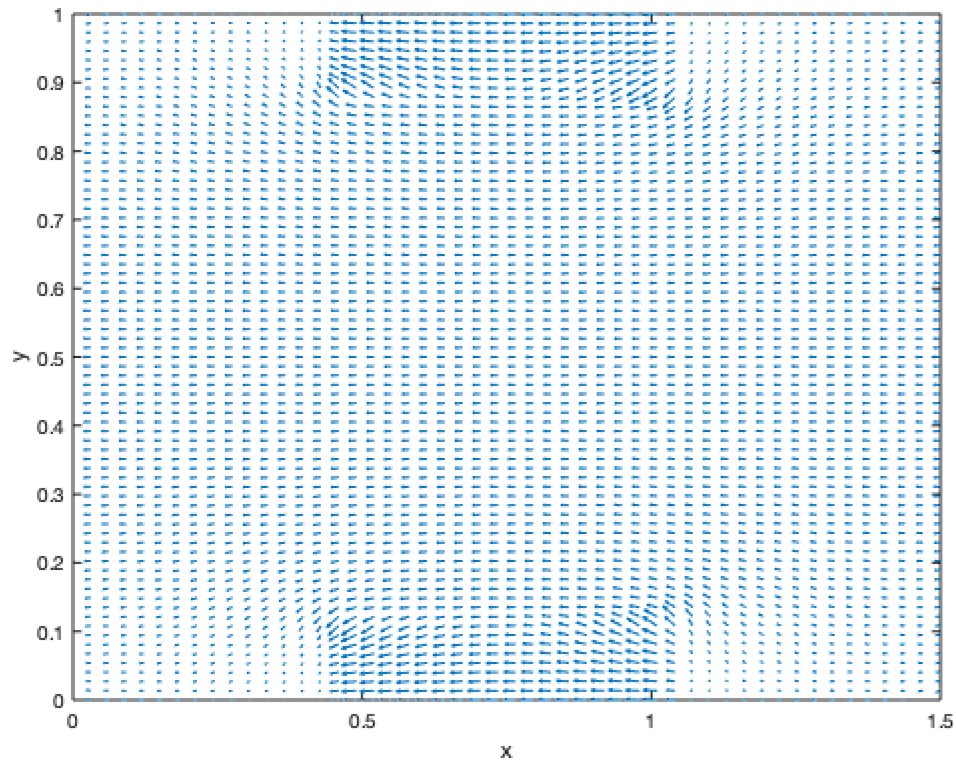
```
nx = 75;
ny = 50;
Lb = 20;
Wb = 10;
V1 = 1;
```

