



ESCOLA POLITÉCNICA DA USP
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA
E DE SISTEMAS MECÂNICOS

Mecânica Computacional
PMR3401

Exercício Programa EXTRA: Método de Elementos
Finitos(MEF)
07/2020

Alessandro Brugnera Silva – 10334040
Vitor Luiz Lima Carazzi – 9834010

Sumário

Sumário	2
Introdução	3
Problema	3
Equacionamentos	4
Equações dos Fenômenos Físicos	4
Equações do Método de Elementos Finitos	4
Resultados	6
1.a) Escalares $V(x,y)$ e $A_z(x,y)$	7
1.b) Vetores Densidade de fluxo magnético, Intensidade de campo magnético e Intensidade de campo elétrico	9
Sobreposição de grandezas vetoriais e escalares	12
Fenômenos elétricos	12
Fenômenos Magnéticos	13
Conclusão Final	14
Códigos	15
main	15
Auxiliares	23
criaNos	23
derivaElemento	25
matrizGlobal	26
montaKe	27

Introdução

Com o objetivo de modelar os campos elétricos e magnéticos de uma torre de transmissão de energia elétrica, energia esta que é utilizada por toda a população, com um carro nas proximidades, foi aplicado o Método de Elementos Finitos com o uso das equações de Maxwell.

Problema

O problema proposto está representado na Figura 1, a seguir:

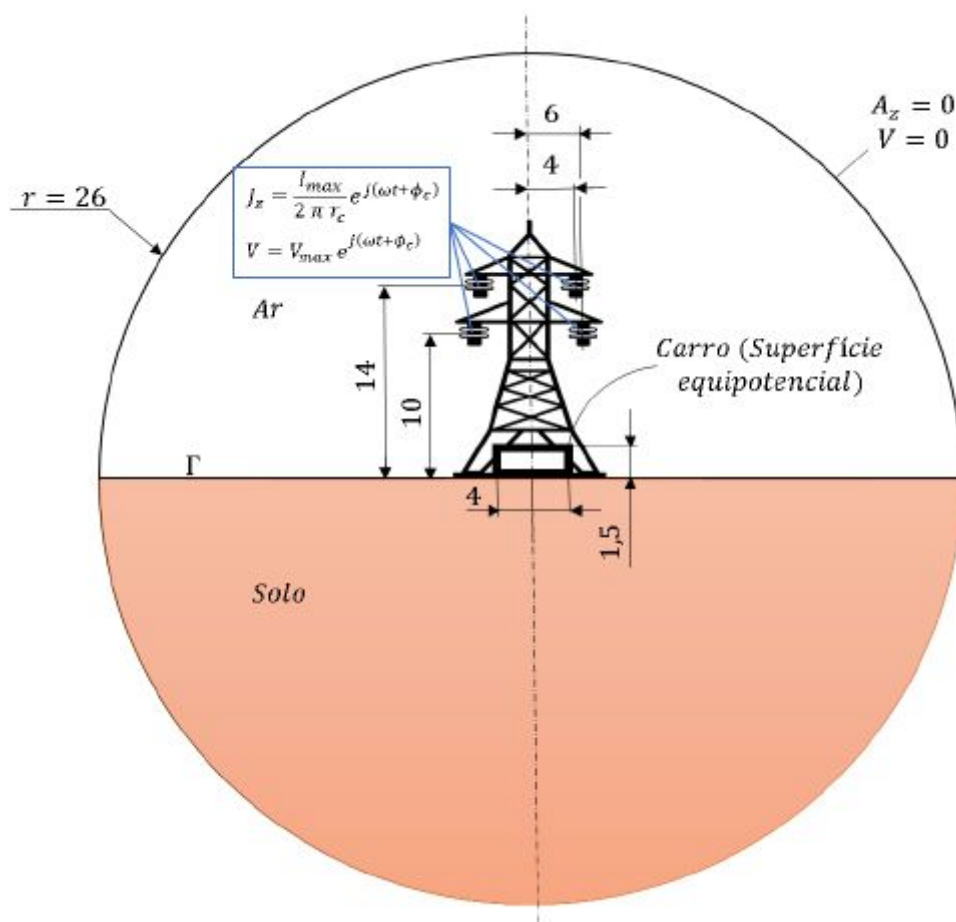


Figura1: Esquema do problema proposto pelo exercício.

Conforme mostrado na figura, o meio em questão é bidimensional, circular e está dividido em dois meios, sendo eles ar e solo. Existem no sistema 4 fontes de energia, todas elas situadas na parte superior da torre. O carro encontra-se no centro do círculo, suas dimensões são dadas e sua superfície é equipotencial. As condições de contorno são nulas tanto para campo magnético (A_z) quanto para campo elétrico (V).

Equacionamentos

As equações dos fenômenos físicos envolvidos no problemas foram dadas no enunciado. Já as equações que envolveram a aplicação do Método de Elementos Finitos foram dadas ao longo do curso de Mecânica Computacional. Em ambos os casos as equações estão apresentadas nesta seção.

Equações dos Fenômenos Físicos

1. Vetor Campo Elétrico E:

$$\vec{E} = - \nabla V$$

2. Potencial Espacial elétrico V:

$$\nabla^2 V = 0$$

3. Potencial Elétrico nas Fases:

$$V = V_{\text{máx}} e^{j(\omega t + \phi_c)}$$

4. Frequência Angular:

$$\omega = 2\pi f$$

5. Campo Magnético H:

$$\vec{H} = \vec{B} / \mu$$

6. Densidade do Fluxo Magnético B:

$$\vec{B} = \nabla \times \vec{A}$$

7. Vetor Potencial Magnético A:

$$\vec{A} = (0, 0, A_z) \quad \nabla^2 A_z = -\mu^* J_z$$

8. Vetor Densidade de corrente J_z :

$$J_z = \frac{Imáx}{2\pi rc} * e^{j(\omega t + \phi_c)}$$

Equações do Método de Elementos Finitos

1. Para as grandezas Φ (V e A_z) vale a equação:

$$\nabla^2 \Phi = f(x, y)$$

2. Dentro de cada elemento triangular, valerá a função interpoladora:

$$\Phi_e(x, y) = N_1 \Phi_1 + N_2 \Phi_2 + N_3 \Phi_3$$

Onde:

$$N_i = (a_i + b_i x + c_i y) / 2A_e$$

$$\begin{aligned} a_1 &= x_2 y_3 - x_3 y_2; & a_2 &= x_3 y_1 - x_1 y_3; & a_3 &= x_1 y_2 - x_2 y_1 \\ b_1 &= y_2 - y_3; & b_2 &= y_3 - y_1; & b_3 &= y_1 - y_2 \end{aligned}$$

$$c_1 = x_3 - x_2; \quad c_2 = x_1 - x_3; \quad c_3 = x_2 - x_1$$

$$A_e = (b_1 c_2 - b_2 c_1)/2$$

Sendo 1, 2 e 3 os nós do elemento triangular, numerados no sentido anti-horário.

