UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

SEMESTRE 2023.2

Trabalho sobre Métodos dos Momentos

ALUNO: João Vitor de Oliveira Fraga

MATRÍCULA: 537377

CURSO: Engenharia de Telecomunicações

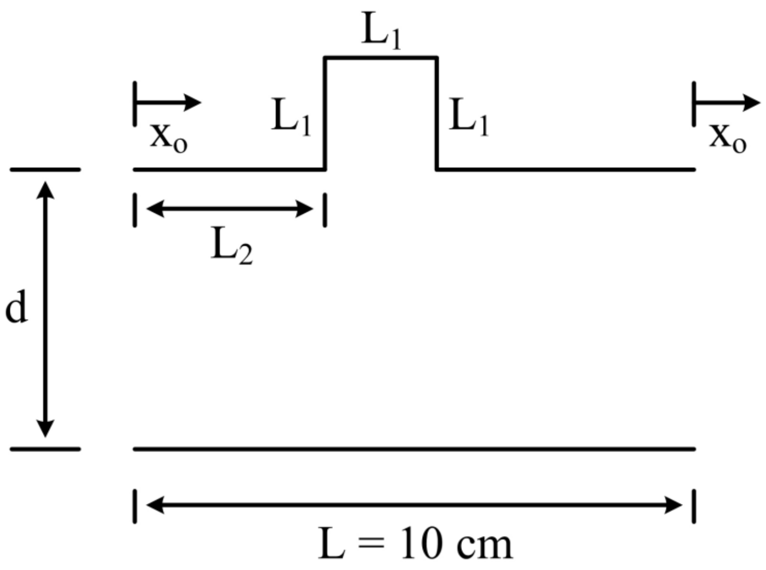
PROFESSOR: Sergio Antenor de Carvalho

QUESTÃO Nº 1

Para a geometria mostrada na figura 1 determine a variação da capacitância para $0 < x_0 \leq$ 5 cm. Nos casos de menor e maior capacitância calcule e analise:

- a distribuição de cargas nos dois condutores;
- a distribuição de potencial e campo elétrico em uma região quadrada de 20 cm de lado em torno do sistema no plano formado pelas linhas.

Figura 1: Par de linhas condutoras



Fonte: Sergio Antenor.

Resposta: Para essa questão, o professor incluiu os valores usados de L_1, d, L_2. No meu caso eu fui o aluno 19, e minhas especificações para a questão foram:

- $L_1 = 0,75cm$
- $d = 7cm$
- $L_2 = 2cm$

Para conseguirmos a imagem que plot a capacitância em função de x_0 primeiros declaramos uma função chamada calculatecapacitance2:

```
function C = calculatecapacitance2(d, l, x0, N, E0)
    RHO = calculateRho2(d, l, x0, N, E0);
```

```
    DL = 1 / N;
    Q = sum(RHO(1:N)) * DL;
    C = abs(Q) / 2;
end
```

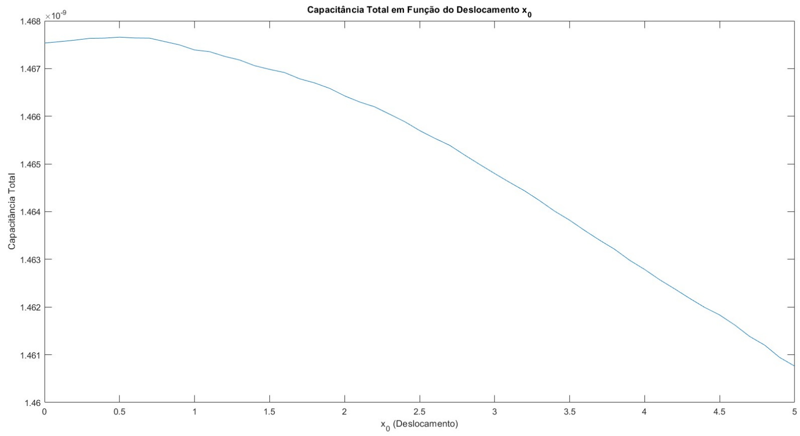
Após declararmos essa função no MatLab, fizemos o seguinte código:

```
ER = 1.0;
E0 = 8.8541e-12;
N = 100; % Número de segmentos em cada fio
V0 = 2.0; % Diferença de potencial

% Define o intervalo para x0
x0_values = 0:0.1:5; % Exemplo: de 0 a 5 com passo de 0.1

% Inicializa as capacit ncias
C_total = zeros(size(x0_values));

% Loop para diferentes valores de x0
for idx = 1:length(x0_values)
    x0 = x0_values(idx);

    % Capacitor 1 (segmento horizontal inicial)
    d1 = 7.0; % Dist ncia entre os fios
    l1 = 2.0+x0; % Comprimento do fio (fixo para a parte horizontal inicial)
    C1 = calculatecapacitance2(d1, l1, x0, N, E0);

    % Capacitor 2 (segmento horizontal final)
    d2 = 7.75; % Dist ncia entre os fios
    l2 = 0.75; % Comprimento do fio (fixo para a parte horizontal final)
    C2 = calculatecapacitance2(d2, l2, x0, N, E0);

    % Capacitor 3 (segmento horizontal final)
    d3 = 7.0; % Dist ncia entre os fios
    l3 = 7.25-x0; % Comprimento do fio (fixo para a parte horizontal final)
    C3 = calculatecapacitance2(d3, l3, x0, N, E0);

```

```
    % Soma as capacit ncias
    C_total(idx) = C1 + C2 + C3;
end

% Exibe a capacit ncia total em função de x0
plot(x0_values, C_total);
xlabel('x_0 (Deslocamento)');
ylabel('Capacit ncia Total');
title('Capacit ncia Total em Função do Deslocamento x_0');
```

Quando adicionamos a função no MatLab e logo em seguida rodamos esse código, obtemos a seguinte imagem:

Figura 2: Par de linhas condutoras



Fonte: Elaborado pelo Autor.

Para calcularmos e plotarmos esse gráfico, é necessário primeiro estabelecer a distribuição de carga no fio. Utilizei uma variação do código disponivel no livro “Elementos de Eletromagnetismo”, livro texto do curso, que calcula a distribuição de carga em fios paralelos, adaptando-o para se adequar à geometria específica deste problema. Comecei desenvolvendo uma função para calcular a distribuição de carga e, em seguida, analisei a variação da capacitância em função de x_0, resultando no gráfico apresentado.

A partir da imagem conseguimos inferir os pontos em que x_0 apresenta uma máxima e minima capacitância, que no caso seriam os pontos $x_0 = 0.5cm$ e $x_0 = 5cm$ respectivamente.

Para calcular a capacitância, dividi o fio em três segmentos. O primeiro vai de 0 à L_2 com altura d, o segundo vai de L_2 à $(L_2 + L_1)$ com uma altura de $L_1 + d$, e o terceiro segmento se estende de $(L_2 + L_1)$ à L com altura d. Ao aplicar tensão no fio superior, o comprimento do fio inferior do capacitor muda para cada segmento: no primeiro, é $L_2 + x_0$, permanece o mesmo no segundo, e no terceiro, é $L - (L_2 + L_1) - x_0$. Há um ponto em que o efeito capacitivo dos segmentos dois e três se torna irrelevante para a capacitância total. No entanto, com um x_0 variando de 0 a 5, esse fator não influencia significativamente, pois o efeito capacitivo do terceiro segmento ainda é considerado.