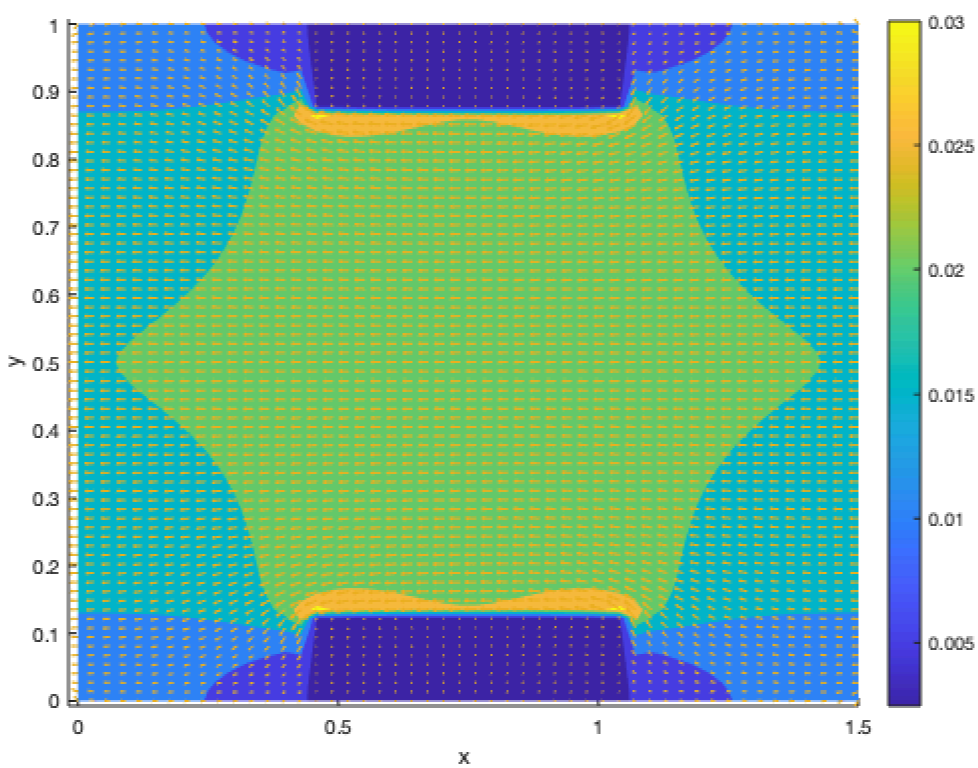
```
figure(4);
hold on;
% Generating the map of conductivity of the area
sigma_conduct = 1;
sigma_insulate = 10e-2;
cMap = sigma_conduct*ones(nx, ny);
cMap(1:Wb,(1:Lb)+ny/2-Lb/2) = sigma_insulate;
cMap((1:Wb)+nx-Wb,(1:Lb)+ny/2-Lb/2) = sigma_insulate;
surf(linspace(0,1.5,ny), linspace(0,1,nx),
 cMap,'EdgeColor','none','LineStyle','none');
xlabel('x');
ylabel('y');
zlabel('Conduction (Mho)');
view([120 25])


% Numeric solution
V = numericSolution(nx, ny, cMap, Inf, Inf, 0, V1);
figure(5);
hold on;
surf(linspace(0,1.5,ny), linspace(0,1,nx),
 V,'EdgeColor','none','LineStyle','none');
xlabel('x');
ylabel('y');
zlabel('Voltage (V)');
view([120 25])
colorbar

% Electric field
[Ex, Ey] = gradient(V);
Ex = -Ex;
Ey = -Ey;
figure(6);
quiver(linspace(0,1.5,ny), linspace(0,1,nx), Ex, Ey);
ylim([0 1]);
xlim([0 1.5]);
xlabel('x');
ylabel('y');

% Current density
Jx = cMap.*Ex;
Jy = cMap.*Ey;
J = sqrt(Jx.^2 + Jy.^2);
figure(7);
hold on;
contourf(linspace(0,1.5,ny), linspace(0,1,nx),
 J,'EdgeColor','none','LineStyle','none');
quiver(linspace(0,1.5,ny), linspace(0,1,nx), Jx, Jy);
xlabel('x');
ylabel('y');
colorbar

%
% *PART B*
```

