



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

**PTC3213 - 2024**

13 de outubro de 2024

**PRIMEIRO EXERCÍCIO COMPUTACIONAL**  
**MÉTODO DAS DIFERENÇAS FINITAS**

Professor: **Luis Lebestein**

Turma: 2

Grupo: **yC**

Alunos	NUSP
Arthur Soresini Bedin	14754306
João Victor Cavalcante Miranda	14582927
Marcos Gabriel	10266322

1. **Dados do problema** ..... 2
2. **Resultados numéricos** ..... 2
3. **Mapa dos Quadrados Curvilíneos** ..... 2
4. **Script Octave** ..... 3

U= 6;P=0;A=3

1. Dados do problema

6	0	6	0	3		6	0	3
<i>a</i> (cm)	<i>b</i> (cm)	<i>c</i> (cm)	<i>d</i> (cm)	<i>g</i> (cm)	<i>h</i> (cm)	$\epsilon_r$	$\sigma$ (mS/m)	$\sigma_{\text{dual}}$ (mS/m)
11	5	4	1	3	0.5	2.5	2.5	3.5

2. Resultados Numéricos

<i>R</i> ( $\Omega$ )	<i>C</i> (pF)	$\rho_{S\text{min}}$ (nC/m <sup>2</sup> )	Tubos de corrente	<i>R</i> <sub>dual</sub> ( $\Omega$ )
128.564	0,068	-201,088	19	0,444

