
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Harmonic Wave Equation in 2D FD and Modes %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% By David, Patrobas, Andrew and Xiaochen
% Febuary 24th, 2019
% Assignment 2
% Patrobas Adewumi
global C;
C.q_0 = 1.60217653e-19;           % electron charge
C.hb = 1.054571596e-34;          % Dirac constant
C.h = C.hb * 2 * pi;              % Planck constant
C.m_0 = 9.10938215e-31;           % electron mass
C.kb = 1.3806504e-23;             % Boltzmann constant
C.eps_0 = 8.854187817e-12;        % vacuum permittivity
C.mu_0 = 1.2566370614e-6;         % vacuum permeability
C.c = 299792458; % speed of light

nx = 75;
L = nx;% length
ny = 50;
W = ny; % width

dx = 1;
dy = 1;
G = sparse(nx*ny, ny*nx);
V = ones(nx*ny,1);
alpha = (C.hb^2) / (2 * C.m_0);

map = @(i,j) j + (i - 1)*ny;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Part a %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set boundary conditions

for i=1:nx
    for j=1:ny
        n = map(i,j);
        nxm = map(i-1,j);
        nxp = map(i+1,j);
        nym = map(i,j-1);
        nyp = map(i,j+1);

        % when Length = 0 (V = Vo)
        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;
            V(n) = 1;
        % When lenght is some given length, L (V = 0)
        elseif i == nx
            G(n,:) = 0;
            G(n,n) = 1;
            V(n) = 0;
        elseif (j == 1 || j == ny)
            G(n,:) = 0;
            G(n,n) = -3;

```

```

        G(n,nxm) = 1;
        G(n,nxp) = 1;
        G(n,nyp) = 1;
    else
        G(n,:) = 0;
        G(n,n) = -4;
        G(n,nxm) = 1;
        G(n,nxp) = 1;
        G(n,nym) = 1;
        G(n,nyp) = 1;
    end
end
end

% GV = F Solve for F
F = G\V;
surfs_up = zeros(nx,ny);
for i = 1:nx
    for j = 1:ny
        n = map(i,j);
        surfs_up(i,j) = F(n);
    end
end

figure(1)
surf(surfs_up)
% I am a simple man and so is the colormap
colormap default
shading interp
colorbar

title('Electrostatic Potential in Rectangular Region L/W = 3/2')
xlabel('Width')
ylabel('Length')
zlabel('Voltage')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Part b %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set boundary conditions

for i=1:nx

    for j=1:ny
        n = map(i,j);
        nxm = map(i-1,j);
        nxp = map(i+1,j);
        nym = map(i,j-1);
        nyp = map(i,j+1);

        if i == 1
            G(n,:) = 0;
            G(n,n) = 1;

```

```

        V(n) = 1;