3. A relação que se estabelece entre os vetores e matrizes é:

$$[K_e]\{\Phi_e\} = \{b_e\}$$

Onde:

Matriz local de cada elemento e:

$$[K]_e = \frac{k}{4A_e} \begin{bmatrix} b_1 b_1 + c_1 c_1 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ & b_2 b_2 + c_2 c_2 & b_2 b_3 + c_2 c_3 \\ \text{sim.} & & b_3 b_3 + c_3 c_3 \end{bmatrix}$$

Vetor b_e é a soma do vetor de carregamento local e do vetor de carga distribuída, respectivamente apresentados abaixo:

$$\left\{ \begin{array}{l} \int_{L_{13}} N_1 k \frac{\partial T}{\partial n} d\ell + \int_{L_{32}} N_1 k \frac{\partial T}{\partial n} d\ell + \int_{L_{21}} N_1 k \frac{\partial T}{\partial n} d\ell \\ \int_{L_{13}} N_3 k \frac{\partial T}{\partial n} d\ell + \int_{L_{32}} N_3 k \frac{\partial T}{\partial n} d\ell + \int_{L_{21}} N_3 k \frac{\partial T}{\partial n} d\ell \\ \int_{L_{13}} N_2 k \frac{\partial T}{\partial n} d\ell + \int_{L_{32}} N_2 k \frac{\partial T}{\partial n} d\ell + \int_{L_{21}} N_2 k \frac{\partial T}{\partial n} d\ell \end{array} \right\}$$

$$\frac{fA_e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

A partir das equações locais dos elementos, é possível se criar as matrizes globais do sistema somando cada termo na sua respectiva posição. Com isso, a partir das condições de contorno e do valor das constantes dadas pelo problema é possível fazer a plotagem das grandezas escalares e vetoriais pedidas no enunciado.

Resultados

Os equacionamentos apresentados na seção anterior foram desenvolvidos no MATLAB com o uso dos seguintes valores para as constantes do problema:

$V_{max}[kV]$	$I_{max}[A]$	$\phi_c[rad]$	$f[Hz]$	$r_c[m]$
500	200	0	60	0,02
$\mu_{ar}[H/m]$	$\mu_{solo}[H/m]$	$\sigma_{ar}[S/m]$	$\sigma_{solo}[S/m]$	
$1,2566 \cdot 10^{-6}$	$2 \cdot 1,2567 \cdot 10^{-6}$	$1,0 \cdot 10^{-10}$	$1,0 \cdot 10^{-2}$	

Tabela 1: Valor das constantes.

Com isso, a partir de um passo h determinado, é criada uma malha que representa o meio em questão com discretização triangular. Por ser um meio simétrico a malha é criada apenas para o semicírculo direito do sistema.

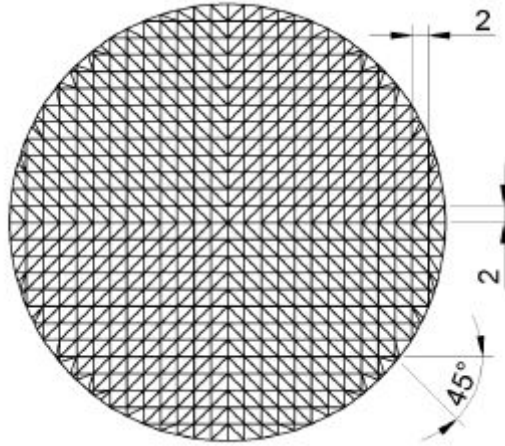


Figura 2: Discretização do domínio do problema.

Além disso, cada elemento triangular é numerado no sentido anti-horário. Para se analisar o efeito de discretizações menores nos resultados da simulação, todos os estudos foram feitos para dois valores de h , sendo eles 1 e 2. Na Figura 2 está apresentada a diferença nas dimensões dos elementos causada por essa alteração de discretização.

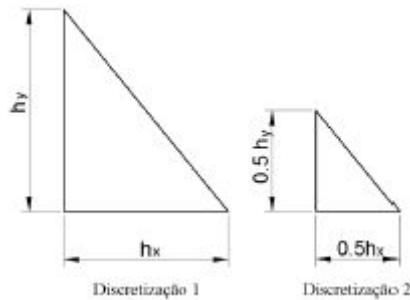


Figura 3: Discretizações utilizadas, com $h_x = h_y = 2$.

1.a) Escalares $V(x,y)$ e $A_z(x,y)$

1.a.1) Os resultados da distribuição do Potencial Elétrico $V(x, y)$ para $h=2$ e $h=1$ estão apresentados nas Figuras 6 e 7.

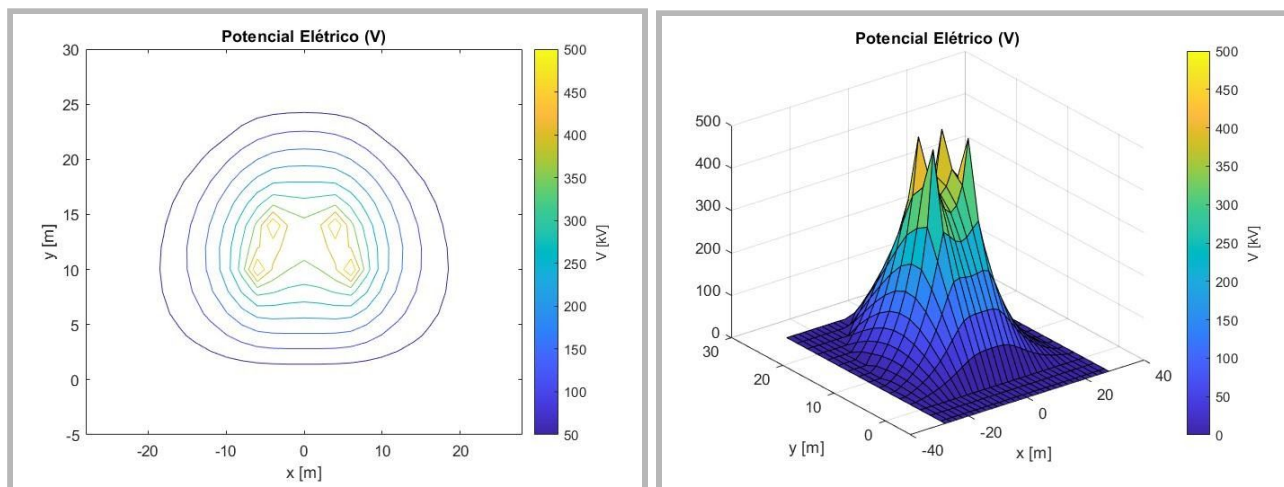


Figura 6: Distribuição do Potencial Elétrico $V(x, y)$ para $h = 2m$.

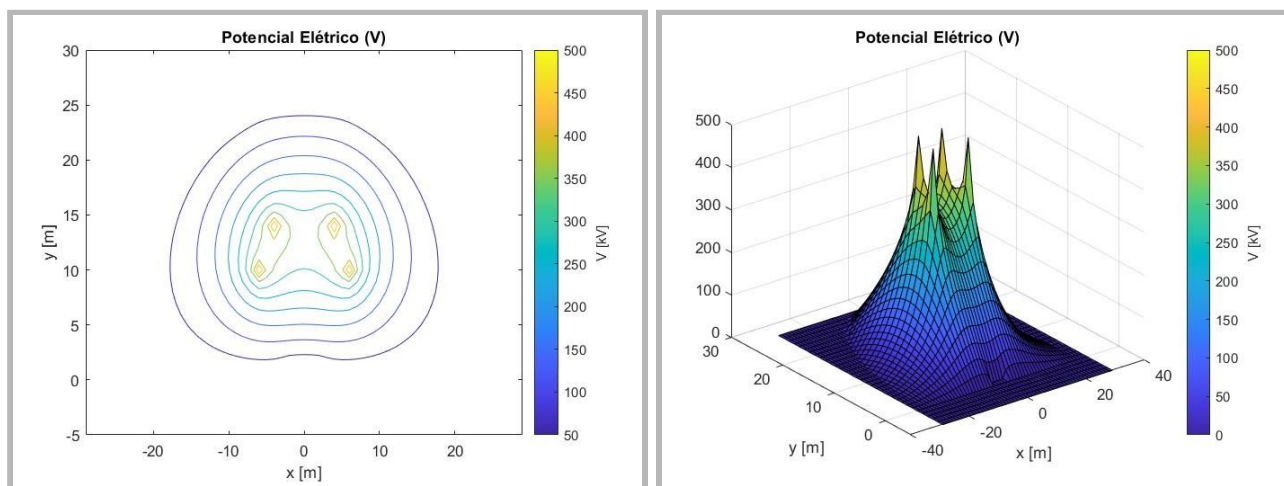


Figura 7: Distribuição do Potencial Elétrico $V(x, y)$ para $h = 1m$.

