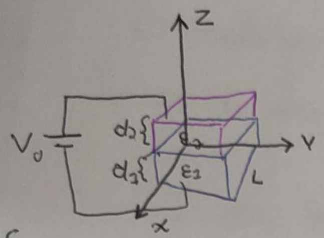
Relatório Trabalho Computacional

Eletromagnetismo - Victor Ximenes (VXCO)

Esse relatório é sobre o método de Galerkin 1D aplicado a um capacitor com dois dielétricos

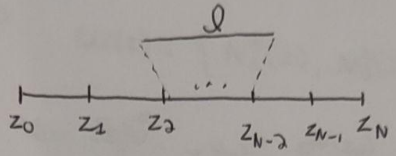
MÉTODO GALERKIN 1D

Victor Ximenes
VXCO



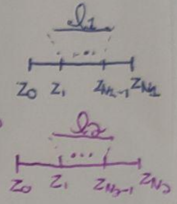
$$\begin{cases} V(0) = 0 \\ V(d) = V_0 \end{cases}$$

I) DISCRETIZAÇÃO



Divido o dielétrico 1 em N_1 segmentos

Divido o dielétrico 2 em N_2 segmentos



II) FUNÇÃO DE INTERPOLAÇÃO

$$\begin{cases} l_1 = \dfrac{d_1}{N_1} \\ l_2 = \dfrac{d_2}{N_2} \end{cases}$$

$V^e(z) = az + b \quad , \quad z_e \leq z \leq z_{e+1}$

$V^e(z_e) = V_e \; ; \; V^e(z_{e+1}) = V_{e+1}$

$$V^e(z) = V_e N_1^e(z) + V_{e+1} N_2^e(z) \; ; \quad z_e \leq z \leq z_{e+1}$$

$$N_1^e(z) = \frac{z_{e+1} - z}{l} \quad , \quad N_2^e(z) = \frac{z - z_e}{l}$$

De forma que para cada dielétrico

$$\begin{cases} N^e_{1(1)}(z) = \dfrac{z_{e+1} - z}{l_1} \\ N^e_{2(1)}(z) = \dfrac{z - z_e}{l_1} \end{cases} \quad , \quad \begin{cases} N^e_{1(2)}(z) = \dfrac{z_{e+1} - z}{l_2} \\ N^e_{2(2)}(z) = \dfrac{z - z_{e+1}}{l_2} \end{cases}$$

III)

Como não há cargas livres no interior do dielétrico, só considerado potencial nas bordas, de forma que possa utilizar a equação de Laplace

$$\nabla(\varepsilon_0 \varepsilon_r \nabla V) = 0$$, em que ε_r é a permissividade relativa

$$\frac{d}{dz}\left(\varepsilon_0 \varepsilon_r \frac{dV}{dz}\right) = 0$$

A partir disso, multiplico ambos lados pela função de base $w(z)$, e faz uma integração por parte

$$\int_{z_e}^{z_{e+1}} \frac{d}{dz}\left(\varepsilon_0 \varepsilon_r \frac{dV}{dz}\right) w(z) = 0$$

$$\int_{z_e}^{z_{e+1}} \left(\varepsilon_0 \varepsilon_r \frac{dV}{dz}\right) \frac{dw(z)}{dz} = \varepsilon_0 \varepsilon_r \frac{dV(z_{e+1})}{dz} w(z_{e+1}) - \varepsilon_0 \varepsilon_r \frac{dV(z_e)}{dz} w(z_e)$$

IV) OBTER SISTEMA LINEAR

O método de Galerkin considera que a função de base é igual a função de interpolação $\omega(z) = \{ N_1^e(z), N_2^e(z) \}$

$$\begin{cases} -V_e \dfrac{\varepsilon_0 \varepsilon_r^e}{l} + V_{e+1} \dfrac{\varepsilon_0 \varepsilon_r^e}{l} = -\varepsilon_0 \varepsilon_r^e \dfrac{dV(z_0)}{dz} \\ V_e \dfrac{\varepsilon_0 \varepsilon_r^e}{l} - V_{e+1} \dfrac{\varepsilon_0 \varepsilon_r^e}{l} = \varepsilon_0 \varepsilon_r^e \dfrac{dV(z_{e+1})}{dz} \end{cases}$$