3. Mapa dos Quadrados Curvilíneos

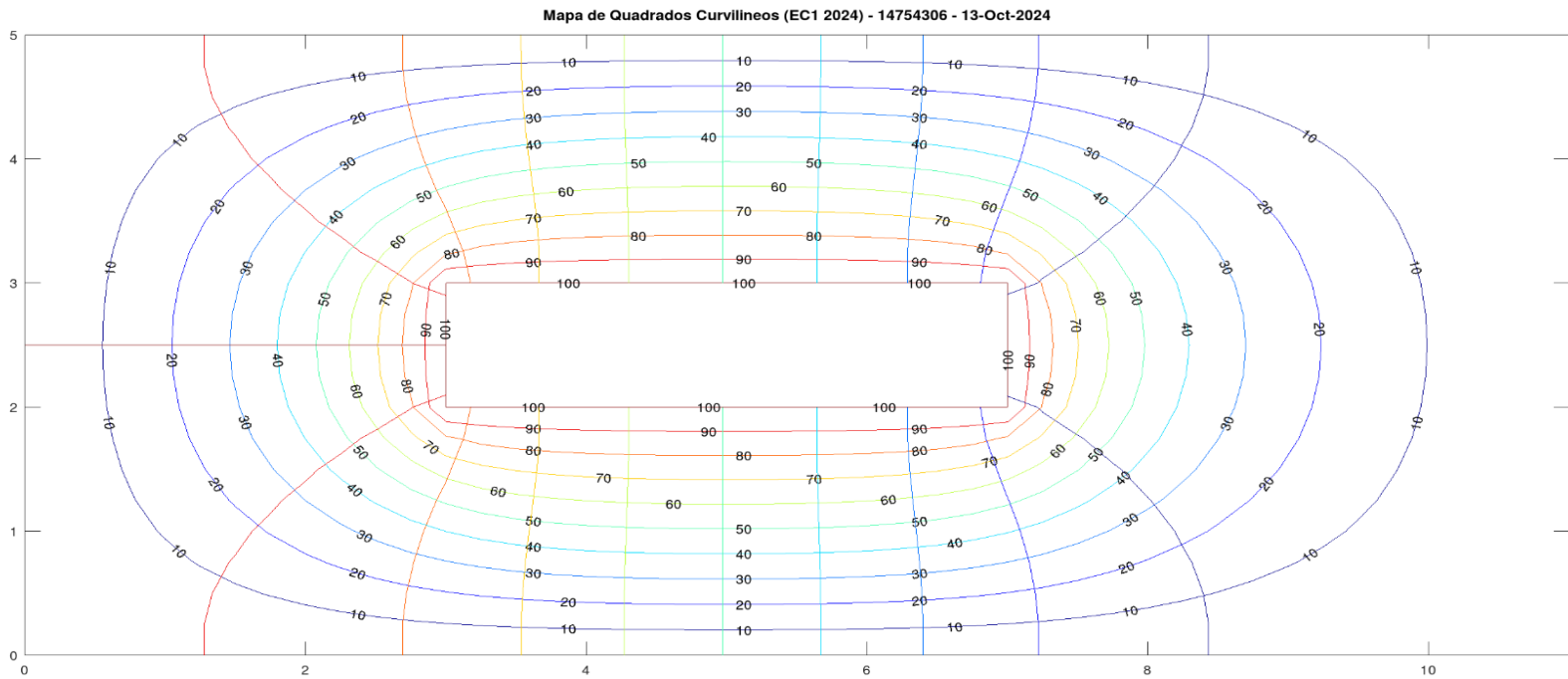


Fig. 1 - Mapa dos quadrados curvilíneos (número de tubos de corrente: 19)

#### 4. Script Octave

```
% ///////////////////////////////////////////////////////////////////
%                               PTC3213 - EC1 - 2024 - DIFERENÇAS FINITAS

%   #14754306 Arthur Soresini Bedin
%   #14582927 João Victor Cavalcante Miranda
%   #10266322 Marcos Gabriel
%                               Complete os campos com [Preencher Aqui]
%
% ///////////////////////////////////////////////////////////////////
clear;
clf;
warning ("off");
pkg install -local -forge  matgeom;
pkg load matgeom;
clc;

%   Dados de entrada

NUSP = 14754306 ; % NUSP do 1o aluno (ordem alfab.)

a=   11;
b=   5 ;
c=   4 ;
d=  b-4; % 1
g=   3 ;

h=(b-d)/2; %

epsr=   2.5   ; % adimensional
sigma=   2.5   ; % mS/m
sigma_dual=   3.5   ; % S/m

eps0=  8.85418717*10^(-12) ; % F/m

Vmin=  0 ;      % Volts, de acordo com o enunciando.pdf
Vmax= 100 ;     % Volts, idem

%                               Definicao do dominio
```

---

```

% A variavel dx abaixo ? a discretizacao utilizada. Valores diferentes
% daqueles sugeridos abaixo nao funcionarao. Diminua o dx para gerar a
%versao final a ser entregue. Ou seja, aumentar a resolu??o da simula??o

% Defini??es para o c?lculo usando o MDF

dx=0.25; %% Sugest?o do Prof: Mude para dx=0.25 somente quando for gerar os
resultados finais!!!
erro=0.0;
start=start_Dual= (Vmin + Vmax)/2; % um bom palpite ? sempre a m?dia, a partir
da? melhoraremos iterativamente.
iter=0;
dy=dx;
lx=a;
ly=b;
% "Resolu??o" da tela, quantidade de quadradinhos que devemos ter em cada
dimens?o
Nx=round(lx/dx)+1;
Ny=round(ly/dy)+1;
% Geometria do problema

% S?o retangulos, mas tem 5 pontos pois o ?ltimo ? igual ao primeiro para fechar
o anel
% Os aneis v?o em dire??es contr?rias
% Os aneis s?o os "vertices dos ret?ngulos um dentro do outro"
ring1= [0 0; lx 0; lx ly; 0 ly; 0 0];
ring2=[g h; g h+d; g+c h+d; g+c h; g h];

%novo anel, onde ser?o realizados os c?lculos
ring3 = [0+1 0+1; lx-1 0+1; lx-1 ly-1; 0+1 ly-1; 0+1 0+1];

% array de arrays
polyg={ring1,ring2};
% ? a uni?o dos dois poligonos
% simplesmente as duas arrays ring1 e ring2 concatenadas
verts = polygonVertices(polyg);
% sintaxe para fazer uma grid
xgv=((1:Nx)-1)*dx;
% faz uma sequ?ncia de 0 at? Nx-1 e multiplica por dx, para assim ter todos os
pontos em x at?
ygv=((1:Ny)-1)*dx;
% tipo um produto cartesiano
[x,y]=meshgrid(xgv,ygv);
% O x tem todos os valores de x para cada ponto no conjunto xgv cartesiano ygv

% aparentemente essas linhas s?o redundantes, pois ring1 e verts1 cont?m
exatamente a mesma informa??o, talvez pud?ssemos chamar ringN de vertsN sem
maiores problemas
verts1 = polygonVertices(ring1);
verts2 = polygonVertices(ring2);