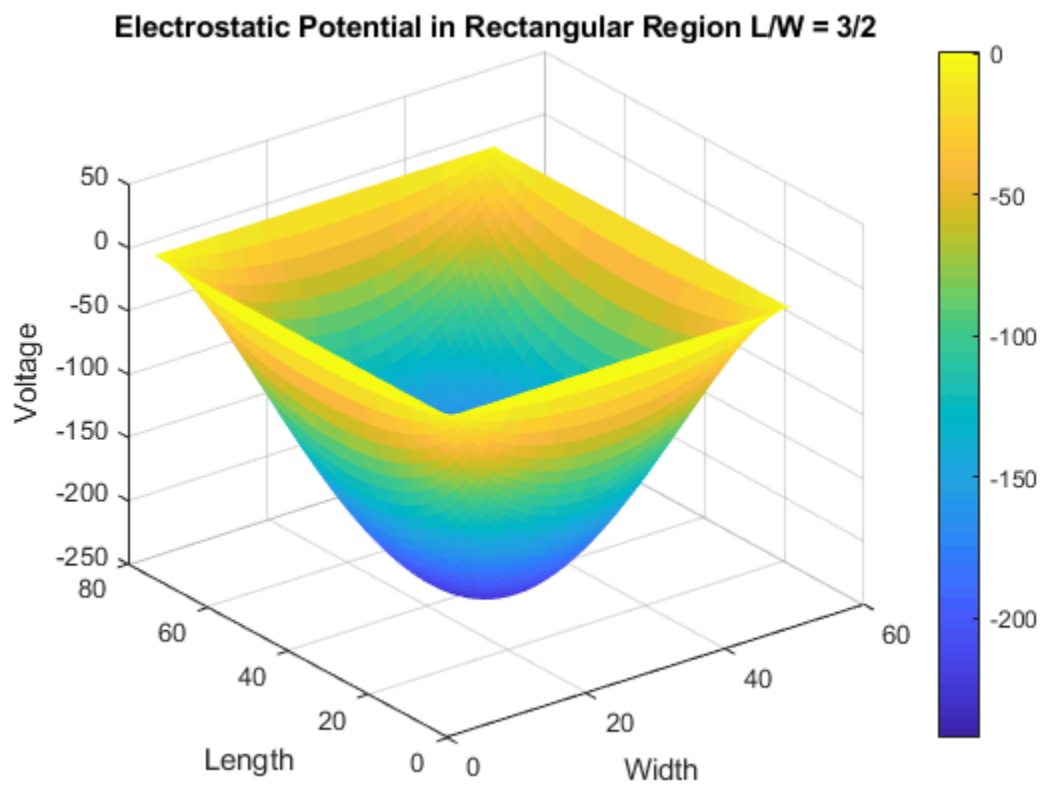
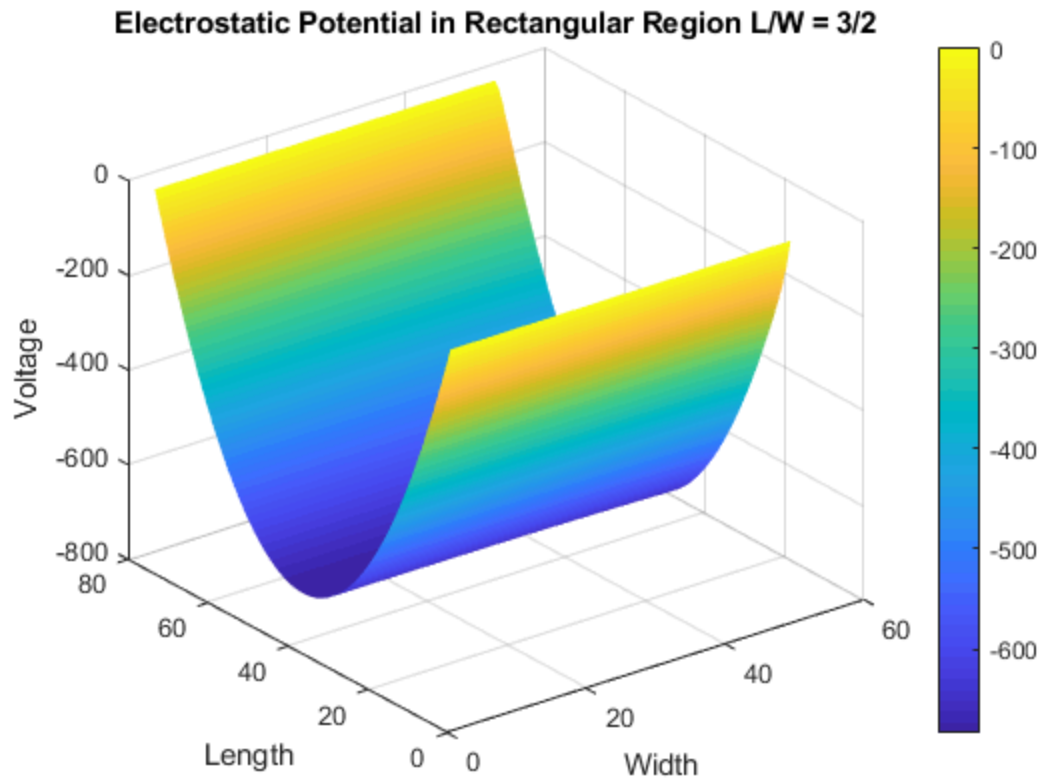
    elseif i == nx
        G(n,:) = 0;
        G(n,n) = 1;
        V(n) = 1;
    elseif j == 1
        G(n,:) = 0;
        G(n,n) = 1;
        V(n) = 0;
    elseif j == ny
        G(n,:) = 0;
        G(n,n) = 1;
        V(n) = 0;
    else
        G(n,:) = 0;
        G(n,n) = -4;
        G(n,nxm) = 1;
        G(n,nxp) = 1;
        G(n,nym) = 1;
        G(n,nyp) = 1;
    end
end
end

% GV = F Solve for F
F = G\V;
% Set up a surf plot
surfs_up = ones(nx,ny);
for i = 1:nx
    for j = 1:ny
        n = map(i,j);
        surfs_up(i,j) = F(n);
    end
end

figure(2)
surf(surfs_up)
% I am a simple man and so is the colormap
colormap default
shading flat
colorbar

title('Electrostatic Potential in Rectangular Region L/W = 3/2')
xlabel('Width')
ylabel('Length')
zlabel('Voltage')

```



Publish this report

Meshing becomes an accurate tool when the amount of points used becomes near infinite. Analytical solutions can be obtained exactly with pencil and paper; Numerical solutions cannot be obtained exactly in finite time and typically cannot be solved using pencil and paper.