Sistema linear 2x2 para cada valor de e

V) OBTER SISTEMA LINEAR GLOBAL

$$\varepsilon_1 \frac{dV^-}{dz} = \varepsilon_2 \frac{dV^+}{dz}$$

$$\begin{vmatrix} K_{11}^e & K_{12}^e \\ K_{21}^e & K_{22}^e \end{vmatrix} \begin{vmatrix} V_e \\ V_{e+1} \end{vmatrix} = \begin{vmatrix} D_e \\ D_{e+1} \end{vmatrix}$$

$$-V_0 \frac{\varepsilon_0 \varepsilon_r^0}{l} + V_1 \frac{\varepsilon_0 \varepsilon_r^0}{l} = -\varepsilon_0 \varepsilon_r^0 \frac{dV(z_0)}{dz}$$

$$V_0 \frac{\varepsilon_0 \varepsilon_r^0}{l} - V_1 \frac{\varepsilon_0 \varepsilon_r^0}{l} = \varepsilon_0 \varepsilon_r^0 \frac{dV(z_2)}{dz}$$

$$-V_1 \frac{\varepsilon_0 \varepsilon_r^1}{l} + V_2 \frac{\varepsilon_0 \varepsilon_r^1}{l} = -\varepsilon_0 \varepsilon_r^1 \frac{dV(z_1)}{dz}$$

$$V_1 \frac{\varepsilon_0 \varepsilon_r^1}{l} - V_2 \frac{\varepsilon_0 \varepsilon_r^1}{l} = \varepsilon_0 \varepsilon_r^1 \frac{dV(z_2)}{dz}$$

...

$$-V_{N-1} \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} + V_N \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} = -\varepsilon_0 \varepsilon_r^{N-1} \frac{dV(z_{N-1})}{dz}$$

$$V_{N-1} \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} - V_N \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} = \varepsilon_0 \varepsilon_r^{N-1} \frac{dV(z_N)}{dz}$$

IV) OBTER SISTEMA LINEAR

O método de Galerkin considera que a função de base é igual a função de interpolação $w(z) = \{ N_1^e(z), N_2^e(z) \}$

$$\begin{cases} -V_e \dfrac{\varepsilon_0 \varepsilon_r^e}{l} + V_{e+1} \dfrac{\varepsilon_0 \varepsilon_r^e}{l} = -\varepsilon_0 \varepsilon_r^e \dfrac{dV(z_e)}{dz} \\ V_e \dfrac{\varepsilon_0 \varepsilon_r^e}{l} - V_{e+1} \dfrac{\varepsilon_0 \varepsilon_r^e}{l} = \varepsilon_0 \varepsilon_r^e \dfrac{dV(z_{e+1})}{dz} \end{cases}$$

Sistema linear 2x2 para cada valor de e

V) OBTER SISTEMA LINEAR GLOBAL

$$\varepsilon_1 \frac{dV^-}{dz} = \varepsilon_2 \frac{dV^+}{dz}$$

$$\begin{vmatrix} K_{11}^e & K_{12}^e \\ K_{21}^e & K_{22}^e \end{vmatrix} \begin{vmatrix} V_e \\ V_{e+1} \end{vmatrix} = \begin{vmatrix} D_e \\ D_{e+1} \end{vmatrix}$$

$$-V_0 \frac{\varepsilon_0 \varepsilon_r^0}{l} + V_1 \frac{\varepsilon_0 \varepsilon_r^0}{l} = -\varepsilon_0 \varepsilon_r^0 \frac{dV(z_0)}{dz}$$

$$V_0 \frac{\varepsilon_0 \varepsilon_r^0}{l} - V_1 \frac{\varepsilon_0 \varepsilon_r^0}{l} = \varepsilon_0 \varepsilon_r^0 \frac{dV(z_2)}{dz}$$

$$-V_1 \frac{\varepsilon_0 \varepsilon_r^1}{l} + V_2 \frac{\varepsilon_0 \varepsilon_r^1}{l} = -\varepsilon_0 \varepsilon_r^1 \frac{dV(z_1)}{dz}$$

$$V_1 \frac{\varepsilon_0 \varepsilon_r^1}{l} - V_2 \frac{\varepsilon_0 \varepsilon_r^1}{l} = \varepsilon_0 \varepsilon_r^1 \frac{dV(z_2)}{dz}$$