Agora que temos as informações dos valores de x_0 para maior e menor capacitância, será possível analisar o restante da questão.

Para melhor analise da questão, farei da seguinte maneira: Colocarei todos os códigos, em seguida as imagens plotadas junto da explicação de como foi feito o uso do Método dos Momentos nessa questão e análise dos resultados obtidos .

- Códigos:

Para que eu conseguisse fazer o código declarei algumas funções no próprio MatLab, irei deixar o código das funções aqui para que possa ser adicionada e logo em seguida o script para rodar.

Sobre as funções, salve-as como “calculateElectricField”, “generateWireGeometryWithCharge”, “calculateRho2”e “calculatePotentialForWires”respectivamente.

```
function [Ex, Ey] = calculateElectricField(RHO, x0, x_range, y_range, dx, dy)
% Calcula o potencial elétrico V
V = calculatePotentialForWires(RHO, x0, x_range, y_range, dx, dy);

% Prepara a grade
[X, Y] = meshgrid(min(x_range):dx:max(x_range), min(y_range):dy:max(y_range));

% Inicializa as componentes do campo elétrico
Ex = zeros(size(V));
Ey = zeros(size(V));

% Calcula o campo elétrico usando diferenças finitas
for i = 1:size(V, 1) - 1
    for j = 1:size(V, 2) - 1
```

```
            Ex(i, j) = -(V(i, j+1) - V(i, j)) / dx;
            Ey(i, j) = -(V(i+1, j) - V(i, j)) / dy;
        end
    end

    % Plotando o campo elétrico
    figure;
    quiver(X, Y, Ex, Ey);
    title('Campo Elétrico');
    xlabel('x');
    ylabel('y');
end
```

```
    function generateWireGeometryWithCharge(RHO, x0)
    % Define os par metros
    L = 10;
    d = 7;
    L2 = 2;
    N = length(RHO) / 2; % Número de segmentos em cada fio
    DL = L / N;

    % Cria a figura
    figure;
    hold on;
    axis equal;
    grid on;
    xlabel('x');
    ylabel('y');

    % Desenha o fio inferior com carga
    for i = 1:N
        x = (i - 0.5) * DL;
        scatter(x, 0, 10, RHO(i), 'filled'); % Carga negativa no fio inferior
    end

    % Desenha o fio superior com "dente" e carga
    for i = 1:N
        x = (i - 0.5) * DL;
```

```

        % Calcula a posição do fio superior considerando o "dente"
         if x < 2
            xUpper = x + x0; % Parte horizontal inicial
            yUpper = d; % Parte horizontal inicial
        elseif x < 2.75
            xUpper = 2 + x0; % Parte vertical
            yUpper = d + (x - 2); % Parte vertical
         elseif x<3.5
            xUpper = x - 0.75 + x0; % Parte horizontal final
            yUpper = d + 0.75; % Parte horizontal final
         elseif x < 4.25
            xUpper = 2.75 + x0; % Parte vertical
            yUpper = 1.6*d -x; % Parte vertical
         else
             xUpper = x-1.5 +x0; % Parte horizontal inicial
             yUpper = d; % Parte horizontal inicial
         end

        scatter(xUpper, yUpper, 10, RHO(i+N), 'filled'); % Carga
    positiva no fio superior
    end

    % Limites do gráfico e ajustes finais
    colormap jet; % Define uma escala de cores
    colorbar; % Adiciona uma barra de cores para referência
    xlim([-1, L+1]);
    ylim([-1, d+2]);

    hold off;
end
```

```
    function RHO = calculateRho2(d, l, x0, N, E0)
    NT = 2 * N;
    DL = l / N;
    A = zeros(NT, NT);
    B = ones(NT, 1);
    B(N+1:end) = -1.0;

```

```
    % Define as posições dos segmentos nos fios
    for K = 1:N
        X(K) = (K - 0.5) * DL;
        Y(K) = 0;
        Z(K) = 0;

        % Calcula a posição X para o fio superior considerando o "dente"
        if X(K) < 2
            X(K+N) = X(K) + x0; % Parte horizontal inicial
        elseif X(K) < 2.75
            X(K+N) = 2 + x0; % Parte vertical
        else
            X(K+N) = X(K) - 0.75 + x0; % Parte horizontal final
        end

        % Calcula a posição Y para o fio superior considerando o "dente"
        if X(K) < 2
            Y(K+N) = d; % Parte horizontal inicial
        elseif X(K) < 2.75
            Y(K+N) = d + (X(K) - 2); % Parte vertical
        else
            Y(K+N) = d + 0.75; % Parte horizontal final
        end

        Z(K+N) = 0;
    end

    % Constrói a matriz de coeficientes
    for I = 1:NT
        for J = 1:NT
            if I == J
                A(I, J) = DL * 0.8814 / (pi * E0);
            else
                R = sqrt((X(I) - X(J))^2 + (Y(I) - Y(J))^2 + (Z(I) - Z(J
))^2);
                A(I, J) = DL^2 / (4 * pi * E0 * R);
            end
        end
```

```
    end

    % Calcula a distribuição de carga e a capacit ncia
    F = inv(A);
    RHO = F * B;
end
```

```
    function V = calculatePotentialForWires(RHO, x0, x_range, y_range,
  dx, dy)
    % Constantes
    epsilon0 = 8.854e-12; % Permissividade do vácuo
    L = 10; % Comprimento total dos fios
    d = 7; % Dist ncia vertical entre os fios
    N = length(RHO) / 2; % Número de cargas em cada fio
    DL = L / N; % Comprimento de cada segmento

    % Cria uma grade para calcular o potencial
    [X, Y] = meshgrid(min(x_range):dx:max(x_range), min(y_range):dy:max(
  y_range));
    V = zeros(size(X)); % Inicializa a matriz de potencial

    % Loop sobre as cargas no fio inferior
    for i = 1:N
        x_charge = (i - 0.5) * DL;
        y_charge = 0;
        r = sqrt((X - x_charge).^2 + (Y - y_charge).^2);
        V = V + RHO(i) ./ (4 * pi * epsilon0 * r);
    end

    % Loop sobre as cargas no fio superior (com a geometria específica)
    for i = 1:N
        x = (i - 0.5) * DL;

        % Define a posição y do fio superior, com base na geometria dada
        if x < 2
            xUpper = x + x0; yUpper = d;
        elseif x < 2.75
            xUpper = 2 + x0; yUpper = d + (x - 2);
        elseif x < 3.5
```

```
            xUpper = x - 0.75 + x0; yUpper = d + 0.75;
        elseif x < 4.25
            xUpper = 2.75 + x0; yUpper = 1.6 * d - x;
        else
            xUpper = x - 1.5 + x0; yUpper = d;
        end

        r = sqrt((X - xUpper).^2 + (Y - yUpper).^2);
        V = V + RHO(i + N) ./ (4 * pi * epsilon0 * r);
    end

    % Lidar com singularidades
    V(isinf(V)) = NaN;

    % Plotando o potencial
    figure;
    surf(X, Y, V);
    shading interp;
    colorbar;
    title('Distribuição de Potencial');
    xlabel('x');
    ylabel('y');
    zlabel('Potencial (V)');
end
```