```
% These distinctions, however, can vary. There are increasingly many
% theorems
% and equations that can only be solved using a computer; however, the
% computer
% doesn't do any approximations, it simply can do more steps than any
% human can ever hope to do without error.

% In numerical computing, we specify a problem, and then crunch
% numbers in a very well-defined, carefully-constructed order.
% If we are very careful about the way in which the numbers are
% crunched,
% we can guarantee that the result is only slightly inaccurate, and
% usually close enough for its intended purpose.
% Numerical solutions very rarely can contribute to proofs of new
% ideas.
% Analytic solutions are generally considered to be "stronger".
% The thinking goes that if we can get an analytic solution, it is
% exact,
% and then if we need a number at the end of the day, we can just plug
% numbers into the analytic solution.
% However, even if analytic solutions can be found, they might not be
% able to be computed quickly.
% As a result, numerical approximation will never go away, and both
% approaches contribute holistically to the fields of mathematics and
% quantitative sciences.
```

Published with MATLAB® R2018a

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Harmonic Wave Equation in 2D FD and Modes %%%%%%%%%%%%%%
% By David, Patrobas, Andrew and Xiaochen
% Febuary 24th, 2019
% Assignment 2
% Patrobas Adewumi
global C;
C.q_0 = 1.60217653e-19;           % electron charge
C.hb = 1.054571596e-34;          % Dirac constant
C.h = C.hb * 2 * pi;              % Planck constant
C.m_0 = 9.10938215e-31;          % electron mass
C.kb = 1.3806504e-23;            % Boltzmann constant
C.eps_0 = 8.854187817e-12;       % vacuum permittivity
C.mu_0 = 1.2566370614e-6;        % vacuum permeability
C.c = 299792458; % speed of light

% Define area of region
W = 20; % width in y dir
L = W*3/2; % length in x dir

% Centre point of given region
mid_x = L/2;
mid_y = W/3;

% Setting up the matrices for evaluation
G = zeros(L*W,L*W);
B = zeros(L*W,1);

% Defining conductivity of the boxes (given area)
s1 = 1;
s2 = 0.01;

% Define resistive region size
res_L = L*1/4;
res_W = W*2/5;

Smap = ones(L,W);
for i = 1:L
    for j = 1:W
        n = j+(i-1)*W;
        nxm = j+(i-2)*W;
        nxp = j+i*W;
        nyp = j+1+ (i-1)*W;
        nym = j-1+ (i-1)*W;

        if(i == 1)
            G(n,:) = 0;
            G(n,n) = 1;
            B(n) = 1;
            Smap(i,j) = s1;
        elseif(i == L)
            G(n,:) = 0;
            G(n,n) = 1;

```

```

        B(n) = 0;
        Smap(i,j) = s1;
elseif(j == 1)
    G(n,:) = 0;
    if(i > mid_x - (res_L/2) && i < mid_x + (res_L/2))
        G(n,nxm) = s2;
        G(n,nxp) = s2;
        G(n,nyp) = s2;
        G(n,n) = -3*s2;
        Smap(i,j) = s2;
    else
        G(n,nxm) = s1;
        G(n,nxp) = s1;
        G(n,nyp) = s1;
        G(n,n) = -3;
        Smap(i,j) = s1;
    end
elseif(j == W)
    G(n,:) = 0;
    if(i > mid_x - (res_L/2) && i < mid_x + (res_L/2))
        G(n,nxm) = s2;
        G(n,nxp) = s2;
        G(n,nym) = s2;
        G(n,n) = -3*s2;
        Smap(i,j) = s2;
    else
        G(n,nxm) = s1;
        G(n,nxp) = s1;
        G(n,nym) = s1;
        G(n,n) = -3;
        Smap(i,j) = s1;
    end
else
    G(n,:) = 0;
    G(n,n) = -4;
    % setting my X and Y Boundaries

    if((i > mid_x - (res_L/2) && i < mid_x + (res_L/2)) && ...
        (j > mid_y + (res_W/2) || j < mid_y - (res_W/2)))
        G(n,nxp) = s2;
        G(n,nxm) = s2;
        G(n,nyp) = s2;
        G(n,nym) = s2;
        Smap(i,j) = s2;
    else
        G(n,nxp) = s1;
        G(n,nxm) = s1;
        G(n,nyp) = s1;
        G(n,nym) = s1;
        Smap(i,j) = s1;
    end
end
end
end
end