...

$$-V_{N-1} \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} + V_N \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} = -\varepsilon_0 \varepsilon_r^{N-1} \frac{dV(z_{N-1})}{dz}$$

$$V_{N-1} \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} - V_N \frac{\varepsilon_0 \varepsilon_r^{N-1}}{l} = \varepsilon_0 \varepsilon_r^{N-1} \frac{dV(z_N)}{dz}$$

```python
#K*V = D
def makeK_e1(N1 , N , e1 , l):
    #Inicializar matriz com 0s
    K1 = np.zeros((N+1, N+1))
    #Adicionar equações lineares, inserindo o bloco de 4 equaçoes para cada iteraçao
    for i in range(0 , N1):
        #diagonal principal
        K1[i][i] = K1[i][i]  + (-e1/l)
        K1[i+1][i+1] = K1[i+1][i+1]  + (-e1/l)
        #diagonal oposta
        K1[i+1][i] = K1[i+1][i] + (e1/l)
        K1[i][i+1] = K1[i][i+1] + (e1/l)
    return K1

def makeK_e2(N2 , N , e2 , l):
    #Inicializar matriz com 0s
    K2 = np.zeros((N+1, N+1))
    #Adicionar equações lineares, inserindo o bloco de 4 equaçoes para cada iteraçao
    for i in range((N-N2) , N): #continuando de N1
        #diagonal principal
        K2[i][i] = K2[i][i]  + (-e2/l)
        K2[i+1][i+1] = K2[i+1][i+1]  + (-e2/l)
        #diagonal oposta
        K2[i+1][i] = K2[i+1][i] + (e2/l)
        K2[i][i+1] = K2[i][i+1] + (e2/l)
    return K2
K1 = makeK_e1(N1 , N , e1 , l1 ) #calcula a matriz de cada dieletrico
K2 = makeK_e2(N2 , N , e2 , l2 )
K = K1+K2 #soma as matrizes dos dieletrico , tendo a matriz global

subK = K[1:N , 1:N] #pegar submatriz cortando a primeira e ultima linha da matriz
```

Como o problema em dois dielétricos diferentes, o ideia usada no código foi somar duas matrizes $(N+1)\times(N+1)$ que no qual possuem suas equações até N_1 na primeira e de N_1 até N_1+N_2 na segunda, para ilustrar farei um exemplo de $N_1=N_2=1$ $(N=2)$

$$\begin{vmatrix} -\frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & \frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & 0 \\ \frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & -\frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & 0 \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} V_0 \\ V_1 \\ 0 \end{vmatrix} = \begin{vmatrix} -\varepsilon_0\varepsilon_1^{\circ}\frac{dV(z_0)}{dz} \\ 0 \\ 0 \end{vmatrix}$$

$$+$$

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & -\frac{\varepsilon_0\varepsilon_2^{\circ}}{l_2} & \frac{\varepsilon_0\varepsilon_2^{\circ}}{l_2} \\ 0 & \frac{\varepsilon_0\varepsilon_2^{\circ}}{l_2} & -\frac{\varepsilon_0\varepsilon_2^{\circ}}{l_2} \end{vmatrix} \begin{vmatrix} 0 \\ V_1 \\ V_2 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ -\varepsilon_0\varepsilon_2^{\circ}\frac{dV(z_2)}{dz} \end{vmatrix}$$

$$\Downarrow$$

$$\begin{vmatrix} -\frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & \frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & 0 \\ \frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} & -\frac{\varepsilon_0\varepsilon_1^{\circ}}{l_1} + -\frac{\varepsilon_0\varepsilon_2^{\circ}}{l_2} & \frac{\varepsilon_0\varepsilon_2^{\circ}}{l_2} \\ 0 & \frac{\varepsilon_0\varepsilon_2}{l_2} & -\frac{\varepsilon_0\varepsilon_2}{l_2} \end{vmatrix} \begin{vmatrix} V_0 \\ V_1 \\ V_2 \end{vmatrix} = \begin{vmatrix} -\varepsilon_0\varepsilon_1^{\circ}\frac{dV(z_0)}{dz} \\ 0 \\ -\varepsilon_0\varepsilon_2^{\circ}\frac{dV(z_2)}{dz} \end{vmatrix}$$