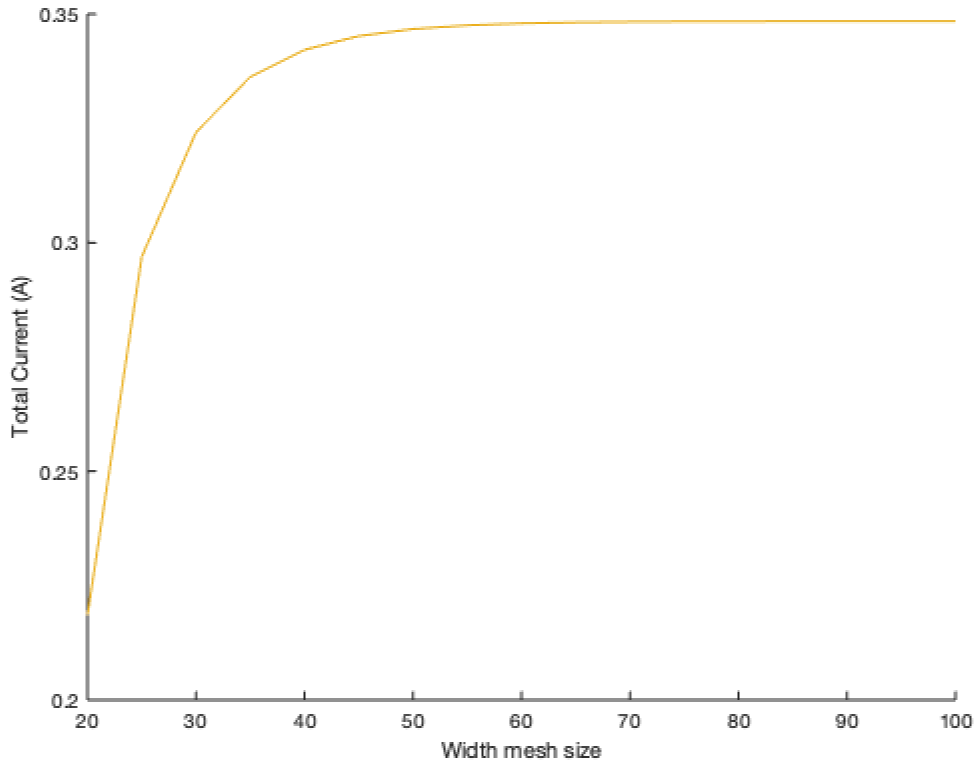
```
figure(8);
hold on;
range = 20:5:100;
I = [];
for x = range
    I = [I totalI(x, ny, V1, sigma_conduct, sigma_insulate, Wb, Lb)];
end
plot(range, I);
ylabel('Total Current (A)');
xlabel('Width mesh size');

%
% *PART C*
figure(9);
range = 0:1:50;
I = [];
for W = range
    I = [I totalI(nx, ny, V1, sigma_conduct, sigma_insulate, W, Lb)];
end
plot(range, I);
ylabel('Total Current (A)');
xlabel('Box width');

%
% *PART D*
figure(10);
hold on;
range = logspace(-5,0, 50);
I = [];
for sigma = range
    I = [I totalI(nx, ny, V1, sigma_conduct, sigma, Wb, Lb)];
end
plot(range, I);
ylabel('Total Current (A)');
xlabel('Box Conduction (Mho)');
```


Conduction (Mho)
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
y
0
0.5
1
1.5
x