Ao analisar esse resultado, é possível observar a influência do passo h na simulação, de modo que o gráfico com $h=1$ é mais refinado que o gráfico com $h=2$. Isso mostra que o uso de uma discretização mais detalhada torna o modelo mais próximo da realidade, sendo indicado essa análise para problemas quantitativos mais específicos e com necessidade de precisão. Por outro lado a simulação mais robusta, com $h=2$, possui praticamente o mesmo formato do modelo refinado, dessa forma o uso de um passo maior se torna suficiente para análises mais qualitativas de um fenômeno. Além disso, o uso de $h=2$ possui um custo computacional muito menor que modelo com $h=1$.

Com relação ao formato do gráfico gerado, o potencial elétrico possui seu valor máximo nas proximidades das fontes, como é esperado, e vai caindo até chegar a zero na chão e no contorno do sistema. Vale ressaltar também que a deformação no centro da curva ocorre pela presença da superfície equipotencial do carro, de modo que no interior do carro o potencial é nulo.

1.a.2) Os resultados da distribuição do Potencial Magnético $A_z(x, y)$ para $h=2$ e $h=1$ estão apresentados nas Figuras 8 e 9.

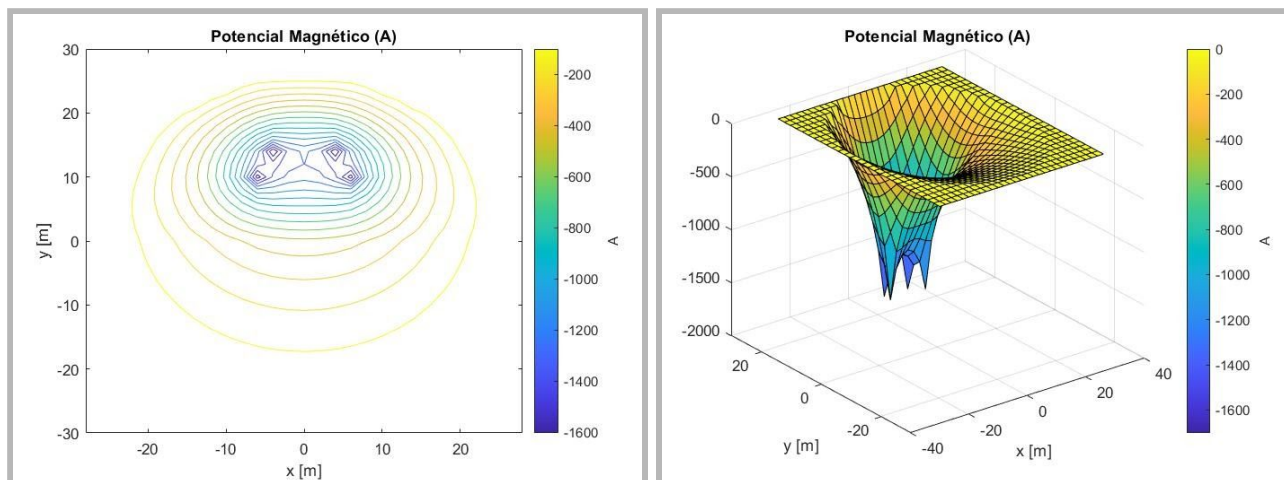


Figura 8: Distribuição do Potencial Magnético $A_z(x, y)$ para $h = 2m$.

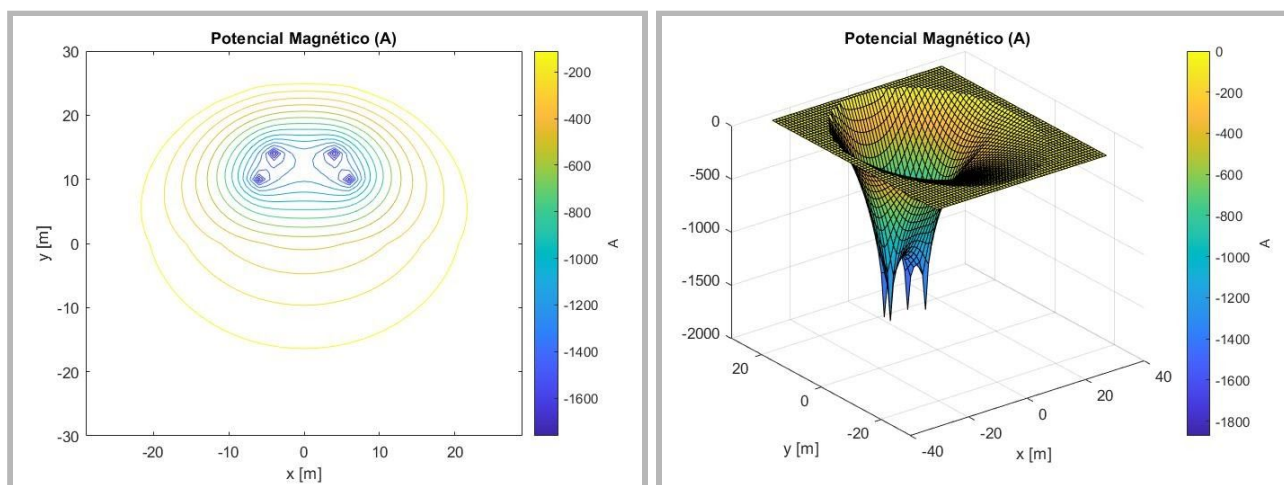


Figura 9: Distribuição do Potencial Magnético $A_z(x, y)$ para $h = 1m$.

Da mesma maneira que no caso do potencial elétrico, analisando a influência do passo h na simulação, fica claro que o gráfico com $h=1$ é mais refinado que o gráfico com $h=2$. Novamente, o passo menor é mais indicado para análise de problemas quantitativos mais específicos e com necessidade de precisão. Em contrapartida, com o passo maior é obtido praticamente o mesmo formato do modelo refinado, logo o uso dessa simulação é suficiente para análises mais qualitativas de um fenômeno. Além disso, o uso de $h=2$ possui um custo computacional muito menor que modelo com $h=1$.

Com relação ao gráfico gerado, o potencial magnético possui valores oposto ao potencial elétrico, com valores negativos. Mesmo assim, o potencial magnético ainda é maior, em módulo, nas proximidades das fontes. O módulo vai caindo até chegar a zero no contorno do sistema. Vale ressaltar que o potencial magnético não zera no solo. Ao contrário do que acontece com o potencial elétrico, o efeito do magnetismo se estende para uma boa parcela do semicírculo inferior do sistema, ainda que com valores baixos, módulo.

1.b) Vetores Densidade de fluxo magnético, Intensidade de campo magnético e Intensidade de campo elétrico

1.b.1) Os resultados do Vetor de Intensidade de Campo Elétrico $E(x, y)$ para $h=2$ e $h=1$ estão apresentados nas Figuras 10 e 11.