Ao unir a parte intermediaria dos sistemas lineares 2x2, obtemos

$$\begin{bmatrix} -\frac{\varepsilon_0\varepsilon_r^0}{l} & \frac{\varepsilon_0\varepsilon_r^0}{l} & 0 & \cdots & 0 & 0 \\ \frac{\varepsilon_0\varepsilon_r}{l} & -\frac{\varepsilon_0\varepsilon_r^0}{l} + \frac{-\varepsilon_0\varepsilon_r^1}{l} & \frac{\varepsilon_0\varepsilon_r^1}{l} & \cdots & 0 & 0 \\ 0 & \frac{\varepsilon_0\varepsilon_r^1}{l} & -\frac{\varepsilon_0\varepsilon_r^1}{l} + \frac{-\varepsilon_0\varepsilon_r^2}{l} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{\varepsilon_0\varepsilon_r^{N-2}}{l} + \frac{-\varepsilon_0\varepsilon_r^{N-1}}{l} & \frac{\varepsilon_0\varepsilon_r^{N-1}}{l} \\ 0 & 0 & 0 & \cdots & \frac{\varepsilon_0\varepsilon_r^{N-1}}{l} & -\frac{\varepsilon_0\varepsilon_r^{N-1}}{l} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_{N-1} \\ V_N \end{bmatrix} = \begin{bmatrix} -\varepsilon_0\varepsilon_r \frac{dV(z)}{dz} \\ 0 \\ 0 \\ \vdots \\ 0 \\ \varepsilon_0\varepsilon_r^{N-1}\frac{dV(z)}{dz} \end{bmatrix}$$

VI) IMPOR CONDIÇÕES DE CONTORNO $\begin{cases} V(0) = 0 \\ V_N = V \end{cases}$

Sabendo as condições de contorno, ou seja, o valor de duas das variaveis V_0 e V_N já são conhecidos, dessa forma podemos acortar a primeira e a ultima linha, nos liberando de duas equações e passando para o lado direito da equacao os coeficientes que envolvem V_0 e V_N, dessa forma restando um sistema linear $(N-1)\times(N-1)$

Resolvendo o exemplo de N=2

$$\begin{vmatrix} -\frac{\varepsilon_0 \varepsilon_1^{\circ}}{l_1} & \frac{\varepsilon_0 \varepsilon_1^{\circ}}{l_1} & 0 \\ \frac{\varepsilon_0 \varepsilon_1^{\circ}}{l_1} & -\frac{\varepsilon_0 \varepsilon_1^{\circ}}{l_1} - \frac{\varepsilon_0 \varepsilon_2}{l_2} & \frac{\varepsilon_0 \varepsilon_2^{\circ}}{l_2} \\ 0 & \frac{\varepsilon_0 \varepsilon_2^{\circ}}{l_2} & -\frac{\varepsilon_0 \varepsilon_2}{l_2} \end{vmatrix} \begin{vmatrix} V_0 \\ V_1 \\ V_2 \end{vmatrix} = \begin{vmatrix} -\varepsilon_0 \varepsilon_1 \frac{dV(z_0)}{dz} \\ 0 \\ \varepsilon_0 \varepsilon_2 \frac{dV(z_2)}{dz} \end{vmatrix}$$

$$\left| \left(-\frac{\varepsilon_0 \varepsilon_1^{\circ}}{l_1} - \frac{\varepsilon_0 \varepsilon_2}{l_2} \right) \right| \left| V_1 \right| = \left| 0 - \frac{\varepsilon_0 \varepsilon_1^{\circ} V_0}{l_1} - \left(-\frac{\varepsilon_0 \varepsilon_2^{\circ} V_2}{l_2} \right) \right|$$

$$\left| \left(-\frac{\varepsilon_0 \varepsilon_1^{\circ}}{l_1} - \frac{\varepsilon_0 \varepsilon_2}{l_2} \right) \right| \left| V_1 \right| = \left| \left(-\frac{\varepsilon_0 \varepsilon_1 V_0}{l_1} + \frac{\varepsilon_0 \varepsilon_2^{\circ} V_2}{l_2} \right) \right|$$