```

---

```
xv1=verts1(:,1);
yv1=verts1(:,2);
xv2=verts2(:,1);
yv2=verts2(:,2);

% vertices do polígono de calculo
verts3 = polygonVertices(ring3);
xv3 = verts3(:,1);
yv3 = verts3(:,2);
[in3,on3] = inpolygon(x,y,xv3,yv3);
% S? para deixar logo tudo junto

% Essa fun?o gera dois valores, in1 e on1, que s?o arrays booleanas que seguem
o seguinte padr?o:
% in1 ? 1 para todos os valores do espao (x,y) dado que estiver dentro do
poligono de vertices xv1 e yv1, e 0 caso contr?rio. Como uma ?rea hachurada
% on1 ? a borda do polígono. Como um per?metro.
[in1,on1] = inpolygon(x,y,xv1,yv1);

[in2,on2] = inpolygon(x,y,xv2,yv2);

% Atribui Condições de contorno

r=find(in1&~in2|on2); % tudo

p=find(in1&~on1&~in2); %so nos internos = tudo que est? na placa e n?o est? na
borda de fora, mas pega a borda de dentro

q=find(on1|on2); %so fronteira = s? os dois perimetros, um retangulo dentro do
outro

iVmax=find(on2);

% Dica do prof: pode ser usado ! no lugar de ~ como uma extens?o do octave.
iFuro=find(in2&~on2);

% inicializa??o da fun??o potencial
Phi_prev=zeros(size(x));
Phi_new=zeros(size(x));

% a borda exterior, definimos que Vmax=100, e isso n?o ser? mais alterado
Phi_new(iVmax)= Vmax;
Phi_new(iFuro)= NaN;

% Um jeito interessant?ssimo de definir os valores dentro de uma vari?vel...
```

```
% da mesma forma que ? feito para contra dom?nio
% Para todo ponto no dom?nio p, atribui-se o valor start ? imagem
Phi_new(p)= start;

% Contador de iteracoes
iter=0;

% Erro maximo entre Phi_new e Phi_prev
% Phi_new - Phi_prev ? uma subtra??o de matrizes
% abs apenas tira o valor absoluto de cada um deles
% max vai pegar o m?ximo de cada coluna
% o segundo max vai pegar o m?ximo das colunas
% Ou seja, estamos calculando o m?ximo erro em qualquer que seja o ponto
erro=max(max(abs(Phi_new-Phi_prev)));

%           Laco iterativo - Metodo das Diferencas Finitas
% agora come?a a brincadeira

while(erro > 1e-4 && iter < 1e4)% Executa ate convergir ou atingir o maximo de
iteracoes
    iter=iter+1; % Incrementa iteracao

%     Atualiza o potencial dos nos internos pela media dos 4 vizinhos - Eq.
Laplace - M.D.F.
%     Funciona especialmente bem para a Eq. de laplace, pois a ideia que ?
consequ?ncia dessa ?
%     O c?lculo do laplaciano
%     Que ? o somat?rio da segunda derivada do potencial em cada dimens?o
%     E segundas derivadas tem tudo a ver com m?dias(segundo o Richard P.
Feynman)

%     Esse la?o ? apenas passando por todos os elementos em Phi
for k=1:size(p,1);
    [i,j]=ind2sub(size(x),p(k));

Phi_new(i,j)=(Phi_new(i-1,j)+Phi_new(i+1,j)+Phi_new(i,j-1)+Phi_new(i,j+1))/4;
end
% Calcula maximo erro entre Phi_atual e Phi_prev de todo o dominio

    erro=max(max(abs(Phi_new-Phi_prev)));
    % coloca na conta do eps em quanto que est? o erro agora.
    eps(iter)=erro;

%     Atualiza a matriz de potenciais
    Phi_prev=Phi_new;
end

niter1=iter;

if (niter1 == 1e4 && erro > 1e-4)
```