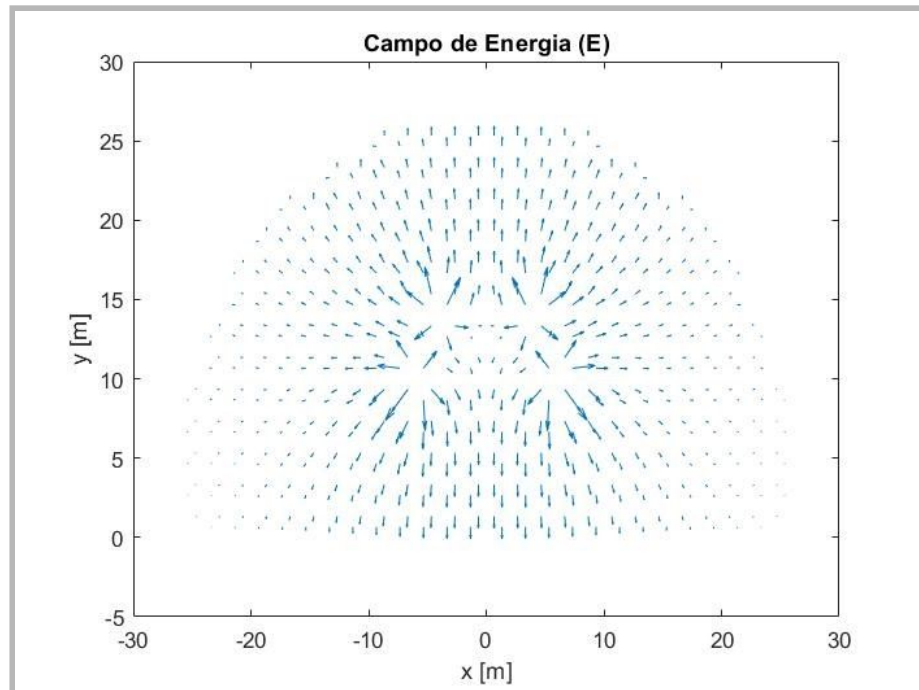


Figura 10: Vetor de Intensidade de Campo Elétrico $E(x, y)$ para $h = 2m$.

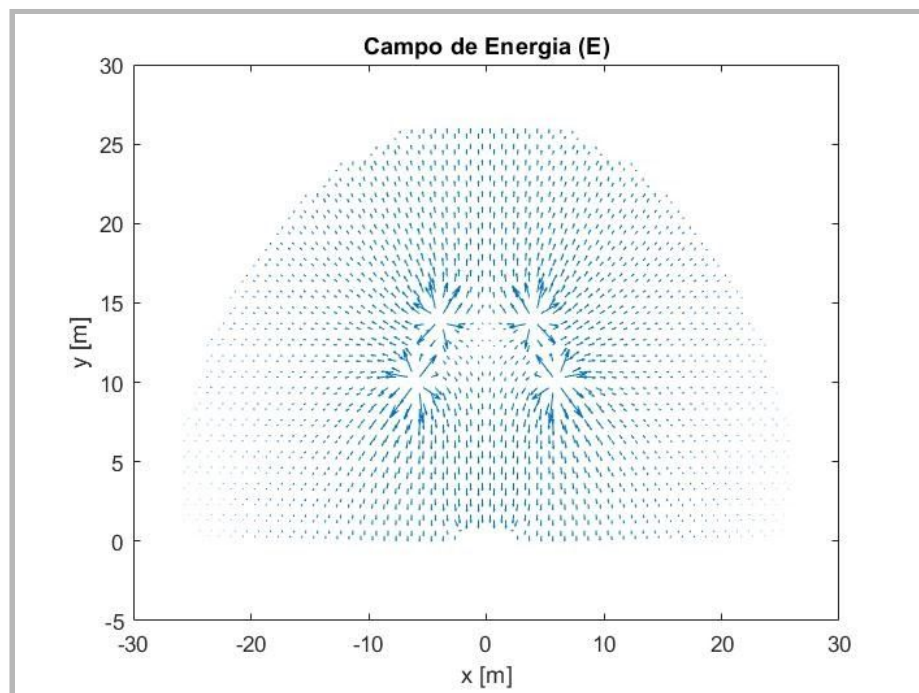


Figura 11: Vetor de Intensidade de Campo Elétrico $E(x, y)$ para $h = 1m$

Seguindo a linha do potencial elétrico V , o vetor campo elétrico possui módulos maiores nas proximidades das fontes. Isso faz sentido, pois o valor do potencial elétrico será diretamente proporcional com o valor do campo.

1.b.2) Os resultados do Vetor de Intensidade de Campo Magnético $H(x, y)$ para $h=2$ e $h=1$ estão apresentados nas Figuras 12 e 13.

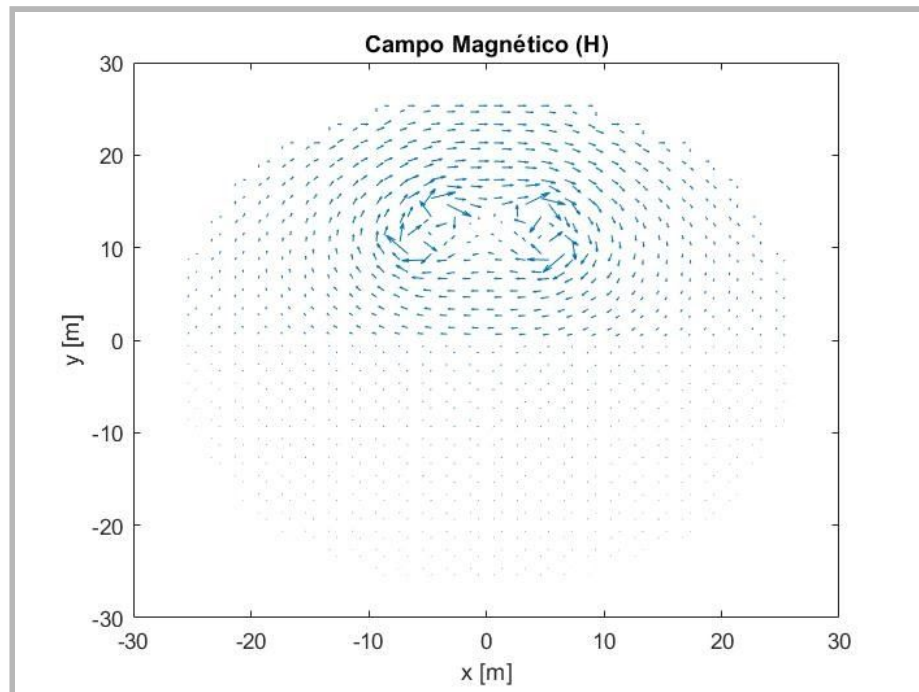


Figura 12: Vetor de Intensidade de Campo Magnético $H(x, y)$ para $h = 2m$.