Agora que temos todas as funções já declaradas no nosso programa, vamos para o código que será rodado:

```

% agora vemos com a distribuição de carga nos fios
% x0= 0.5 já que foi o observado na questão como maior capacit ncia
x0 = 0.5; % Valor de exemplo para x0
d = 7; % Dist ncia entre os fios
l = 10.0; % Comprimento do fio
N = 100; % Maior número de segmentos para melhor resolução
E0 = 8.8541e-12;

RHO = calculateRho2(d, l, x0, N, E0);

```

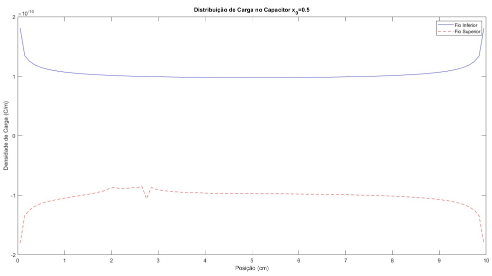
```
% Posições dos segmentos
DL = l / N;
X = (DL/2):DL:l;

% Plot
figure;
plot(X, RHO(1:N), 'b-', X, RHO(N+1:end), 'r--');
legend('Fio Inferior', 'Fio Superior');
xlabel('Posição (m)');
ylabel('Densidade de Carga (C/m)');
title('Distribuição de Carga no Capacitor x_0=0.5');

% x0= 5 já que foi o observado na questão como menor capacit ncia
x0 = 5; % Valor de exemplo para x0
d = 7; % Dist ncia entre os fios
l = 10.0; % Comprimento do fio
N = 100; % Maior número de segmentos para melhor resolução
E0 = 8.8541e-12;

RHO = calculateRho2(d, l, x0, N, E0);

% Posições dos segmentos
DL = l / N;
X = (DL/2):DL:l;

% Plot
figure;
plot(X, RHO(1:N), 'b-', X, RHO(N+1:end), 'r--');
legend('Fio Inferior', 'Fio Superior');
xlabel('Posição (m)');
ylabel('Densidade de Carga (C/m)');
title('Distribuição de Carga no Capacitor x_0=5');

%Agora para p na geometria
x0=0.5
generateWireGeometry(RHO, x0);

x0=5;
```

```
generateWireGeometry(RHO, x0);
%%
%por fim potencial e campo elétrico
%x0=0.5
x_range = [-5, 15];
y_range = [-10, 10];
dx = 0.1;
dy = 0.1;
x0= 0.5;
calculatePotentialForWires(RHO, x0, x_range, y_range, dx, dy);
calculateElectricField(RHO, x0, x_range, y_range, dx, dy);


%%
%x0=5
x_range = [-5, 15];
y_range = [-10, 10];
dx = 0.1;
dy = 0.1;
x0= 5;
calculatePotentialForWires(RHO, x0, x_range, y_range, dx, dy);
calculateElectricField(RHO, x0, x_range, y_range, dx, dy);
```

- Plotagem + Explicação:

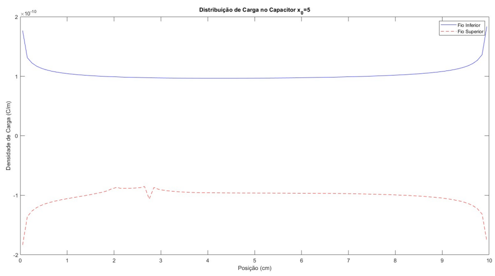
Com todas as informações do código já informadas, conseguimos agorar plotar as figuras e fazer a explicação pedida.

Após analisarmos os valores de x_0, plotamos as imagens com a distribuição de carga:



(a) Distribuição de carga em $x_0 = 0.05cm$

Fonte: Elaborado pelo autor.



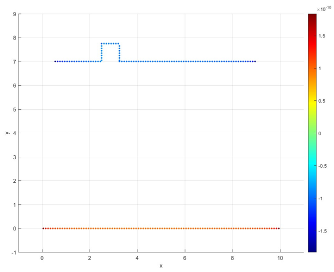
(b) Distribuição de carga em $x_0 = 5cm$

Fonte: Elaborado pelo autor.

Quando analisamos o fio inferior dos dois casos, percebemos que os mesmos tem praticamente o mesmo comportamento, sem conseguir inferir de forma visual a diferença, contudo, uma análise detalhada, avaliando ponto a ponto, revela variações significativas na distribuição de carga ao longo do fio. Em contrapartida o fio superior apresenta uma diferença mais notória, para um valor de $x_0 = 5$, observamos uma distribuição de carga mais uniforme ao longo do fio superior, com uma concentração mais acentuada de cargas nas extremidades. Por outro lado, para $x_0 = 0.05$, a distribuição se torna mais assimétrica, com uma acumulação de carga de menor intensidade.

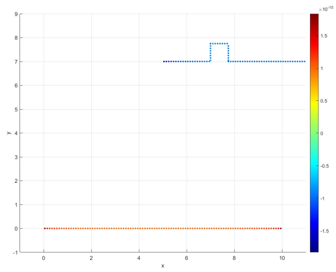
Após isso foi necessário fazer a geometria para que conseguissemos calcular ponto a ponto tanto o Potêncial Elétrico como o Campo Elétrico. Inicialmente, definimos a geometria e, para cada ponto, atribuímos o valor de ρ correspondente.

Ao gerar os gráficos, enfrentamos dificuldades. Devido à mesma discretização utilizada para ambos os fios, surgiu um problema: o fio superior parecia mais curto do que deveria. Essa percepção resultou da igualdade na discretização. Ao definir ponto a ponto, o fio superior, que deveria ser mais longo, acabou tendo o mesmo comprimento que o fio inferior. Isso criou a impressão de uma diferença no comprimento, mas essa discrepância não impacta significativamente nossos cálculos. No fio superior, onde há o "dente", a carga correspondente está presente. Idealmente, deveríamos ter aplicado uma discretização mais fina para o fio superior para ajustar o comprimento. Contudo, tal ação não afetaria de forma significativa o resultado.



(a) Geometria para $x_0 = 0.05cm$

Fonte: Elaborado pelo autor.

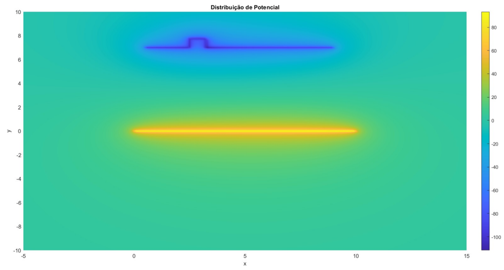


(b) Geometria para $x_0 = 5cm$

Fonte: Elaborado pelo autor.