Para $\varepsilon_1 = 2$ $\quad V_0 = 0$

$\varepsilon_2 = 4$ $\quad V_2 = 1$

$l_1 = l_2 = 10^{-3}$

Resolvendo $\boxed{V_1 = 0,6667}$

```python
def makeD(N  , V0 , Vn , K):
    Dn = np.zeros(N+1)
    Dn[1] = Dn[0] - K[1][0]*V0 #substituo o valor de V0 e altero no D[1]
    Dn[N-1] = Dn[N] - K[N-1][N]*Vn #substituo o valor de Vn e altero no D[n-1]
    Dn = Dn[1:N] #pega o vetor D com os cortes

    return Dn

D = makeD(N  , V0 , Vn , K)
V = np.linalg.solve(subK , D) #resolver sistema linar K*V = D
```

CAPACITÂNCIA TEÓRICA

$$C = \frac{\varepsilon A}{d}$$

A capacitância pode ser calculada com a associação em série da capacitância de cada dielétrico

$$\begin{cases} C_1 = \dfrac{\varepsilon_1 A}{d_1} = \dfrac{\varepsilon_1 L^2}{d_1} \\ C_2 = \dfrac{\varepsilon_2 A}{d_2} = \dfrac{\varepsilon_2 L^2}{d_2} \end{cases}$$

$$\frac{1}{C_{EQ}} = \frac{1}{C_1} + \frac{1}{C_2} \Leftrightarrow C_{EQ} = \frac{C_1 * C_2}{C_1 + C_2}$$

Em que: $L = 0{,}02$

$d_1 = d_2 = 0{,}001$

$\varepsilon_1 = 2\varepsilon_0$

$\varepsilon_2 = 4\varepsilon_0$

$$C_1 = \frac{2 \cdot (8{,}85 \cdot 10^{-12})\,(2 \cdot 10^{-2})^2}{10^{-3}} = 7{,}08 \cdot 10^{-12}$$

$$C_2 = \frac{4\,(8{,}85 \cdot 10^{-12})\,(2 \cdot 10^{-2})^2}{10^{-3}} = 14{,}16 \cdot 10^{-12}$$

$$\boxed{C_{EQ} = \frac{C_1 * C_2}{C_1 + C_2} = 4{,}72 \cdot 10^{-12}\ F}$$

CAPACITANCIA NUMÉRICA

Precisamos obter o valor da carga armazenada

$$D_n = -\varepsilon_0 \varepsilon_r \frac{\partial V}{\partial n} = \sigma_F$$

como o vetor aponta para dentro

$$n = -\hat{z}$$

$$\varepsilon_0 \varepsilon_r \frac{dV}{dz} = \sigma_F$$

Pegando a placa superior $\frac{dV}{dz}$ é o ultimo segmento

$$\frac{dV}{dz} = \frac{V_N - V_{N-1}}{l_2}$$, pois V é constante nos extremos do capacitor

Para obter a carga basta multiplicar a densidade pela área

$$Q = A \cdot \sigma_F = L^2 \frac{\varepsilon_0 \varepsilon_2 (V_N - V_{N-1})}{l_2}$$

$$C = \frac{Q}{V} = \frac{L^2 \varepsilon_0 \varepsilon_2 (V_N - V_{N-1})}{l_2 (V_N - V_0)}$$

N	C
2	$4{,}720000000000001 \cdot 10^{-12}$
3	$4{,}720000000001 \cdot 10^{-12}$
4	$4{,}7199999999999996 \cdot 10^{-12}$
5	$4{,}7199999999999996 \cdot 10^{-12}$
6	$4{,}7199999999999998 \cdot 10^{-12}$
10	$4{,}72000000000000015 \cdot 10^{-12}$
100	$4{,}720000000000000109 \cdot 10^{-12}$

```
##capacitancia teorica
C1 = e1*(L*L)/d1
C2 = e2*(L*L)/d2
#capacitancia em serie
Ceq = (C1*C2)/(C2+C1)
##capacitancia numerica
Qsuperficie = L*L*(e2*(Vn - V[len(V) - 1 ]))/l2
Vtotal = Vn - V0
Cnumerica = Qsuperficie/Vtotal
print("Valor de Capacitancia numerica: " , Cnumerica)
print("Valor de Capacitancia teórica: " , Ceq)
```