---

```

        disp([' Numero maximo de iteracoes atingido sem convergencia :',
num2stg(niter1), ' iteracoes \? Erro: \n', num2str(erro), 'Os resultados podem
nao ter significado!\n']);
end

% Para calcular as derivadas parciais... Tem um jeitinho...

% Define-se as fun??es dos potenciais...
campoEx = zeros(size(x));
campoEy = zeros(size(y));

% pois o campo el?trico dentro do condutor ? zero

campoEx(iFuro)= 0;
campoEx(on1) = 0;
campoEy(iFuro)= 0;
campoEy(on1) = 0;

% p s?o as "coordenadas" em uma dimens?o de cada ponto v?lido da tela
discretizada...

% para cada elemento em p, come?ando em 1
for k=1:size(p,1)
    % pegar as coordenadas, em uma dela do formado dado por size(x) na
coordenada p(k)
    [i,j] = ind2sub(size(x), p(k));
    % parece que est? trocado, mas ? assim mesmo pois ? indexado na coluna e
depois na linha.
    campoEx(i,j) = ( (Phi_new(i,j)-Phi_new(i,j+1)) +
(Phi_new(i+1,j)-Phi_new(i+1,j+1)) )/(2*dx);
    campoEy(i,j) = ( (Phi_new(i,j)-Phi_new(i+1,j)) +
(Phi_new(i,j+1)-Phi_new(i+1,j+1)) )/(2*dy);
end

% agora que temos os campos
% Como pela geometria do problema os campos ficar?o sempre na mesma dire??o da
superf?cie podemos tirar o valor absoluto.
Sigma = 0;
for k=1:size(on3,1)
    [i,j]=ind2sub(size(x),find(on3)(k));
    if(i==ring3(1,1)|i==ring3(3,1))
        Sigma += abs(campoEy(i,j))*dy;
    end
    if(j==ring3(1,2)|j==ring3(3,2))
        Sigma += abs(campoEx(i,j))*dx;
    end
end
end

```

---

```

% Cálculo da densidade superficial de carga max min
% posso não saber qual é a forma da superfície eq. pot.
% Mas sei que o campo elétrico é perpendicular a ela...
% portanto, será a diferença dos módulos dos campos elétricos em cada ponto
% mas e qual será o próximo ponto ?

rho_s = 0;

for k=1:size(find(in1&~on1&~in2),1)
    [i,j] = ind2sub(size(x), find(in1&~on1&~in2)(k));
    % Para cada ponto, vamos calcular o módulo do campo elétrico nas redondezas
    E = (campoEx(i,j)^2 + campoEy(i,j)^2)^0.5;
    % a unidade de [campoE] = V/cm
    Eij = (campoEx(i+1,j)^2 + campoEy(i+1,j)^2)^0.5;
    EIj = (campoEx(i-1,j)^2 + campoEy(i-1,j)^2)^0.5;
    EIj = (campoEx(i,j+1)^2 + campoEy(i,j+1)^2)^0.5;
    Eij = (campoEx(i,j-1)^2 + campoEy(i,j-1)^2)^0.5;

    rho_s(k+1) = max( [ abs(E-Eij) , abs(E-EIj) , abs(E-EIj) , abs(E-Eij) ] );
end

rho_s_max = eps0*epsr*max(rho_s);

% Atenção às unidades
I = sigma*1*Sigma; % [sigma] = (kohm*m)^-1, 1 m , [Sigma] = Volts % [I] = mA
R = (Vmax-Vmin)/I; % kohm
Cap = (eps0*epsr)/(R*sigma); % [C] = Farad

% dx=.5 -> I = 50.69
% dx=.25 -> I = 0.76
% dx=.125 -> I = 3.42
% dx=.0625 -> I = 0.00

% para corrigir
dx;
I = I; % mA
R *= 1e3; % Ohm
Cap *= 1e12; % pF
rho_s_max *= -1e11; % nC/m²

% Problema Dual (Somente para tracado dos Quadrados Curvilineos!)
% Atribui Condições de Contorno

iyDual=find( (y(:, :) < ly/1.999) & (y(:, :) > ly/2.001) );
iVmaxdual=find( (x(iyDual) > (-0.01)) & (x(iyDual) < (1.0001*g)));
i0=find( (x(iyDual) > (0.9999*(g+c))) & (x(iyDual) < (1.0001*l_x)) );
xfe=find( x(iVmax) < 1.0001*min(x(iVmax)) ); xfd=find( x(iVmax) >