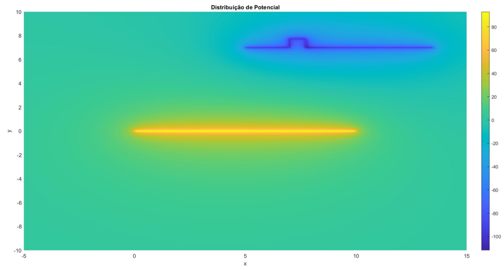
Agora para que consigamos calcular o potêncial dos fios, foi utilizado as mesmas funções que geraram os gráficos mostrados anteriormente. Logo conseguimos as seguintes distribuições de potêncial:

A análise dos gráficos revela que o potencial sofre uma variação notável na presença do "dente"no fio superior. No fio inferior, essa variação não ocorre. O motivo para a variação no



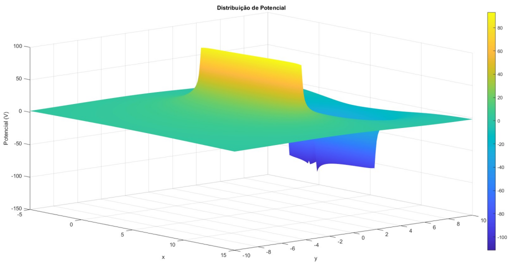
(a) Potencial em $x_0 = 0.05cm$

Fonte: Elaborado pelo autor.



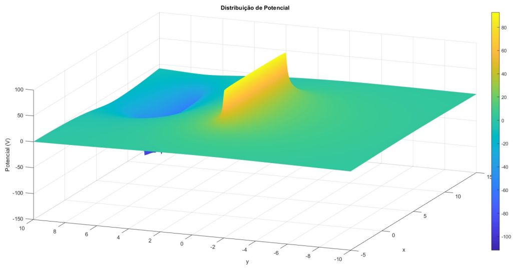
(b) Potencial em $x_0 = 5cm$

Fonte: Elaborado pelo autor.



(c) Potencial em 3D em $x_0 = 0.05cm$

Fonte: Elaborado pelo autor.

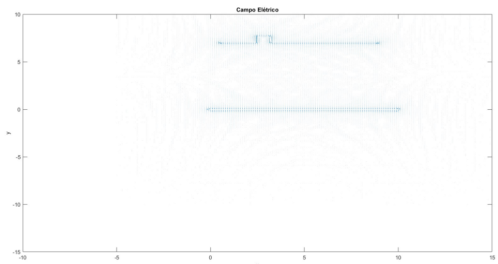


(d) Potencial em 3D em $x_0 = 5cm$

Fonte: Elaborado pelo autor.

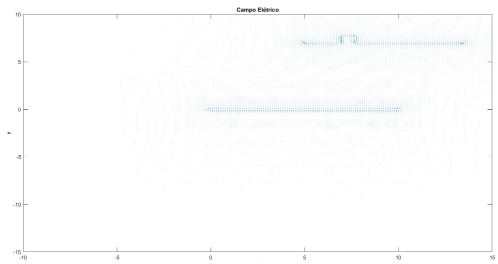
fio superior é o acúmulo, ainda que mínimo, de cargas no dente. Essa pequena alteração no potencial é sutil e só pode ser observada ao interagirmos diretamente com o gráfico, como por exemplo, ao clicar nele para uma análise mais detalhada.

Já para obter o Campo Elétrico, foi criada uma função que usa o gradiente, obtendo então os seguintes gráficos:



(a) Campo Elétrico para $x_0 = 0.05cm$

Fonte: Elaborado pelo autor.



(b) Campo Elétrico para $x_0 = 5cm$

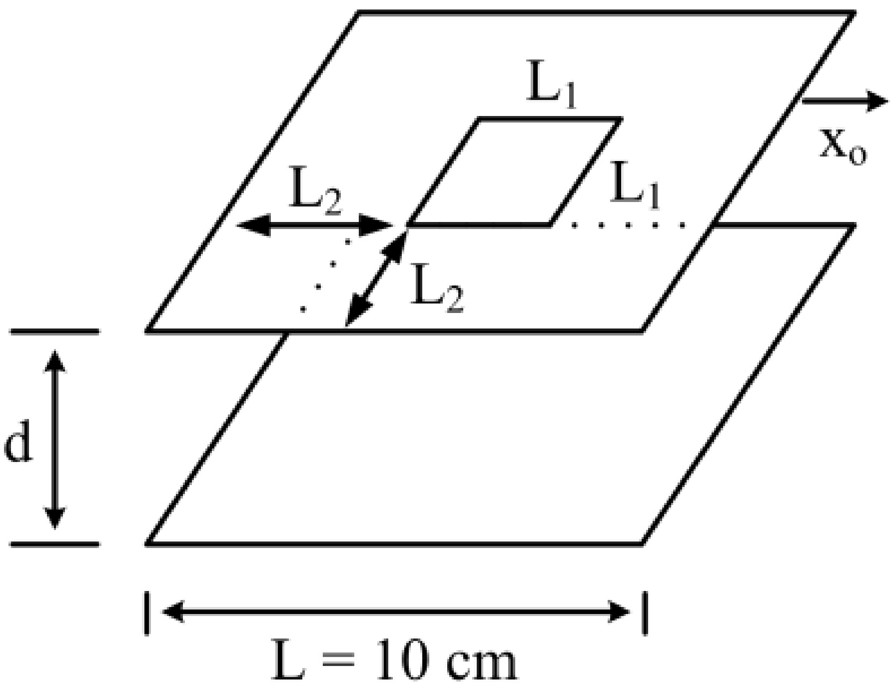
Fonte: Elaborado pelo autor.

QUESTÃO Nº 2

Para a geometria mostrada na Figura 7 determine a variação da capacitância para $0 < x_0 \leq 5$ cm. Nos casos de menor e maior capacitância calcule e analise:

- a distribuição de cargas nos dois condutores;
- a distribuição de potencial e campo elétrico em uma região quadrada de 20 cm de lado em torno do sistema no plano formado pelas linhas.

Figura 7: Par de placas condutoras



Fonte: Sergio Antenor.

Resposta: Para essa questão, o professor incluiu os valores usados de L_1, d, L_2. No meu caso eu fui o aluno 19, e minhas especificações para a questão foram:

- $L_1 = 2.5cm$
- $d = 5.0cm$
- $L_2 = 4.5cm$

Diferentemente da primeira questão, essa apresenta uma maior complexidade, devido a sua natureza tridimensional, aumentando a dificuldade em relação a primeiro questão que era apenas bidimensional.

O primeiro passo foi consultar o livro para entender como o método dos momentos é aplicado a capacitores de placas paralelas. No livro, encontramos um exemplo que foca apenas no cálculo da capacitância. A partir desse código, realizei adaptações, incluindo uma modificação significativa.

Inseri no código as coordenadas do vértice do buraco na placa em relação à origem. Inicialmente, tentei definir a distância entre os vértices, mas isso levou a erros. Depois de várias tentativas, percebi que, ao definir a distância do vértice, estava na verdade especificando a coordenada do vértice do buraco. No meu caso, com $L = 4.5cm$, o vértice estava a uma distância de aproximadamente $6.36cm$, o que se tornou um grande obstáculo na resolução do problema.