```

```
V = G\B;
Vmap = zeros(L,W);
for i =1:1:L
    for j = 1:1:W
        n = j+(i-1)*W;
        Vmap(i,j) = V(n);
    end
end

[Ey,Ex] = gradient(Vmap);

E = gradient(Vmap);
J = Smap.*E;

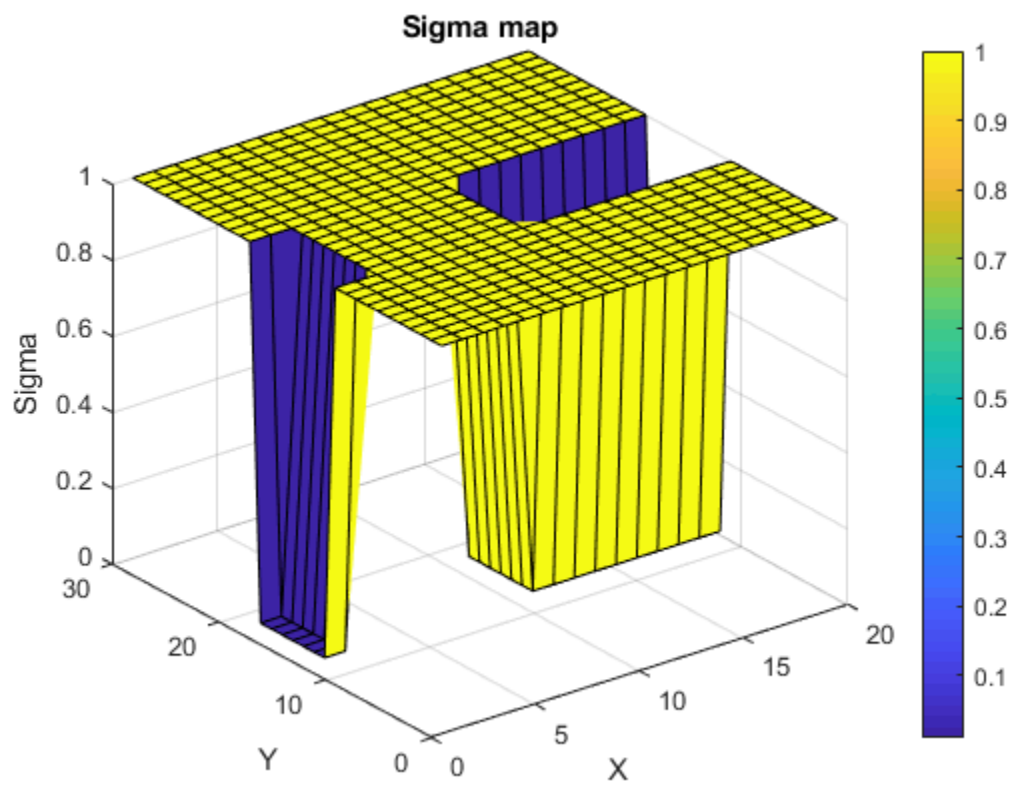
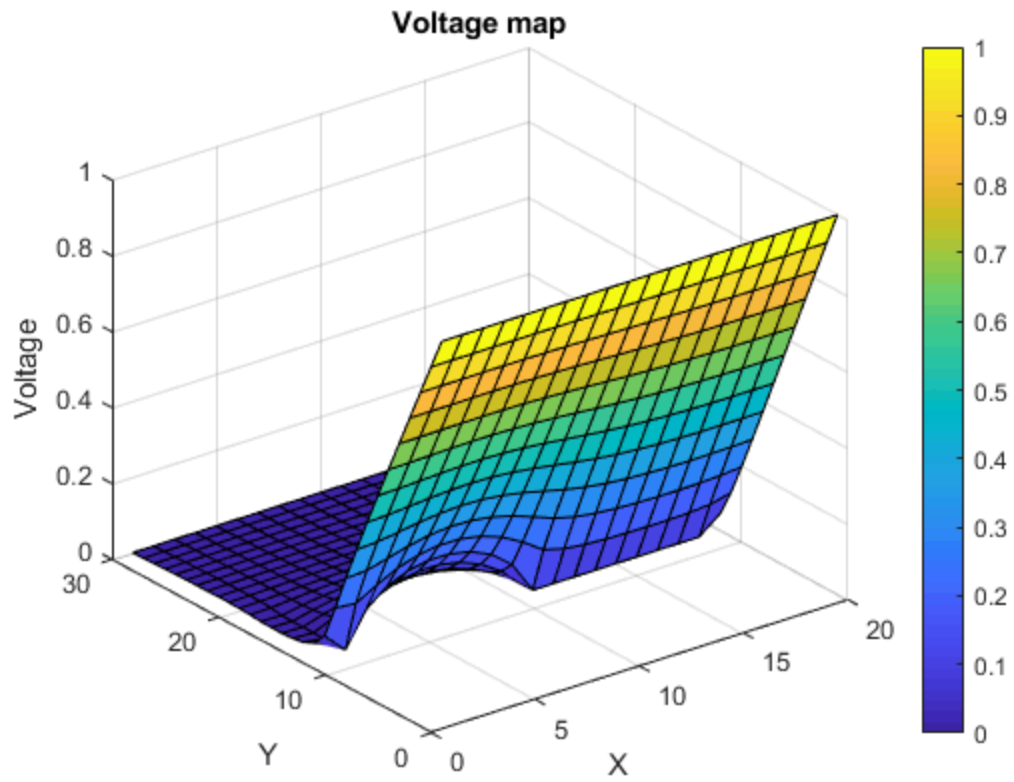
figure(5)
surf(Vmap)
colorbar