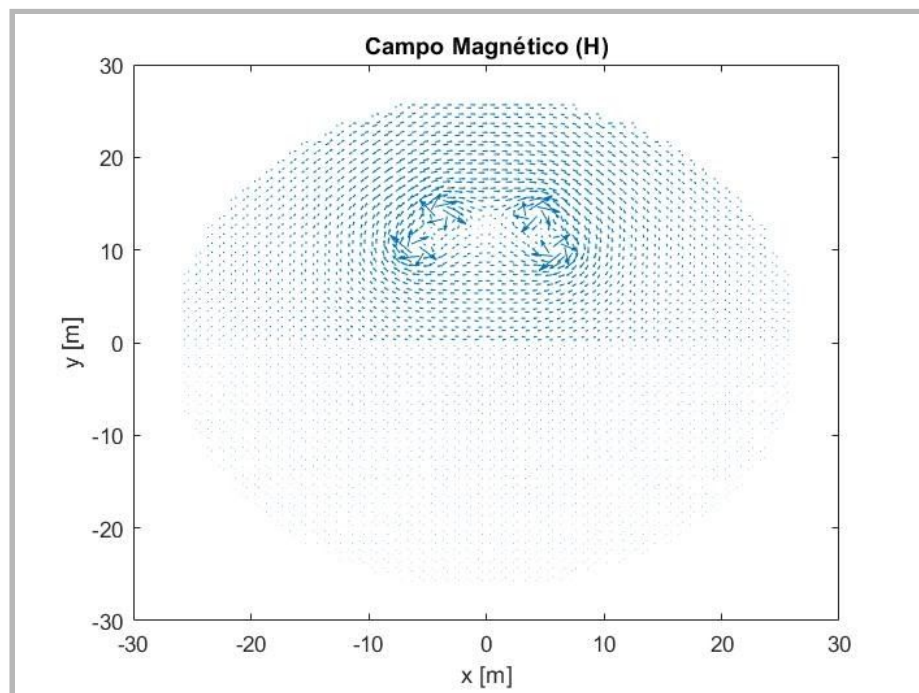


Figura 13: Vetor de Intensidade de Campo Magnético $H(x, y)$ para $h = 1m$.

Assim como no caso das grandezas elétricas, o vetor campo magnético possui valores diretamente proporcionais ao módulo do potencial magnético A_z . Na mesma linha, é possível notar que há campo magnético no espaço compreendido pelo solo no sistema, diferente do campo elétrico que é nulo nesta região.

1.b.3) Os resultados do Vetor de Densidade de Fluxo Magnético $B(x, y)$ para $h=2$ e $h=1$ estão apresentados nas Figuras 12 e 13.

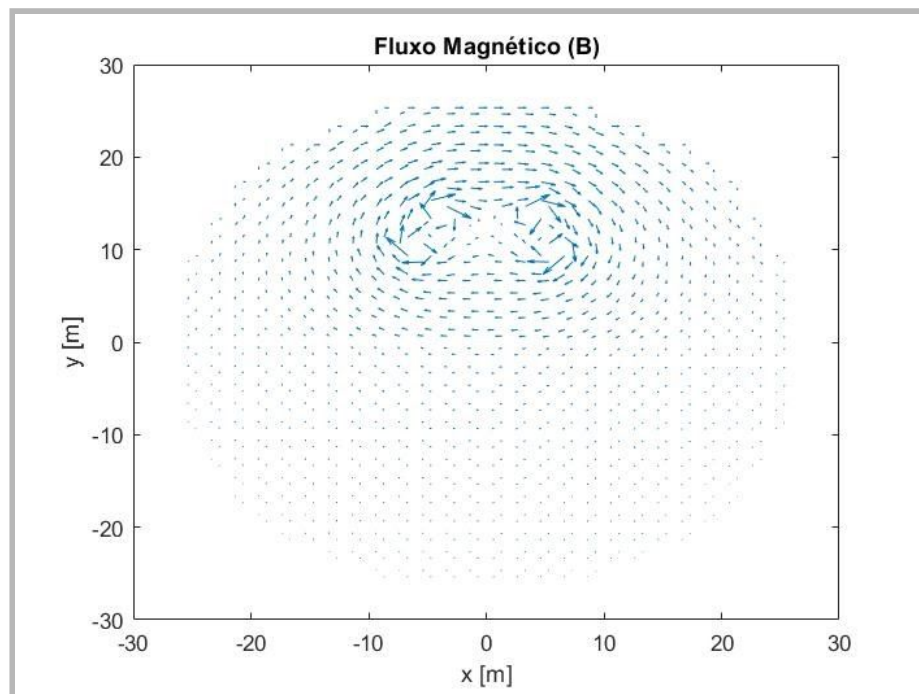


Figura 14: Vetor de Densidade de Fluxo Magnético $B(x, y)$ para $h = 2m$.

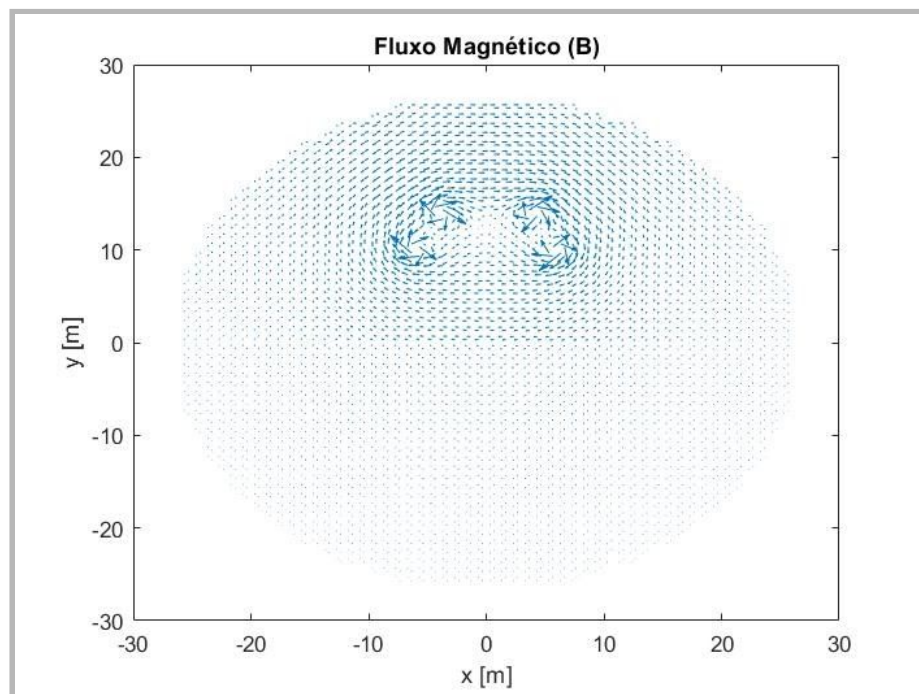


Figura 15: Vetor de Densidade de Fluxo Magnético $B(x, y)$ para $h = 1m$.

Da mesma maneira que o campo magnético, o módulo do vetor fluxo magnético é diretamente proporcional ao módulo do potencial magnético

1.b.4) Vale ressaltar também, que a influência do passo h na análise das grandezas vetoriais é mesma que para as grandezas escalares (potencial elétrico V e potencial magnético A_z)

Sobreposição de grandezas vetoriais e escalares

Para validar o argumento de que as grandezas vetoriais possuem módulo diretamente proporcional com o módulo das grandezas escalares, ambas as grandezas foram plotadas no mesmo gráfico nesta seção.