Para calcular a capacitância, utilizei a base do código do livro, mas com modificações para tratar o caso específico de um capacitor de placas paralelas com um buraco em uma das placas. Apliquei um loop no código para remover os pontos correspondentes ao buraco na placa.

Optei por uma discretização relativamente alta. Como o problema agora é bidimensional, a resolução da matriz MxM difere do caso unidimensional. Usei uma discretização de $N = 400$ e defini $M = N$, o que simplificou um pouco o problema. Ao dividir a placa em uma matriz $20x20$, muitos pontos precisavam ser calculados. Sem essa "simplificação", a quantidade de pontos seria ainda maior, tornando a matriz mais densa e complexa para ser preenchida e invertida para encontrar o ρ.

Defini a seguinte função no MatLab chamada "calcularCapacitancia".

```
function [C, RHO] = calcularCapacitancia(x0)
    % Constantes
    ER = 1.0;
    E0 = 8.8541e-12;
    ladoPlaca = 0.1; % 10 cm
    distanciaPlacas = 0.05; % 7 cm
    N = 400;
    NT = 2 * N;
    M = sqrt(N);
    DX = ladoPlaca / M;
    DY = ladoPlaca / M;
    DL = DX;

    % Par metros do buraco
    ladoBuraco = 0.002025; % 2.5 cm
```

```
    distanciaVertice = 2.5/ 100; % 5  2   cm em metros

    % Inicialização das matrizes e vetores
    A = zeros(NT, NT);
    B = zeros(NT, 1);
    X = zeros(NT, 1);
    Y = zeros(NT, 1);
    Z = zeros(NT, 1);

    % Preenchimento de X, Y, Z com deslocamento na placa superior
    K = 0;
    for K1 = 1:2
        for K2 = 1:M
            for K3 = 1:M
                K = K + 1;
                if K1 == 1
                    % Placa inferior
                    X(K) = DX * (K2 - 0.5);
                    Y(K) = DY * (K3 - 0.5);
                else
                    % Placa superior com deslocamento x0
                    X(K) = DX * (K2 - 0.5) + x0;
                    Y(K) = DY * (K3 - 0.5);
                end
            end
        end
    end
    Z(1:N) = 0.0;
    Z(N+1:end) = distanciaPlacas;

    % Cálculo da matriz A e vetor B
  for I = 1:NT
        for J = 1:NT
            if I == J
                A(I, J) = DL * 0.8814 / (pi * E0);
            else
                R = sqrt((X(I) - X(J))^2 + (Y(I) - Y(J))^2 + (Z(I) - Z(J
 ))^2);
```

```
                A(I, J) = DL^2 / (4 * pi * E0 * R);
            end
        end
        % Verificando se o ponto está dentro do buraco
        if X(I) > distanciaVertice && X(I) < distanciaVertice + ladoBuraco && Y(I) > distanciaVertice && Y(I) < distanciaVertice + ladoBuraco
            B(I) = 0;
        else
            B(I) = I <= N;
        end
    end
    % Calculando    e Capacit ncia
    F = inv(A);
    RHO = F * B;
    Q = sum(RHO(1:N)) * (DL^2);
    V0 = 2.0;
    C = abs(Q) / V0;
end
```

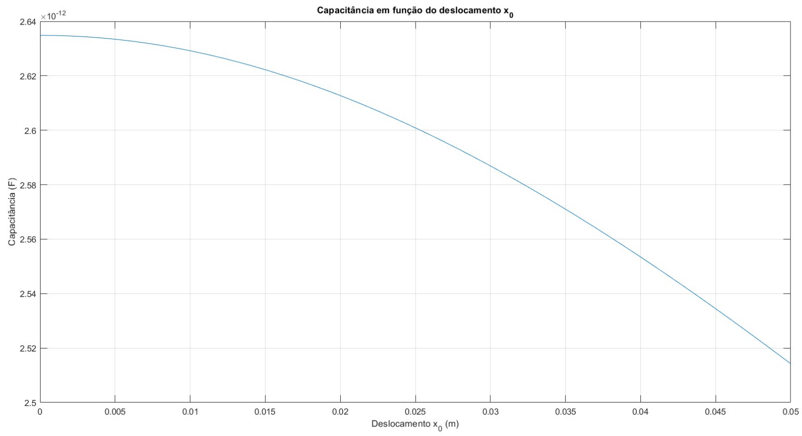
Após declararmos essa função no MatLab, fizemos o seguinte código:

```
x0_min = 0; % Deslocamento mínimo em metros
x0_max = 0.05; % Deslocamento máximo em metros
num_pontos = 50; % Número de pontos no intervalo

% Vetores para armazenar valores de x0 e capacit ncias
x0_valores = linspace(x0_min, x0_max, num_pontos);
capacitancias = zeros(1, num_pontos);

% Cálculo da capacit ncia para cada valor de x0
for i = 1:num_pontos
    [C, ~] = calcularCapacitancia(x0_valores(i));
    capacitancias(i) = C;
end

% Plotando os resultados
figure;
plot(x0_valores, capacitancias);
```

```
xlabel('Deslocamento x_0 (m)');
ylabel('Capacit ncia (F)');
title('Capacit ncia em função do deslocamento x_0');
grid on;
```

Quando adicionamos a funcão no MatLab e logo em seguida rodamos esse código, obtemos a seguinte imagem:

Figura 8: Capacitância em relação à x_0



Fonte: Elaborado pelo Autor.

A partir dela conseguimos inferir os pontos em que x_0 apresenta uma máxima e minima capacitância, que no caso seriam os pontos $x_0 = 0cm$ e $x_0 = 5cm$ respectivamente. Tal comportamento era esperado, pois diferente do primeiro problema onde tivemos que dividir o capacitor em 3, já aqui a melhor estratégia envolve considerar as placas como paralelas e ajustar a geometria para levar em conta o efeito capacitivo resultante da presença do buraco.

Agora que temos as informações dos valores de x_0 para maior e menor capacitância, será possível analisar o restante da questão.

Para melhor analise da questão, farei da seguinte maneira: Colocarei todos os códigos, em seguida as imagens plotadas junto da explicação de como foi feito o uso do Método dos Momentos nessa questão e análise dos resultados obtidos.

- Códigos:

Para que eu conseguisse fazer o código declarei algumas funções no próprio MatLab, irei deixar o código das funções aqui para que possa ser adicionada e logo em seguida o script para rodar.