title('Voltage map')
xlabel('X')
ylabel('Y')
zlabel('Voltage')

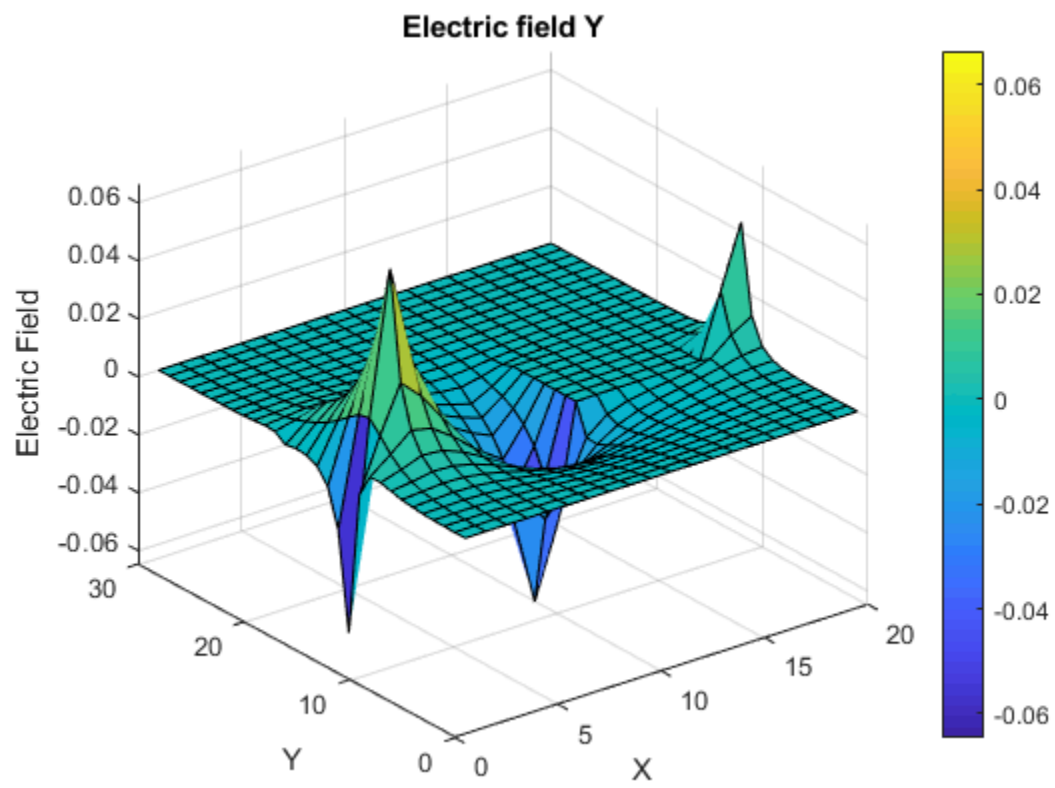
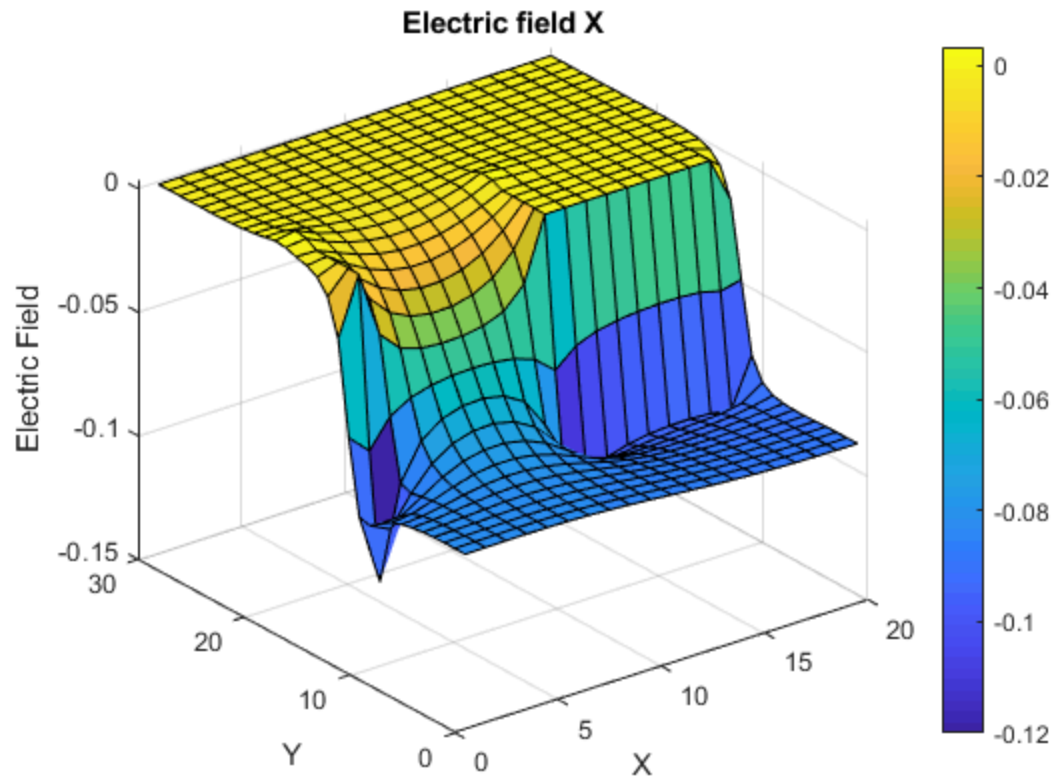
figure(6)
surf(Smap)
colorbar
title('Sigma map')
xlabel('X')
ylabel('Y')
zlabel('Sigma');

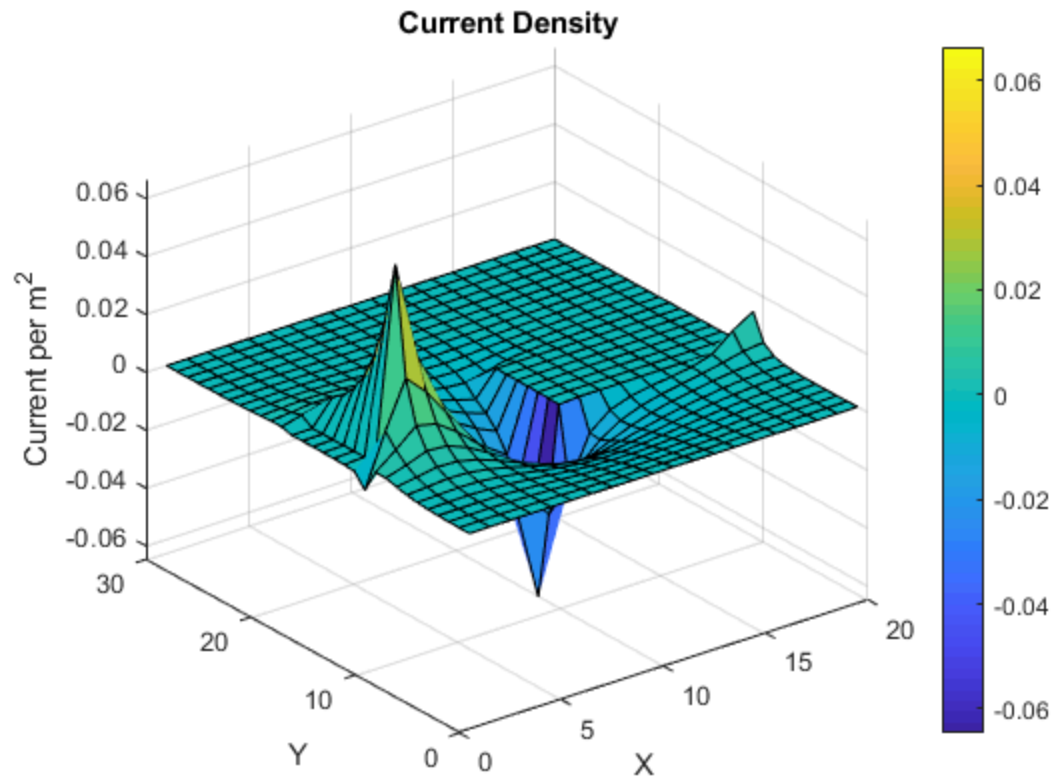
figure(7)
surf(Ex)
colorbar
title('Electric field X')
xlabel('X')
ylabel('Y')
zlabel('Electric Field');

figure(8)
surf(Ey)
colorbar
title('Electric field Y')
xlabel('X')
ylabel('Y')
zlabel('Electric Field');

figure(9)
surf(J)
colorbar
title('Current Density')
xlabel('X')
ylabel('Y')
zlabel('Current per m^2');
```