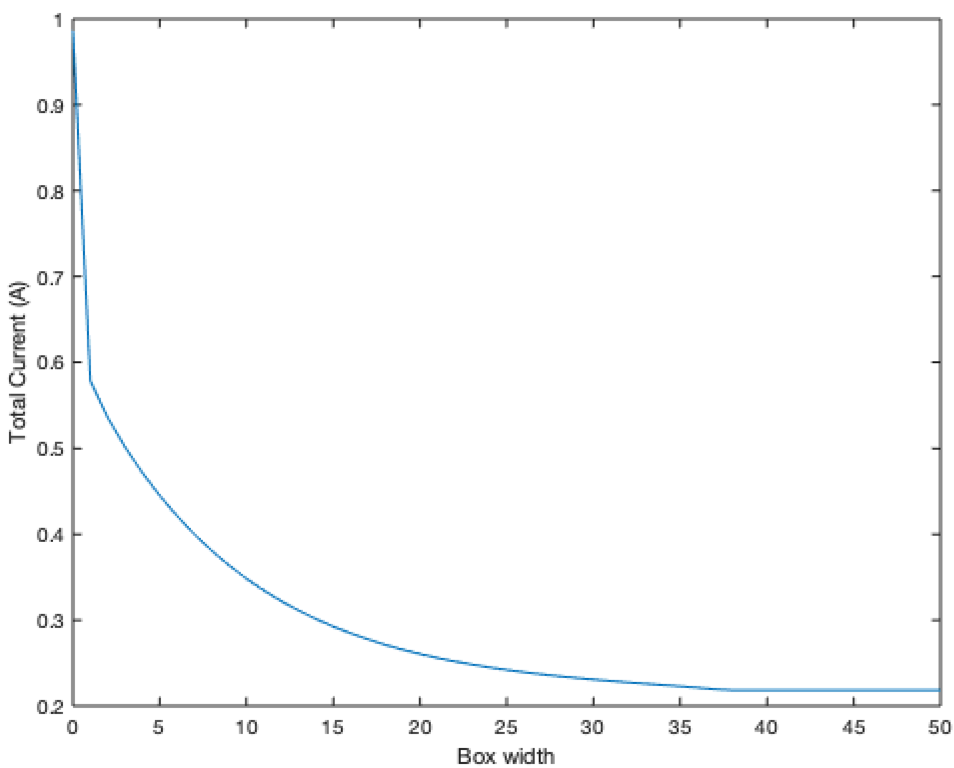

Voltage (V)
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
y
0
0.5
1
1.5
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1.5

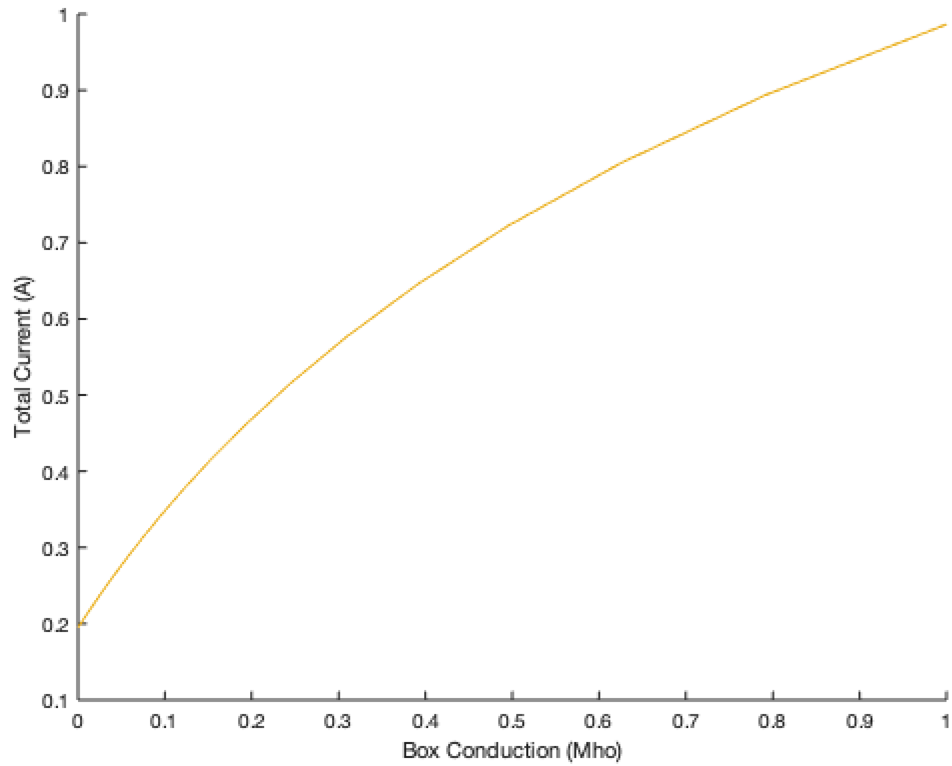

y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1.5
0.03
0.025
0.02
0.015
0.01
0.005



0.35
0.3
0.25
0.2
Total Current (A)
20
30
40
50
60
70
80
90
100
Width mesh size



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
Total Current (A)
0
5
10
15
20
25
30
35
40
45
50
Box width



Published with MATLAB® R2019b