Salve a função como “calcularDistribuicaoDeCarga”.

```
function [rho_placa1, rho_placa2] = calcularDistribuicaoDeCarga(x0)
% Constantes
ER = 1.0;
E0 = 8.8541e-12;
ladoPlaca = 0.1; % 10 cm
distanciaPlacas = 0.5; % 7 cm
N = 400;
NT = 2 * N;
M = sqrt(N);
DX = ladoPlaca / M;
DY = ladoPlaca / M;
DL = DX;

% Par metros do buraco
ladoBuraco = 0.045; %
% Centralizando o buraco na placa
distanciaVertice = 2.5/100;

% Inicialização das matrizes e vetores
A = zeros(NT, NT);
B = zeros(NT, 1);
X = zeros(NT, 1);
Y = zeros(NT, 1);
Z = zeros(NT, 1);

% Preenchimento de X, Y, Z com deslocamento na placa superior
K = 0;
for K1 = 1:2
    for K2 = 1:M
        for K3 = 1:M
            K = K + 1;
            if K1 == 1
                % Placa inferior
                X(K) = DX * (K2 - 0.5);
                Y(K) = DY * (K3 - 0.5);
            else
                % Placa superior com deslocamento x0
```

```
                    X(K) = DX * (K2 - 0.5) + x0;
                    Y(K) = DY * (K3 - 0.5);
                end
            end
        end
    end
    Z(1:N) = 0.0;
    Z(N+1:end) = distanciaPlacas;

    % Cálculo da matriz A e vetor B
    for I = 1:NT
        for J = 1:NT
            if I == J
                A(I, J) = DL * 0.8814 / (pi * E0);
            else
                R = sqrt((X(I) - X(J))^2 + (Y(I) - Y(J))^2 + (Z(I) - Z(J))^2);
                A(I, J) = DL^2 / (4 * pi * E0 * R);
            end
        end
        % Verificando se o ponto está dentro do buraco
        if X(I) > distanciaVertice && X(I) < distanciaVertice + ladoBuraco && Y(I) > distanciaVertice && Y(I) < distanciaVertice + ladoBuraco
            B(I) = 0;
        else
            B(I) = I <= N; % Ajuste na atribuição de valores para B
        end
    end

    % Calculando    e Capacit ncia
    F = inv(A);
    RHO = F * B;
    rho_placa1 = RHO(1:N); % Densidade de carga na placa inferior
    rho_placa2 = RHO(N+1:end); % Densidade de carga na placa superior
% Preparando dados para plotagem
    [X_grid, Y_grid] = meshgrid(linspace(0, ladoPlaca, M), linspace(0, ladoPlaca, M));
```

```
    rho_plot1 = reshape(rho_placa1, [M, M]);
    rho_plot2 = reshape(rho_placa2, [M, M]);

    % Plotagem da distribuição de carga
    figure;
    subplot(1, 2, 1);
    surf(X_grid, Y_grid, rho_plot1);
    title('Distribuição de Carga na Placa Inferior');
    xlabel('X (m)');
    ylabel('Y (m)');
    zlabel('Densidade de Carga (C/m^2)');

    subplot(1, 2, 2);
    surf(X_grid, Y_grid, rho_plot2);
    title('Distribuição de Carga na Placa Superior');
    xlabel('X (m)');
    ylabel('Y (m)');
    zlabel('Densidade de Carga (C/m^2)');

    % Retornando dados para plotagem externa, se necessário
    X_plot = X_grid;
    Y_plot = Y_grid;
    rho_plot1 = rho_plot1;
    rho_plot2 = rho_plot2;
end
```

Após declarar todas as nossas funções no programa, vamos para o código que será rodado:

```
    %Agora o cálculo da distribuição de carga com o x0= 0 e x0=5cm
calcularDistribuicaoDeCarga(0);
calcularDistribuicaoDeCarga(0.05);
%%
% Potencial e campo elétrico para x0=0
epsilon0 = 8.8541e-12; % Permissividade do vácuo
ladoPlaca = 0.1; % Lado da placa (10 cm)

% Calculando a distribuição de carga
x0 = 0; % Exemplo de deslocamento de 2 cm
[rho_inferior, rho_superior] = calcularDistribuicaoDeCarga(x0);
```

```

% Número de pontos em cada dimensão da placa
M = sqrt(length(rho_inferior));

% Redimensionando os vetores de rho para matrizes 2D
rho_matriz_inferior = reshape(rho_inferior, M, M);
rho_matriz_superior = reshape(rho_superior, M, M);

% Criando uma grade para cálculo do potencial
[X, Y] = meshgrid(linspace(0, ladoPlaca, M), linspace(0, ladoPlaca, M));
Z = 0.01; % Altura fixa acima das placas para cálculo do potencial

% Inicializando o potencial elétrico
V = zeros(size(X));

% Calculando o potencial elétrico
for i = 1:M
    for j = 1:M
        r_inferior = sqrt((X - X(i, j)).^2 + (Y - Y(i, j)).^2 + Z^2);
        V = V + rho_matriz_inferior(i, j) ./ (4 * pi * epsilon0 * r_inferior);
        r_superior = sqrt((X - (X(i, j) + x0)).^2 + (Y - Y(i, j)).^2 + Z^2);
        V = V + rho_matriz_superior(i, j) ./ (4 * pi * epsilon0 * r_superior);
    end
end
% Calculando o campo elétrico a partir do potencial
[Ex, Ey] = gradient(-V);

% Plotando o campo elétrico
figure;
quiver(X, Y, Ex, Ey);
title('Campo Elétrico Acima das Placas x_0=0');
xlabel('x (m)');
ylabel('y (m)');
axis tight;
% Plotando o potencial elétrico
```

```
figure;
surf(X, Y, V);
title('Potencial Elétrico Acima das Placas x_0=0');
xlabel('x (m)');
ylabel('y (m)');
zlabel('Potencial Elétrico (V)');
colorbar; % Adiciona uma barra de cores para representar a variação do
    potencial
shading interp; % Para uma visualização mais suave
%%
% Potencial e campo elétrico x0=5cm
epsilon0 = 8.8541e-12; % Permissividade do vácuo
ladoPlaca = 0.1; % Lado da placa (10 cm)

% Calculando a distribuição de carga
x0 = 0.05; % Exemplo de deslocamento de 2 cm
[rho_inferior, rho_superior] = calcularDistribuicaoDeCarga(x0);

% Número de pontos em cada dimensão da placa
M = sqrt(length(rho_inferior));

% Redimensionando os vetores de rho para matrizes 2D
rho_matriz_inferior = reshape(rho_inferior, M, M);
rho_matriz_superior = reshape(rho_superior, M, M);

% Criando uma grade para cálculo do potencial
[X, Y] = meshgrid(linspace(0, ladoPlaca, M), linspace(0, ladoPlaca, M));
Z = 0.01; % Altura fixa acima das placas para cálculo do potencial

% Inicializando o potencial elétrico
V = zeros(size(X));

% Calculando o potencial elétrico
for i = 1:M
    for j = 1:M
        r_inferior = sqrt((X - X(i, j)).^2 + (Y - Y(i, j)).^2 + Z^2);
        V = V + rho_matriz_inferior(i, j) ./ (4 * pi * epsilon0 *
    r_inferior);
```

```
        r_superior = sqrt((X - (X(i, j) + x0)).^2 + (Y - Y(i, j)).^2 + Z^2);
        V = V + rho_matriz_superior(i, j) ./ (4 * pi * epsilon0 * r_superior);
    end
end
% Calculando o campo elétrico a partir do potencial
[Ex, Ey] = gradient(-V);

% Plotando o campo elétrico
figure;
quiver(X, Y, Ex, Ey);
title('Campo Elétrico Acima das Placas x_0=5cm');
xlabel('x (m)');
ylabel('y (m)');
axis tight;
% Plotando o potencial elétrico
figure;
surf(X, Y, V);
title('Potencial Elétrico Acima das Placas x_0=5cm');
xlabel('x (m)');
ylabel('y (m)');
zlabel('Potencial Elétrico (V)');
colorbar; % Adiciona uma barra de cores para representar a variação do potencial
shading interp; % Para uma visualização mais suave
%%
% Potencial e campo eletrico perpendicular x0= 5cm

% Par metros do problema
epsilon0 = 8.8541e-12; % Permissividade do vácuo
L = 200; % Dimensão da grade
x0 = 5 / 100; % Deslocamento x0 em metros (5 cm)

% Inicializando a matriz de potencial
U = zeros(L, L);

% Calculando a densidade de carga rho
```

```
[rho_placa1, rho_placa2] = calcularDistribuicaoDeCarga(x0);

% A densidade de carga é convertida em potencial
potencial_placa1 = 220; % Potencial arbitrário para placa 1
potencial_placa2 = -220; % Potencial arbitrário para placa 2

% Posicionamento das placas na matriz de potencial
placa1_y = L/2 - 40;
placa2_y = L/2 + 40;
placa_x_start = L/2 - 80;
placa_x_end = L/2 + 80;

% Convertendo x0 para índice na matriz
x0_index = round(x0 * L / (0.1 * 2)); % Supondo que o tamanho total é
    0.1*2 metros

% Aplicando potencial às placas
U(placa1_y, placa_x_start:placa_x_end) = potencial_placa1;
U(placa2_y, (placa_x_start + x0_index):(placa_x_end + x0_index)) =
    potencial_placa2;

% Método de relaxamento para calcular o potencial no espaço
for iter = 1:1000
    for i = 2:L-1
        for j = 2:L-1
            if U(i, j) == potencial_placa1 || U(i, j) ==
    potencial_placa2
                continue; % Mantendo o potencial nas placas
            end
            U(i, j) = (U(i+1, j) + U(i-1, j) + U(i, j+1) + U(i, j-1)) /
    4;
        end
    end
end

% Plotando o potencial elétrico
figure;
surf(U);
```