```



---

```

0.9999*max(x(iVmax)) );
yfa=find( y(iVmax)> 0.9999*max(y(iVmax)) ); yfb=find( y(iVmax)<
1.0001*min(y(iVmax)) );
tol=1e-4;
for k=1:size(iVmax,1);
    if ( abs( x(iVmax(k))-min(x(iVmax)) )< tol && abs( y(iVmax(k))-min(y(iVmax))
)< tol)
        [ieb,jeb]=ind2sub(size(x), iVmax(k));
        elseif (abs( x(iVmax(k))-min(x(iVmax)) )< tol && abs(
y(iVmax(k))-max(y(iVmax)) )< tol)
            [iea,jea]=ind2sub(size(x), iVmax(k));
            elseif ( abs( x(iVmax(k))-max(x(iVmax)) )< tol && abs(
y(iVmax(k))-min(y(iVmax)) )< tol)
                [idb,jdb]=ind2sub(size(x), iVmax(k));
                elseif (abs( x(iVmax(k))-max(x(iVmax)) )< tol && abs(
y(iVmax(k))-max(y(iVmax)) )< tol)
                    [ida,jda]=ind2sub(size(x), iVmax(k));
            end
        end
end

Dual_prev=zeros(size(x));
Dual_new=Dual_prev;
Dual_new(r)= -1;
Dual_new(iFuro)= NaN;
Dual_new(iyDual(iVmaxdual))=Vmax;
Dual_new(iyDual(i0))=Vmin;
p2=find(Dual_new(p) < 0);
Dual_new(r)= start_Dual;
Dual_new(iFuro)= NaN;
Dual_new(iyDual(iVmaxdual))=Vmax;
Dual_new(iyDual(i0))=Vmin;

% Contador de iteracoes - dual
iter2=0;

% Erro maximo entre Phi_new e Phi_prev (Dual)
erro2=max(max(abs(Dual_new-Dual_prev)));

%      Laco iterativo (Problema Dual) - MDF(N?o ? a placa de madeira, pode
significar M?todo das Diferen?as Finitas)

while(erro2 > 1e-3 && iter2 < 1e4)% Executa ate convergir ou atingir o maximo de
iteracoes
    iter2=iter2+1; % Incrementa iteracao
%      Atualiza o potencial das fronteiras
%      Da mesma forma que foi feito com a matrix de potenciais ?

    Dual_new(1,:)=Dual_prev(2,:);
    Dual_new(Ny,:)=Dual_prev(Ny-1,:);
    Dual_new(:,1)=Dual_prev(:,2);

```

---

---

```

Dual_new(2:Ny-1,Nx)=Dual_prev(2:Ny-1,Nx-1);
for k=2:size(xfe,1)-1
    [ie,je]=ind2sub(size(Dual_new), iVmax(xfe(k)));
    Dual_new(ie,je)=Dual_new(ie,je-1);
end
for k=2:size(xfd,1)-1
    [id,jd]=ind2sub(size(Dual_new), iVmax(xfd(k)));
    Dual_new(id,jd)=Dual_new(id,jd+1)
end
for k=2:size(yfb,1)-1
    [ib,jb]=ind2sub(size(Dual_new), iVmax(yfb(k)));
    Dual_new(ib,jb)=Dual_new(ib-1,jb);
end
for k=2:size(yfa,1)-1
    [ia,ja]=ind2sub(size(Dual_new), iVmax(yfa(k)));
    Dual_new(ia,ja)=Dual_new(ia+1,ja);
end
Dual_new(iyDual(iVmaxdual))=Vmax;
Dual_new(iyDual(i0))=Vmin;

% Atualiza o potencial dos nos internos pela media dos 4 vizinhos - Eq. Laplace
- M.D.F.

    for k=1:size(p2,1);
        [i,j]=ind2sub(size(x),p(p2(k)));

Dual_new(i,j)=(Dual_new(i-1,j)+Dual_new(i+1,j)+Dual_new(i,j-1)+Dual_new(i,j+1))/
4;
    end

% Cantos n?o-gregorianos (efeito das pontas)

Dual_new(ieb,jeb)=(Dual_new(ieb-1,jeb)+Dual_new(ieb+1,jeb)+Dual_new(ieb,jeb-1)+D
ual_new(ieb,jeb+1))/4;

Dual_new(iea,jea)=(Dual_new(iea-1,jea)+Dual_new(iea+1,jea)+Dual_new(iea,jea-1)+D
ual_new(iea,jea+1))/4;

Dual_new(idb,jdb)=(Dual_new(idb-1,jdb)+Dual_new(idb+1,jdb)+Dual_new(idb,jdb-1)+D
ual_new(idb,jdb+1))/4;

Dual_new(ida,jda)=(Dual_new(ida-1,jda)+Dual_new(ida+1,jda)+Dual_new(ida,jda-1)+D
ual_new(ida,jda+1))/4;

% Calcula maximo erro entre Phi_atual e Phi_prev de todo o dominio
    erro2=max(max(abs(Dual_new-Dual_prev)));
    eps2(iter2)=erro2;
% Atualiza a matriz de potenciais
    Dual_prev=Dual_new;