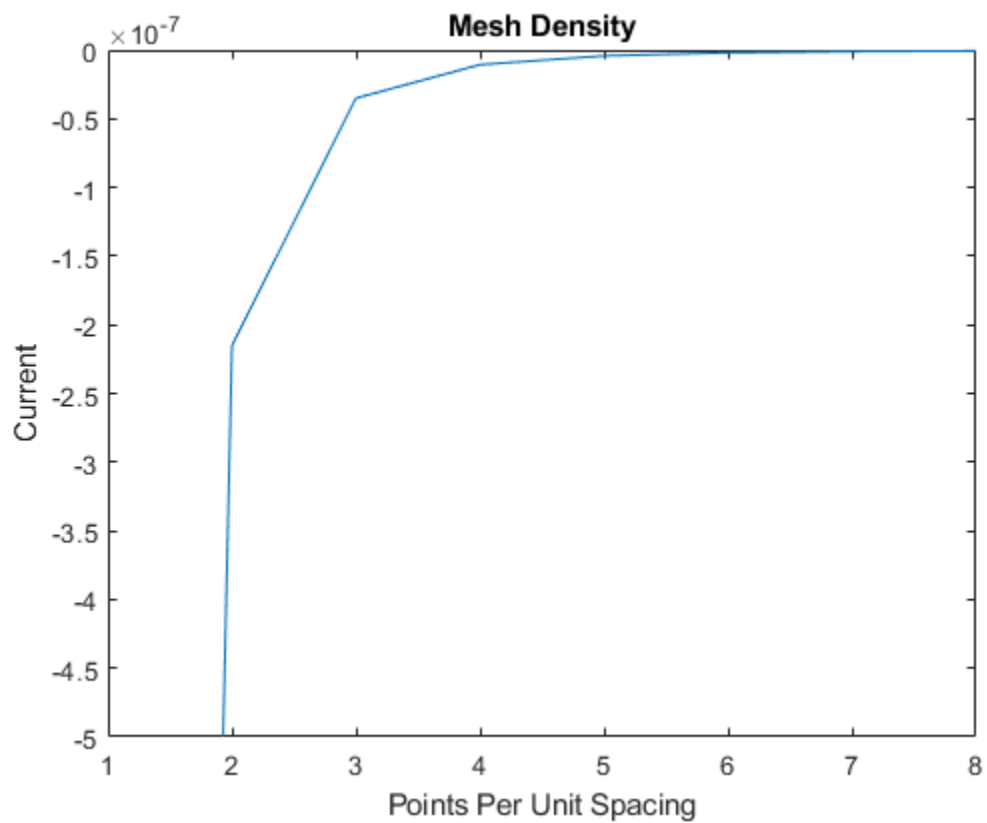




Published with MATLAB® R2018a

```
loop = 8;
current = zeros(1,loop);
for i = 1:1:loop
    current(i) = a2part2b(i);
end

figure(10)
plot(1:1:loop,current);
title('Mesh Density')
xlabel('Points Per Unit Spacing')
ylabel('Current')
ylim([-0.5e-6 0])
```