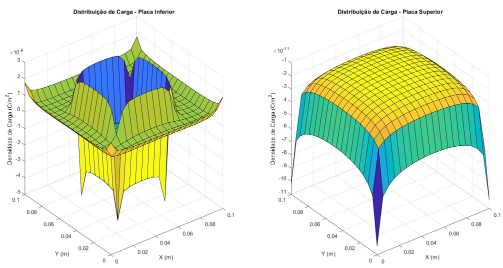
```
shading interp;
colorbar;
xlabel('Lx');
ylabel('Ly');
zlabel('Potencial Elétrico, em Volts');
title('Distribuição de Potencial Elétrico x_0=5');

% Calculando e plotando o campo elétrico
[Ex, Ey] = gradient(-U);
figure;
contour(U, 'LineWidth', 2);
hold on;
quiver(Ex, Ey, 4);
hold off;
title('Campo Elétrico x_0=5');
%%
% Potencial e campo elétrico x0=0
epsilon0 = 8.8541e-12; % Permissividade do vácuo
L = 200; % Dimensão da grade
x0 = 0; % Deslocamento x0 em metros

% Inicializando a matriz de potencial
U = zeros(L, L);

% Calculando a densidade de carga rho
[rho_placa1, rho_placa2] = calcularDistribuicaoDeCarga(x0);

% A densidade de carga é convertida em potencial
potencial_placa1 = 220; % Potencial arbitrário para placa 1
potencial_placa2 = -220; % Potencial arbitrário para placa 2

% Posicionamento das placas na matriz de potencial
placa1_y = L/2 - 40;
placa2_y = L/2 + 40;
placa_x_start = L/2 - 80;
placa_x_end = L/2 + 80;

% Convertendo x0 para índice na matriz
```

```
x0_index = round(x0 * L / (0.1 * 2)); % Supondo que o tamanho total é
    0.1*2 metros

% Aplicando potencial às placas
U(placa1_y, placa_x_start:placa_x_end) = potencial_placa1;
U(placa2_y, (placa_x_start + x0_index):(placa_x_end + x0_index)) =
    potencial_placa2;

% Método de relaxamento para calcular o potencial no espaço
for iter = 1:1000
    for i = 2:L-1
        for j = 2:L-1
            if U(i, j) == potencial_placa1 || U(i, j) ==
    potencial_placa2
                continue; % Mantendo o potencial nas placas
            end
            U(i, j) = (U(i+1, j) + U(i-1, j) + U(i, j+1) + U(i, j-1)) /
    4;
        end
    end
end

% Plotando o potencial elétrico
figure;
surf(U);
shading interp;
colorbar;
xlabel('Lx');
ylabel('Ly');
zlabel('Potencial Elétrico, em Volts');
title('Distribuição de Potencial Elétrico x_0=0');

% Calculando e plotando o campo elétrico
[Ex, Ey] = gradient(-U);
figure;
contour(U, 'LineWidth', 2);
hold on;
quiver(Ex, Ey, 4);
```

```
hold off;
title('Campo Elétrico x_0=0');
```

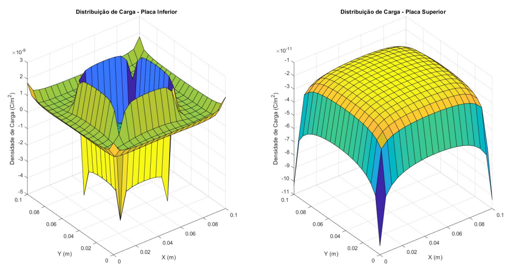
- Plotagem + Explicação:

Com todas as informações do código já informadas, conseguimos agorar plotar as figuras e fazer a explicação pedida.



(a) Destribuição de Carga para $x_0 = 0cm$

Fonte: Elaborado pelo autor.



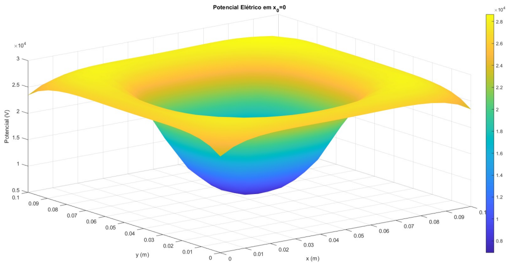
(b) Destribuição de Carga para $x_0 = 5cm$

Fonte: Elaborado pelo autor.

Inicialmente, a similaridade entre os gráficos pode sugerir que não há diferenças significativas entre a placa com e sem buraco. No entanto, uma inspeção mais detalhada, especialmente da escala, indica mudanças relevantes. Na placa sem buraco, a variação na escala é um indicativo de alterações, enquanto na placa com buraco, nota-se uma mudança no acúmulo de cargas tanto nas bordas do buraco quanto nos vértices da placa. O fato de que a magnitude da carga é maior para $x_0 = 0cm$ do que para $x_0 = 5cm$ valida a eficácia do código.