Fenômenos elétricos

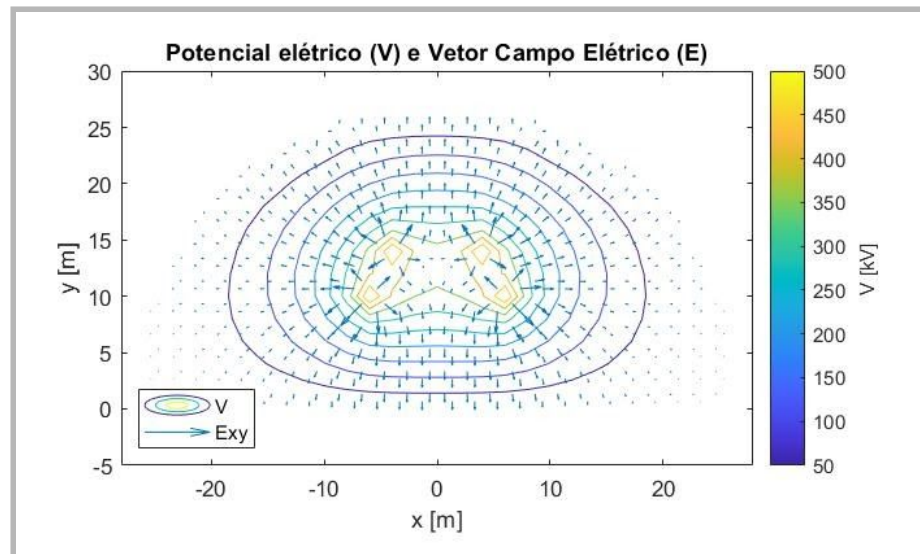


Figura 16: Potencial Elétrico $V(x, y)$ e Vetor Campo Elétrico $E(x, y)$ para $h=2$.

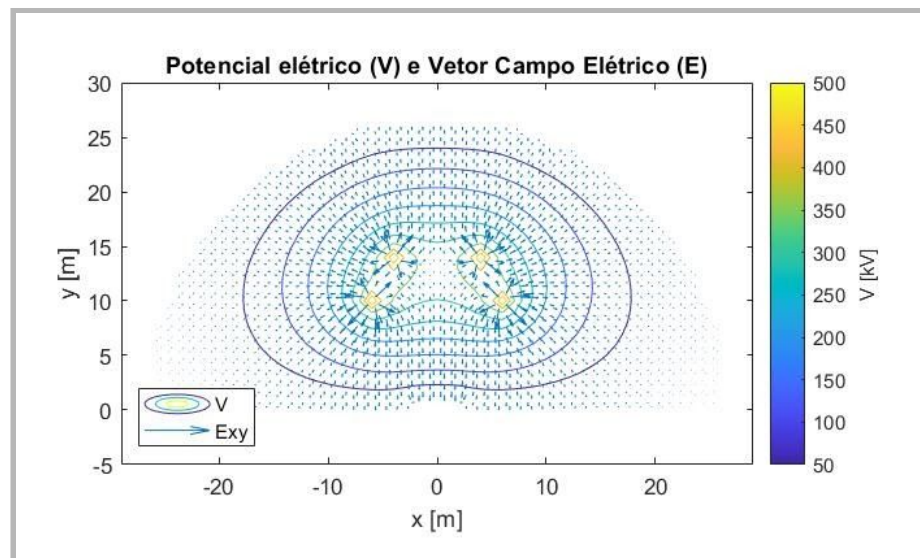


Figura 17: Potencial Elétrico $V(x, y)$ e Vetor Campo Elétrico $E(x, y)$ para $h=1$.

Fenômenos Magnéticos

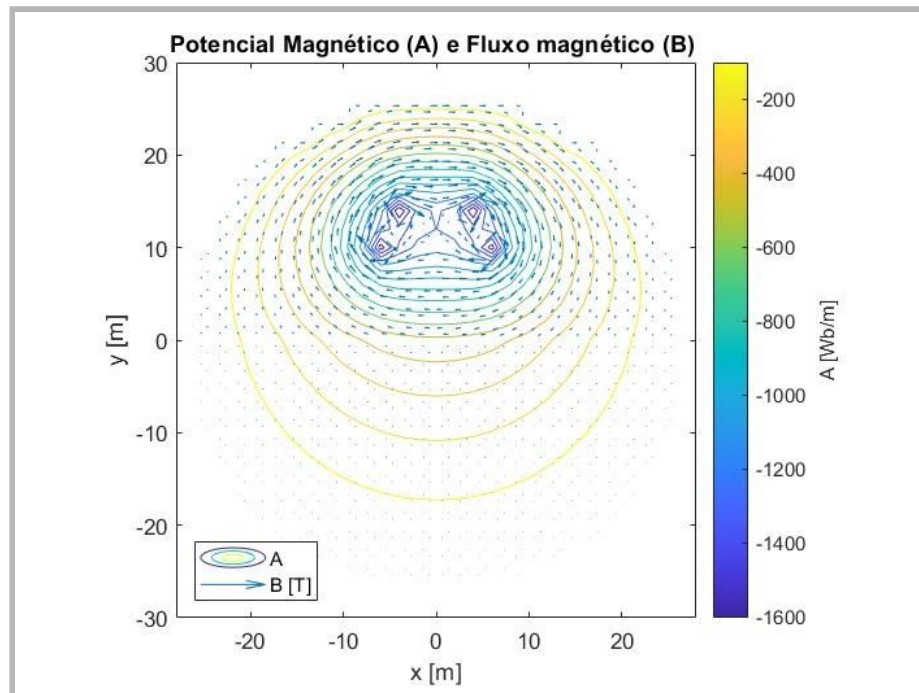


Figura 18: Potencial Magnético $A_z(x, y)$ e Vetor Fluxo Magnético $B(x, y)$ para $h=2$.

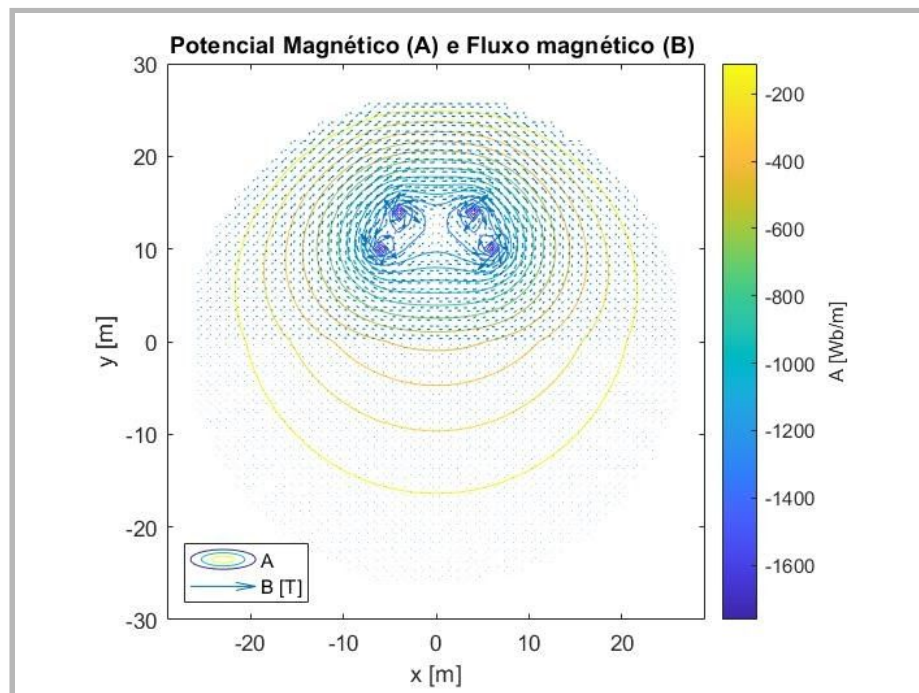


Figura 19: Potencial Magnético $A_z(x, y)$ e Vetor Fluxo Magnético $B(x, y)$ para $h=1$.

Conclusão Final

Após a realização do Exercício Programa Opcional observamos que, portanto, a discretização do sistema com elementos menores influencia o resultado da simulação tornando-o mais detalhado, ou seja, mais próximo da realidade. Esse benefício, por outro lado, vem acompanhado de um custo computacional elevado.

