
```

L=50;
H=50;
dx=int64(L/4);%distance between plates
h=int64(L/2);%height of plates
t=1;% plate thickness (1 gives 1 gridline of thickness)
Vpl= 1; %fixed plate potential value
%set up box and secondary box
V=zeros(H,L);
%put plates into V and make adjustments for weird dimensions
if mod(L,2)==0 && mod(h,2)~=0
    bit = 1;
else
    bit = 0;
end
plate = [zeros(int64((H-h)/2)-bit, 1); Vpl.*ones(h,1); zeros(int64((H-
h)/2) ,1)];
%plate= [zeros(int64((L-h)/2)-bit ,1); Vpl.*ones(h,1); zeros(int64((L-
h)/2) , 1)];
pos1=int64(L/2- dx/2);
topPlaterow = int64((H-h)/2)-bit + 1;
botPlaterow = int64((H-h)/2)-bit+h;
%leftPlatecol = pos1-1;
leftPlatecol = pos1;
rightPlatecol = pos1+dx+1;
%for larger thickness t, add t plates outside each plate
for i = 0:t-1
    V(:,pos1-i)=plate;
    V(:,rightPlatecol+i)=-1.*plate;
end

%assume left is + and right is -
%fill in boundary conditions
%LHS parallel to left plate
%V( topPlaterow:botPlaterow,1) = double(Vpl)/double(leftPlatecol-1-
(t-1));
%RHS parallel to right plate
%V( topPlaterow:botPlaterow,L) = -double(Vpl)/double(L-(rightPlatecol
+(t-1)));
%fill in the bits on LHS and RHS above and below plates

%for i = 1:topPlaterow
    % V(i,1) = double(Vpl)/double(leftPlatecol-1-(t-1) + topPlaterow -
i) ;
    % V(i,L) = -double(Vpl)/double(L-(rightPlatecol+(t-1)) +
topPlaterow - i) ;
%end
%for i = 1:L-botPlaterow
    % V(botPlaterow+i,1) = double(Vpl)/ double(leftPlatecol-1-(t-1) +
i);
    % V(botPlaterow+i,L) = -double(Vpl)/ double(L-(rightPlatecol+(t-1))+
i);
%end

```

```

%fill in top and bottom boundary conditions
%boundaries on top bottom take into account superposition of plate
%potentials, left and right assume potential contribution is on due to
%nearby plate (as of now, it should be reasonable approximation with
    large
%enough L

% + plate
%for i = 1:leftPlatecol-1-(t-1)
    % V(1,i) = double(Vpl)/double( leftPlatecol-1-(t-1)+topPlaterow-i);
    % V(L,i) = double(Vpl)/double( leftPlatecol-1-(t-1)+(L-botPlaterow
+1)-i);
%end
%for i =1:L-leftPlatecol
    % V(1,leftPlatecol+i) = double(Vpl)/double( topPlaterow-1+i ) ;
    % V(L,leftPlatecol+i) = double(Vpl)/double( (L-botPlaterow)+i);
%end
% - plate
%for i = 1:L-(rightPlatecol+(t-1))
    % V(1,rightPlatecol+(t-1)+i) = V(1,rightPlatecol+i) + -
double(Vpl)/double(topPlaterow-1+i);
    % V(L,rightPlatecol+(t-1)+i) = V(1,rightPlatecol+i) + -
double(Vpl)/double( (L-botPlaterow)+i);
%end

% for i = 1:rightPlatecol-1
    % V(1,i) = V(1,i) + -double(Vpl)/
double(topPlaterow-1+rightPlatecol-i);
    % V(L,i) = V(L,i) + -double(Vpl)/double(rightPlatecol+(L-
botPlaterow)-i);
%end

%now for superpositioning with boundaries directly above and below
plates
%of thickness t
% for i = 1:t
    % V(1,leftPlatecol+1-i)=V(1,leftPlatecol+1-i) + double(Vpl)/
double(topPlaterow-1);
    % V(L,leftPlatecol+1-i)=V(L,leftPlatecol+1-i) + double(Vpl)/
double(L-botPlaterow);
    % V(1,rightPlatecol-1+i)=V(1,rightPlatecol-1+i) + -double(Vpl)/
double(topPlaterow-1);
    % V(L,rightPlatecol-1+i)=V(L,rightPlatecol-1+i) + -double(Vpl)/
double(L-botPlaterow);
%end
%may need to superpostion the left and right edges later
%going to ignore it for now

%relax
%secondary matrix to pass new values back and forth with
V2 = zeros(H,L);
Vold = V2;
iterations = 0;

```

```

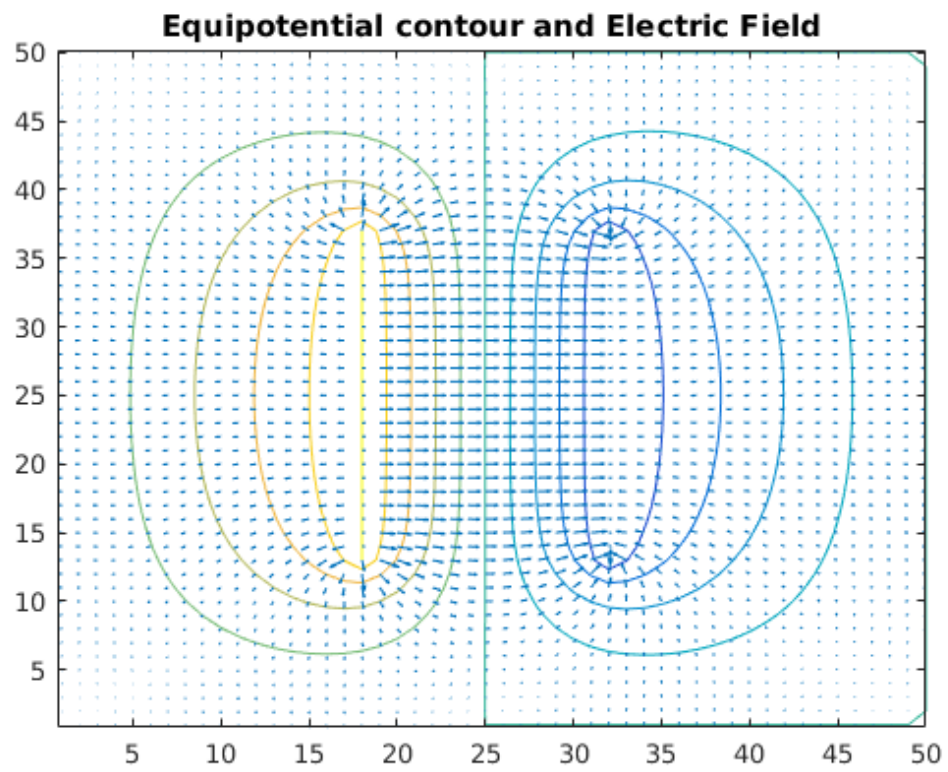
flag = 0;
while( flag == 0 || abs(sum(abs(sum(Vold-V,1)),2)) > 1e-8)
flag= 1;
V2 = V;
    for i = 2:H-1
        for j = 2:L-1
            if(Vp1 ~= abs(V(i,j)) )
                V2(i,j)=(1/4)*(V(i+1,j)+V(i-1,j)+V(i,j+1)+V(i,j-1));
            end
        end
    end

Vold = V;
V=V2;
iterations = iterations +1;
end
iterations
figure();
contour(V);
hold on;
[ex,ey] = gradient(V);
quiver(-ex,-ey);
title('Equipotential contour and Electric Field');
hold off;

```

iterations =

2338



Published with MATLAB® R2015b