```

---

```

end
niter2=niter2;
if (niter2 == 1e4 && erro2 > 1e-3)
    disp([' Numero maximo de iteracoes atingido sem convergencia :',
num2stg(niter2), ' iteracoes \? Erro: \n', num2str(erro2), 'Interprete este
resultado com ressalvas!\n']);
end

% Dados de Sa?da - S?o coisas que eu irei calcular

%      CORRENTE TOTAL (A)
Somat=sum(Phi_new(2,:))+sum(Phi_new(Ny-1,:))+sum(Phi_new(:,2))+sum(Phi_new(:,Nx-
1));

I = I;

%      RESISTENCIA em ohms
R= R;

%      CAPACITANCIA em pF
Cap= Cap;

%      RESISTENCIA DUAL em ohms
% obtida atrav? da separa?o da resistencia inicial em paralelo
% em seguida pela rela?o RRdual = 1/(sig*sigdual*1)
Rdual= 1/(2*R*sigma*(1e-3)*sigma_dual*(1e-3));

%      VETOR DESLOCAMENTO
Dn=[Phi_new(2,1:Nx-1),Phi_new(1:Ny-1,Nx-1)',Phi_new(Ny-1,1:Nx-1),Phi_new(1:Ny-1,
2)']*epsr*eps0/dx*100;

%      Densidade de carga m?nima em nC/m^2
Rho_s_min= rho_s_max;

%      Numero de tubos de corrente
nsnp= 19;
% Corre?o
ntubos=10/nsnp;

%      IMPRESSAO DE RESULTADOS NO TERMINAL
%      ATENCAO para as unidades:
%      R e Rdual em ohms      Cap em pF      Rho_s em nC/m^2
%fprintf('\n\n nUSP: %d\n R = %g ohms\n C = %g pF\n Rho_s_min = %g nC/m^2\n
Rdual = %g ohms\n Tubos: %g\n', NUSP, R, Cap, Rho_s_min,Rdual,floor(ntubos) );
fprintf('\n\n nUSP: %d\n R = %g ohms\n C = %g pF\n Rho_s_min = %g nC/m^2\n Rdual
= %g ohms\n Tubos: %g\n', NUSP, R, Cap, Rho_s_min, Rdual, floor(ntubos));
FIG=figure (1);

%      TRACADO DE EQUIPOTENCIAIS

```

```
V=0:10:Vmax;
colormap cool;
[C,H]=contour(x,y,Phi_new,V);
clabel(C,V);
axis('equal');
hold on

% EQUIPOTENCIAIS PROBLEMA DUAL (para tracado dos quadrados curvilineos)

deltaV= 10;
V=0:deltaV:Vmax;
colormap jet;
contour(x,y,Dual_new , V );
axis('equal');
strusp=sprintf('%d',NUSP);
titulo=['Mapa de Quadrados Curvilineos (EC1 2024) - ', strusp, ' - ', date()];
title(titulo);
hold off

% ARQUIVO DE SAIDA COM O MAPA DOS QUADRADOS CURVILINEOS
%(Grava na pasta exibida no Navegador de Arq. da interface grafica do Octave)

arq=['EC1_2024_QC_',strusp,'.png'];
print(FIG,arq);

%% Foi divertido! A partir de dois "%" o meu interpretador de linguagem faz um
outro tipo de highlight, por isso removi todos para o que parecia mais simples
para mim. Obrigado pelo c?digo, foi um desafio proveitoso
```