Foi observado também que a discretização mais robusta resulta em um modelo com praticamente o mesmo formato do modelo refinado. Logo, caso o intuito do estudo seja fazer uma análise mais qualitativa de um fenômeno, é válido usar uma discretização com elementos maiores, economizando assim trabalho computacional. Porém, se a análise requisitar respostas quantitativas mais precisas e específicas, será necessária uma simulação refinada.

Com relação ao fenômeno estudado, ficou constatado que nas proximidades das fontes o potencial elétrico e magnético possui seu valor máximo, em módulo, assim como suas respectivas grandezas vetoriais, sendo a tensão elétrica positivas e a magnética negativa.

Códigos

main

Executa o algoritmo principal utilizando as funções auxiliares:

```
clear all;
close all;
clc

% Constantes
h = 1;
r = 26;
miAr = 1.2566E-6;
miSolo = 2*1.2567E-6;
sigmaAr = 1E-10;
sigmaSolo = 1E-2;
iMax = 200;
rc = 0.02;
vMax = 500;
alturaCarro=1.5;
lCarro=4;

% gera a malha
x = 0;
y = 0;
elementos = []; %matriz de elementos
%criacao dos elementos triangulares a partir de coordenadas
while y < r % enquanto esta dentro do raio
    x=0;
    xMax = (r^2 - y^2)^(1/2);
    while x< xMax+1
        %representando cada triangulo de cada elemento
        %Triangulo debaixo no ar
        t1.x1 = x;
        t1.y1 = y;
        t1.x2 = x+h;
        t1.y2 = y;
        t1.x3 = x+h;
        t1.y3 = y+h;
        %Triangulo de cima no ar
        t2.x1 = x;
        t2.y1 = y;
```

```

        t2.x2 = x+h;
        t2.y2 = y+h;
        t2.x3 = x;
        t2.y3 = y+h;
        %Triangulo de cima no solo
        t3.x1 = x;
        t3.y1 = -y;
        t3.x2 = x+h;
        t3.y2 = -y-h;
        t3.x3 = x+h;
        t3.y3 = -y;
        %Triangulo debaixo no solo
        t4.x1 = x;
        t4.y1 = -y;
        t4.x2 = x;
        t4.y2 = -y-h;
        t4.x3 = x+h;
        t4.y3 = -y-h;

        elementos = [elementos t1 t2 t3 t4];
        x = x+h; %proximo passo
    end
    y = y+h; %proximo passo
end
%cria nohs nos elemetnos
[nos, elementos] = criaNos(elementos);

% nos onde estao as fontes
Asource = - miAr*iMax/(2*pi*rc);
fonte1x = 4;
fonte2x = 6;
fonte1y = 14;
fonte2y = 10;

%mi e sigma aos elementos
for i = 1:length(elementos)
    if elementos(i).y3 > 0
        elementos(i).sigma = sigmaAr;
        elementos(i).mi = miAr;
    else
        elementos(i).sigma = sigmaSolo;
        elementos(i).mi = miSolo;
    end
end
end

%Potencial 0 nos pontos da borda
for i = 1:length(nos)

```



```

        no = nos(i);
        nRad = ((no.x)^2 + (no.y)^2)^(1/2);
        if (nos(i).x == fonte1x && nos(i).y == fonte1y) || (nos(i).x == fonte2x
&& nos(i).y == fonte2y)
            nos(i).V = vMax;
            nos(i).ASource = Asource;
        end
        % carro
        if (nos(i).x <=lCarro/2 && nos(i).y >=0 && nos(i).y <=alturaCarro)
            nos(i).V = 0;
        end
        % fontes
        if nRad >= r
            nos(i).V = 0;
            nos(i).A = 0;
        end
    end
end

% pot eletrico e magnetico
F = zeros(length(nos),1);
K = matrizGlobal(elementos,length(nos),1);
FA = zeros(length(nos),1);
KA = matrizGlobal(elementos,length(nos),0);

% matrizes F
for i = 1:length(nos)
    % matriz FV
    if ~isempty(nos(i).V)
        for j = 1:length(K)
            F(j) = F(j) - K(j,i) * nos(i).V;
            K(i,j) = 0;
            K(j,i) = 0;
        end
        F(i) = nos(i).V;
        K(i,i) = 1;
    end
    % Matriz FA
    if ~isempty(nos(i).ASource)
        FA(i) = nos(i).ASource;
    end
end

```

```

end
% Borda
if ~isempty(nos(i).A)
    for j = 1:length(K)
        FA(j) = FA(j) - KA(j,i) * nos(i).A;
        KA(i,j) = 0;
        KA(j,i) = 0;
    end
    FA(i) = nos(i).V;
    KA(i,i) = 1;
end
end

% solucao
V = linsolve(K, F);
A = linsolve(KA, FA);

% valores nos nohs
for i = 1:length(nos)
    nos(i).V = V(i);
    nos(i).A = A(i);
end

% derivando
for i = 1:length(elementos)
    [dx_,dy_] = derivaElemento(elementos(i), nos, 1);
    elementos(i).Ex = -dx_;
    elementos(i).Ey = -dy_;
    [dxA_,dyA_] = derivaElemento(elementos(i), nos, 0);
    elementos(i).Bx = dyA_;
    elementos(i).By = -dxA_;
    elementos(i).Hx = elementos(i).Bx/elementos(i).mi;
    elementos(i).Hy = elementos(i).By/elementos(i).mi;
end

%-----
-----

% graficos
Vlinha = zeros(56/h, 30/h);
Alinha = zeros(56/h, 30/h);

```

```

for k = 1:length(nos)
    x = nos(k).x;
    y = nos(k).y;
    j = 1 + x/h;
    i = 1+(y+28)/h;
    Vlinha(i,j) = V(k);
    Alinha(i,j) = A(k);
end
Vlinha = [fliplr(Vlinha) Vlinha];
Alinha = [fliplr(Alinha) Alinha];

[X,Y] = meshgrid(0:h:29, -28:h:27);
xFlip = [-fliplr(X) X];
yFlip = [fliplr(Y) Y];

% V
figure;
contour(xFlip, yFlip, Vlinha, 'ShowText','off', 'LabelSpacing',500);
title("Potencial Elétrico (V)");
c = colorbar;
c.Label.String = 'V [kV]';
xlabel("x [m]");
ylabel("y [m]");
ylim([-5 30]);

% A
figure;
contour(xFlip, yFlip, Alinha,16, 'ShowText','off', 'LabelSpacing',500);
title("Potencial Magnético (A)");
c = colorbar;
c.Label.String = 'A';
xlabel("x [m]");
ylabel("y [m]");
ylim([-30 30]);

% E
figure();
xms = zeros(length(elementos),1);
yms = zeros(length(elementos),1);
Exs = zeros(length(elementos),1);
Eys = zeros(length(elementos),1);
for i = 1:length(elementos)
    xms(i) = elementos(i).xm;