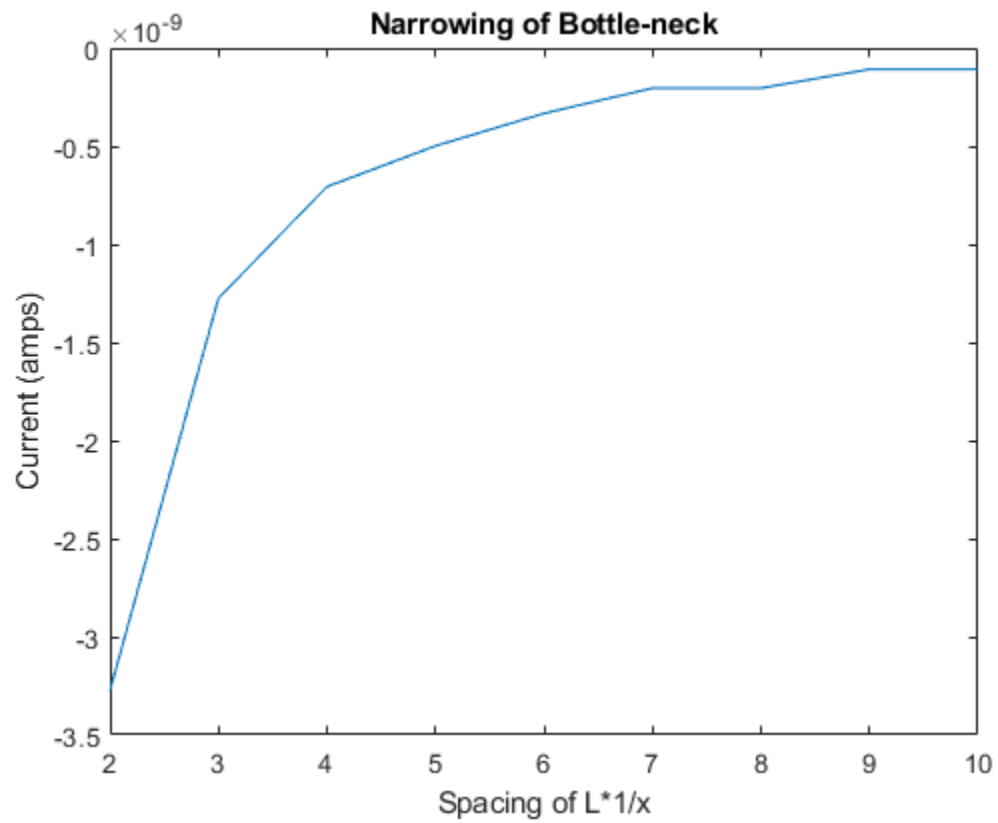


Published with MATLAB® R2018a

```
loop = 10;

current = zeros(1,loop);
for i = 2:1:loop
    current(i) = a2part2c(1/i);
end

figure(11)
plot(1:1:loop,current);
title('Narrowing of Bottle-neck')
xlabel('Spacing of  $L*1/x$ ')
ylabel('Current (amps)')
xlim([2 loop])
```

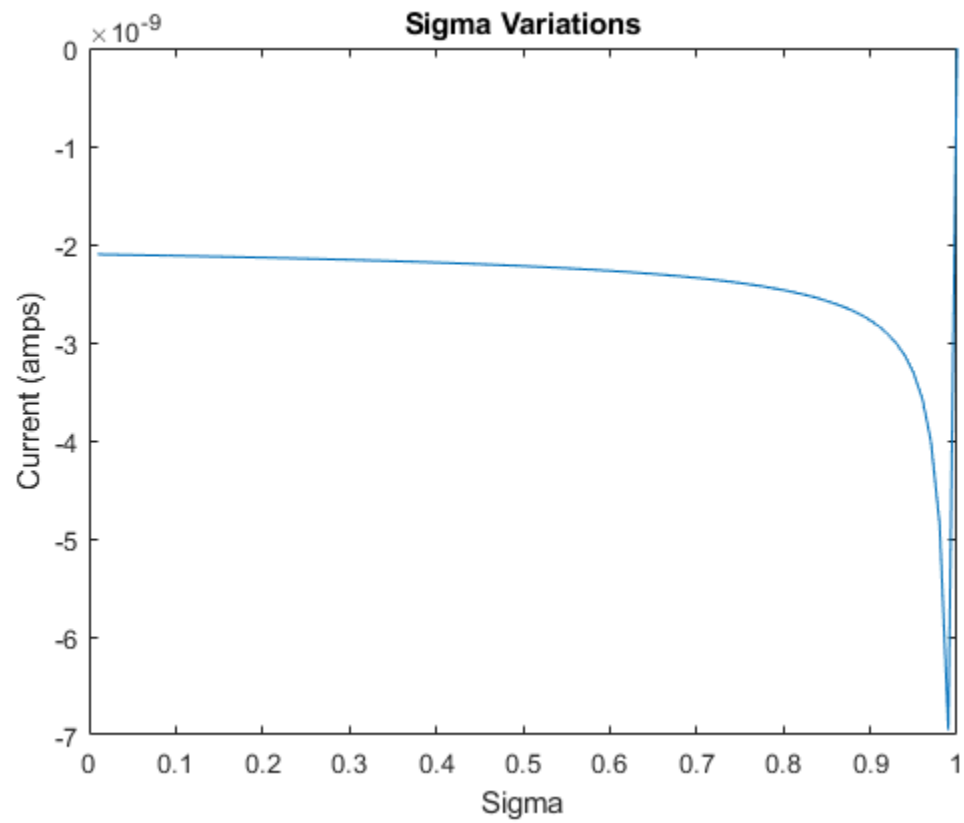


Published with MATLAB® R2018a

```
loop = 100;

current = zeros(1,loop);
for i =1:1:loop
    sig = i*1e-2;
    current(i) = a2part2d(sig);
end

figure(12)
plot(1e-2:1e-2:1,current);
title('Sigma Variations')
xlabel('Sigma')
ylabel('Current (amps)')
```



Published with MATLAB® R2018a