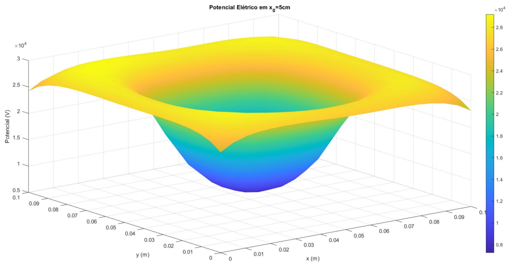
O desafio mais significativo surgiu na etapa de modelagem da distribuição de potencial e do campo elétrico em três dimensões, abordando o problema sob diferentes perspectivas. Optei por uma estratégia que não necessitava da representação completa do potencial ou do campo elétrico em 3D. Em vez disso, concentrei-me em planos específicos, como os localizados em $z = 0cm$ e $z = d$. Esta abordagem me permitiu integrar a carga e visualizar de forma eficaz tanto o potencial quanto o campo elétrico nesses planos específicos. Escolhi o plano com o buraco para a plotagem, pois oferecia uma clareza maior nas diferenças de potencial e campo elétrico, ilustrando assim de forma mais vívida as nuances do fenômeno estudado.

Ao analisarmos os gráficos de Potencial Elétrico, notamos imediatamente uma alteração significativa quando x_0 é igual a 5 cm. Essa mudança ocorre devido ao cálculo do potencial baseado na distribuição de carga. Mesmo uma pequena variação na carga resulta em uma mudança



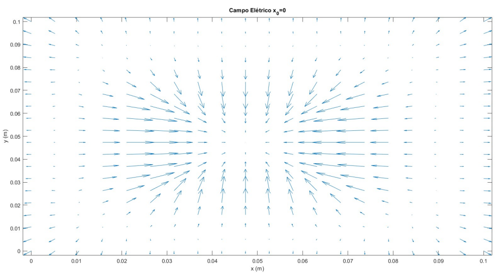
(a) Destribuição de Potencial Acima das Placas para $x_0 = 0cm$

Fonte: Elaborado pelo autor.



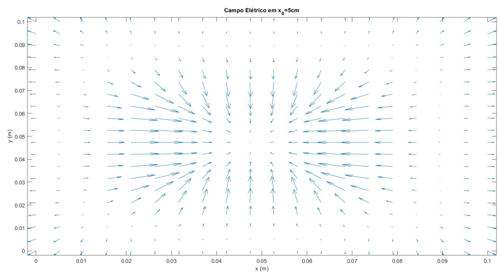
(b) Destribuição de Potencial Acima das Placas para $x_0 = 5cm$

Fonte: Elaborado pelo autor.



(c) Campo Elétrico Acima das Placas para $x_0 = 5cm$

Fonte: Elaborado pelo autor.



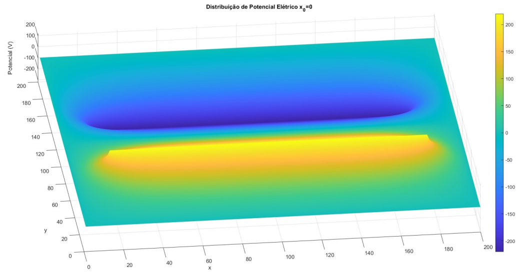
(d) Campo Elétrico Acima das Placas para $x_0 = 5cm$

Fonte: Elaborado pelo autor.

considerável no potencial, especialmente nas áreas onde a simetria entre as placas é interrompida. Por exemplo, em x_0, as placas mantêm uma simetria relativa, mas com a introdução de um x_0 na placa com o buraco, observamos mudanças substanciais na capacitância e na carga. Como o potencial é calculado a partir da carga, ele também sofre alterações. Consequentemente, nas regiões da placa onde a simetria com a outra placa é perdida, há uma redução notável no potencial, refletindo uma diminuição no acúmulo de cargas.

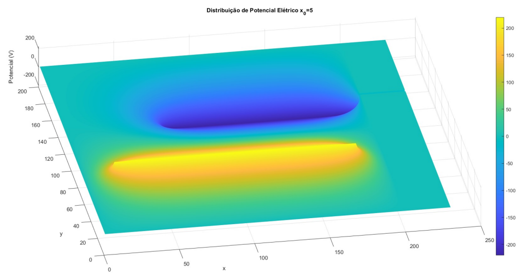
Quanto ao campo elétrico, percebe-se que não há grandes mudanças em sua configuração, o que é esperado, visto que estamos observando o campo a partir de cima da placa. As variações mais significativas ocorrem na intensidade do campo, decorrentes da queda no potencial entre um gráfico e outro.

Para obter uma visão diferente das placas, explorei diversas abordagens. Para alcançar essa nova perspectiva, foi necessário utilizar um código diferente, aproveitando o meu ρ como base. Esse novo método permitiu mudar a perspectiva de análise, mantendo os parâmetros utilizados anteriormente, como a discretização, por exemplo. Essa abordagem permitiu uma compreensão mais ampla e detalhada do problema, oferecendo insights adicionais sobre o comportamento das placas sob diferentes condições.



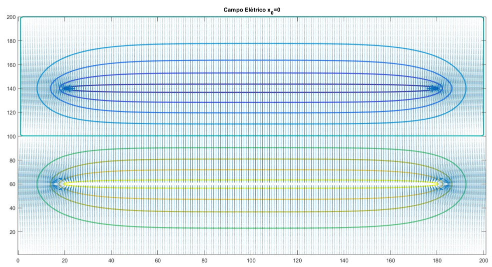
(a) Potêncial Elétrico para $x_0 = 0cm$

Fonte: Elaborado pelo autor.



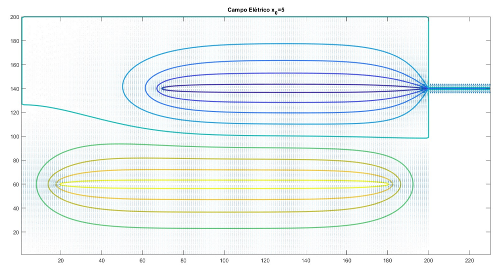
(b) Potêncial Elétrico para $x_0 = 5cm$

Fonte: Elaborado pelo autor.



(c) Campo Elétrico para $x_0 = 5cm$

Fonte: Elaborado pelo autor.



(d) Campo Elétrico para $x_0 = 5cm$

Fonte: Elaborado pelo autor.

Conclusão

Ao final deste trabalho concluímos que o método dos momentos se mostra extremamente eficaz na aplicação à equação de Poisson, fornecendo uma distribuição de carga precisa para a estrutura em análise. Levando mais de mês para que fosse possível finalizar esse trabalho fica evidente que o método dos momentos é uma ferramenta valiosa para esse tipo de problema. No entanto, a grande dificuldade reside em representar a geometria do problema.

De forma resumida, foi bastante árduo, complexo e enriquecedor, mas com resultados sastisfátorios, apesar de ao final utilizarmos outros métodos para concluir a questão.