```

```

        yms(i) = elementos(i).ym;
        Exs(i) = elementos(i).Ex;
        Eys(i) = elementos(i).Ey;
    end
    quiver([-fliplr(xms) xms],[fliplr(yms) yms],[-fliplr(Exs)
    Exs],[fliplr(Eys) Eys], 0.1);
    title("Campo de Energia (E)");
    xlabel("x [m]");
    ylabel("y [m]");
    ylim([-5 30]);

% B
figure();
Bxs = zeros(length(elementos),1);
Bys = zeros(length(elementos),1);
Hxs = zeros(length(elementos),1);
Hys = zeros(length(elementos),1);
for i = 1:length(elementos)
    Bxs(i) = elementos(i).Bx;
    Bys(i) = elementos(i).By;
    Hxs(i) = elementos(i).Hx;
    Hys(i) = elementos(i).Hy;
end
quiver([-fliplr(xms) xms],[fliplr(yms) yms],[fliplr(Bxs)
Bxs],[-fliplr(Bys) Bys], 0.1);
title("Fluxo Magnético (B)");
xlabel("x [m]");
ylabel("y [m]");
ylim([-30 30]);

% H
figure;
quiver([-fliplr(xms) xms],[fliplr(yms) yms],[fliplr(Hxs)
Hxs],[-fliplr(Hys) Hys], 0.1);
title("Campo Magnético (H)");
xlabel("x [m]");
ylabel("y [m]");
ylim([-30 30]);

%magneticos
figure;

```

```

contour(xFlip, yFlip, Alinha,16, 'ShowText','off', 'LabelSpacing',500);
title("Potencial Magnético (A) e Fluxo magnético (B) ");
c = colorbar;
c.Label.String = 'A [Wb/m]';
xlabel("x [m]");
ylabel("y [m]");
axis equal;

hold on;
quiver([-fliplr(xms) xms],[fliplr(yms) yms],[fliplr(Hxs)
Hxs],[-fliplr(Hys) Hys], 0.1);
legend({'A ', "B [T] "},'Location','southwest');
ylim([-30 30]);

%eletricos
figure;
contour(xFlip, yFlip, Vlinha, 'ShowText','off', 'LabelSpacing',500);
title("Potencial elétrico (V) e Vetor Campo Elétrico (E)");
c = colorbar;
c.Label.String = 'V [kV]';
xlabel("x [m]");
ylabel("y [m]");
axis equal;

hold on;
quiver([-fliplr(xms) xms],[fliplr(yms) yms],[-fliplr(Exs)
Exs],[fliplr(Eys) Eys], 0.1);
legend({'V ', "Exy "},'Location','southwest');
ylim([-5 30]);

% Vsurf
figure;
surf(xFlip, yFlip, Vlinha);
title("Potencial Elétrico (V)");
c = colorbar;
c.Label.String = 'V [kV]';
xlabel("x [m]");
ylabel("y [m]");
ylim([-5 30]);

```

```
% Asurf
figure;
surf(xFlip, yFlip, Alinha);
title("Potencial Magnético (A)");
c = colorbar;
c.Label.String = 'A';
xlabel("x [m]");
ylabel("y [m]");
ylim([-30 30]);
```

Auxiliares

criaNos

Recebe cada elemento, adiciona e relaciona os nós aos elementos da modelagem.

```
function [nos, elementos] = criaNos(elementos)
    no.x = 0;
    no.y = 0;
    nos=[no];

    %cria um ponto se nao estiver na lista
    for i = 1:length(elementos)
        elementos(i).xm = (elementos(i).x1 + elementos(i).x2 +
elementos(i).x3)/3;
        elementos(i).ym = (elementos(i).y1 + elementos(i).y2 +
elementos(i).y3)/3;
        elemento = elementos(i);
        %vertice 1
        elementos(i).no1 = -1;
        noParcial.x = elemento.x1;
        noParcial.y = elemento.y1;
        for j = 1:length(nos)
            if isequal(noParcial, nos(j))
                elementos(i).no1 = j;
                break
            end
        end
        if elementos(i).no1 == -1
            nos = [nos, noParcial];
            elementos(i).no1 = j+1;
        end
        %vertice 2
        elementos(i).no2 = -1;
        noParcial.x = elemento.x2;
        noParcial.y = elemento.y2;
        for j = 1:length(nos)
            if isequal(noParcial, nos(j))
                elementos(i).no2 = j;
            end
        end
        if elementos(i).no2 == -1
            nos = [nos, noParcial];
            elementos(i).no2 = j+1;
        end
    end
end
```

```

        %vertice 3
        elementos(i).no3 = -1;
        noParcial.x = elemento.x3;
        noParcial.y = elemento.y3;
        for j = 1:length(nos)
            if isequal(noParcial, nos(j))
                elementos(i).no3 = j;
            end
        end
        if elementos(i).no3 == -1
            nos = [nos, noParcial];
            elementos(i).no3 = j+1;
        end
    end
end
end

```


derivaElemento

Calula as derivadas espaciais de cada nó para utilizar em outras contas.

```
function [dx,dy] = derivaElemento(elemento, nos, ehSigma)
    %Calcula as derivadas parciais em um elemento
    x1 = elemento.x1;
    x2 = elemento.x2;
    x3 = elemento.x3;
    y1 = elemento.y1;
    y2 = elemento.y2;
    y3 = elemento.y3;

    if ehSigma
        V1 = nos(elemento.no1).V;
        V2 = nos(elemento.no2).V;
        V3 = nos(elemento.no3).V;
    else
        V1 = nos(elemento.no1).A;
        V2 = nos(elemento.no2).A;
        V3 = nos(elemento.no3).A;
    end
    b1= y2 - y3;
    b2= y3 - y1;
    b3= y1 - y2;
    c1= x3 - x2;
    c2= x1 - x3;
    c3= x2 - x1;
    Ae= (b1*c2 - b2*c1)/2;
    dx = (1/(2*Ae))* (b1*V1 + b2*V2 + b3*V3);
    dy = (1/(2*Ae))* (c1*V1 + c2*V2 + c3*V3);
end
```

matrizGlobal

Recebe uma matrizLocal de um elemento e a soma na matriz global K.

```
function K = matrizGlobal(elementos, lenNos, ehSigma)
    K = zeros(lenNos);
    for i = 1:length(elementos)
        matrizLocal = montaKe(elementos(i),ehSigma);
        n1 = elementos(i).no1;
        n2 = elementos(i).no2;
        n3 = elementos(i).no3;
        K(n1,n1) = K(n1,n1) + matrizLocal(1,1);
        K(n1,n2) = K(n1,n2) + matrizLocal(1,2);
        K(n1,n3) = K(n1,n3) + matrizLocal(1,3);
        K(n2,n1) = K(n2,n1) + matrizLocal(2,1);
        K(n2,n2) = K(n2,n2) + matrizLocal(2,2);
        K(n2,n3) = K(n2,n3) + matrizLocal(2,3);
        K(n3,n1) = K(n3,n1) + matrizLocal(3,1);
        K(n3,n2) = K(n3,n2) + matrizLocal(3,2);
        K(n3,n3) = K(n3,n3) + matrizLocal(3,3);
    end
end
```

montaKe

Cria uma matriz K_e para um elemento de triângulo:

```
function [K_el] = montaKe(elemento, ehSigma)
    % Montagem da matriz K do elemento
    x1 = elemento.x1;
    x2 = elemento.x2;
    x3 = elemento.x3;
    y1 = elemento.y1;
    y2 = elemento.y2;
    y3 = elemento.y3;

    b1 = y2 - y3;
    b2 = y3 - y1;
    b3 = y1 - y2;
    c1 = x3 - x2;
    c2 = x1 - x3;
    c3 = x2 - x1;

    Ae = (b1*c2 - b2*c1)/2;

    if ehSigma
        const = elemento.sigma;
    else
        const = elemento.mi;
    end

    K_el = 1/(4*Ae)*const*[b1*b1+c1*c1 b1*b2+c1*c2 b1*b3+c1*c3;
                          b1*b2+c1*c2 b2*b2+c2*c2 b2*b3+c2*c3;
                          b1*b3+c1*c3 b2*b3+c2*c3 b3*b3+c3*c3];
end
```