

Modelagem computacional em problemas de eletrostática: efeito de campos de borda em capacitores cilíndricos finitos

(Computational modeling in electrostatics: fringe field effect in finite cylindrical capacitors)

Ademir L. Xavier Jr.¹

Centro de Pesquisas Avançadas Wernher von Braun, Campinas, SP, Brasil
Recebido em 15/9/2006; Aceito em 23/10/2006

Nesse trabalho apresenta-se um estudo de soluções numéricas para o problema de capacitores cilíndricos finitos em eletrostática, a partir da integração da equação de Poisson utilizando um esquema de diferenças finitas. Soluções são obtidas pelo método da relaxação sucessiva. O problema ilustra o tratamento numérico na obtenção de soluções eletrostáticas em problemas diferentes dos encontrados em livros-texto. Para o problema em questão, diversas quantidades de interesse (distribuição de cargas, capacitância, energia do campo eletrostático, força) podem ser comparadas com o caso analítico infinito.

Palavras-chave: capacitância, campo elétrico, diferenças finitas.

In this work we present a study of numerical solutions for finite cylindrical capacitors based on the integration of Poisson equation using a finite difference scheme. Solutions are obtained by successive over relaxation (SOR). The problem illustrates the numerical treatment involved in the solution of problems not found in textbooks. Several interesting quantities (charge distribution, capacitance, electric field energy, force) may be compared to analytical results for the infinite capacitor.

Keywords: capacitance, electric field, finite differences.

1. Introdução

É bem conhecida a relação que fornece a capacitância para estruturas metálicas infinitas de forma cilíndrica com comprimento L , Ref. [1],

$$C = 2\pi\epsilon_0 \frac{L}{\ln\left(\frac{R_{ex}}{R_{in}}\right)}, \quad (1)$$

onde R_{ex} e R_{in} são os raios externos e internos das armaduras, respectivamente, e ϵ_0 é a permissividade elétrica do vácuo que, por simplicidade, será admitido o meio que permeia as armaduras. A capacitância, diz-se, é um parâmetro que só depende da disposição geométrica das estruturas do capacitor. A relação acima pode ser obtida admitindo-se uma distribuição de cargas uniforme sobre a superfície das armaduras - o que é bem plausível para cilindros infinitos - que é utilizada para calcular a diferença de potencial ΔV entre elas. A partir disso, utiliza-se diretamente a definição de capacitância

$$C = \frac{Q}{\Delta V}, \quad (2)$$

com Q a carga envolvida. Entretanto, tal relação não

leva em conta efeitos de borda que se acentuam quando o comprimento L é da mesma ordem dos raios dos cilindros. Isso pode ser visto facilmente supondo a mesma diferença de potencial entre as armaduras e diminuindo o comprimento L . A densidade de cargas resultante não pode permanecer a mesma, tornando-se não uniforme.

Nosso objetivo aqui é determinar uma solução numérica para a capacitância que dependa do comprimento L dos capacitores cilíndricos de forma "exata", isto é, cujo erro seja função somente da aproximação numérica utilizada para calcular a solução, e não de uma aproximação baseada em um limite analítico. Podemos ter alguma idéia sobre o efeito dos campos nas bordas a partir de gráficos que mostrem a concentração de linhas de campo nas bordas das armaduras: este é o local onde a maior parte das cargas vai se localizar. Trata-se, portanto, de um problema paradoxal uma vez que a maior parte das cargas nos casos de cilindros "curtos" está envolvida justamente na distribuição de campo que se despreza no caso ideal infinito. Com isso, nosso objetivo é ilustrar ao estudante de física e engenharia o processo de modelagem computacional de uma estrutura real a partir de um método de integração numérica para um problema simples.

¹E-mail: xavier@vonbraunlabs.com.br.

2. Geometria do problema

A Fig. 1 ilustra a geometria do problema do capacitor cilíndrico. Duas cascas cilíndricas concêntricas (armaduras) de raio R_{ex} e R_{in} ($R_{ex} > R_{in}$) têm comprimentos L_{ex} e L_{in} , e são submetidas a uma diferença de potencial elétrico $V_{ex} - V_{in}$. Em princípio, fixamos a posição do cilindro interno (o ponto médio coincide com a origem) e os cilindros são completamente concêntricos. Podemos considerar ou não tampas nas extremidades do cilindro, fato que se manipula facilmente de forma numérica na definição das condições de fronteira da equação para o potencial. Para obter os potenciais e campo elétrico no interior do capacitor (admitindo a presença de tampas) é necessário integrar a equação de Laplace (equação diferencial parcial elíptica) para o potencial U , Ref. [2], dada por

$$\nabla^2 U = 0, \tag{3}$$

sujeita a condições de fronteira específicas. No caso presente estas são condições de potenciais fixos nas armaduras. Em coordenadas cilíndricas, a equação acima se escreve como

$$\frac{\partial^2 U}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial U}{\partial \rho} + \frac{1}{\rho^2} \frac{\partial^2 U}{\partial \phi^2} + \frac{\partial^2 U}{\partial z^2} = 0, \tag{4}$$

onde ϕ é a coordenada azimutal (ρ é a coordenada radial e pode ser visto na Fig. 1). Uma vez que sobre as armaduras o potencial é constante e não depende de ϕ , não existe dependência alguma com a variável do ângulo azimutal, isto é

$$U = U(\rho, z). \tag{5}$$

Um esquema de separação de variáveis é possível, [2]. As soluções em z envolvem somas de termos exponenciais (os potenciais diminuem exponencialmente em z) enquanto que o comportamento radial envolve soma de funções de Bessel. De maneira alguma o problema da Fig. 1 resulta em uma solução com forma analiticamente fechada.

3. Relação a diferenças finitas para a equação de Laplace

3.1. Condições de fronteira

Nossa tarefa é encontrar uma solução numérica para a equação

$$\frac{\partial^2 U}{\partial \rho^2} + \frac{1}{\rho} \frac{\partial U}{\partial \rho} + \frac{\partial^2 U}{\partial z^2} = 0, \tag{6}$$

que seja flexível, admitindo variações arbitrárias nas condições de contorno. Uma vez determinado o potencial para todos os pontos do espaço, diversas quantidades de interesse podem ser calculadas facilmente.

Um esquema conhecido de aproximação numérica envolve o uso de diferenças finitas para todos os pontos do espaço onde uma solução seja procurada. Para isso, discretiza-se o espaço (Fig. 2) por meio de uma malha de $N \times M$ pontos, onde N e M são números inteiros muito maiores que 1. Tal discretização leva em conta a redução progressiva do erro numérico à medida que a “granulação” da malha diminua (o que corresponde ao limite $N \rightarrow \infty, M \rightarrow \infty$).

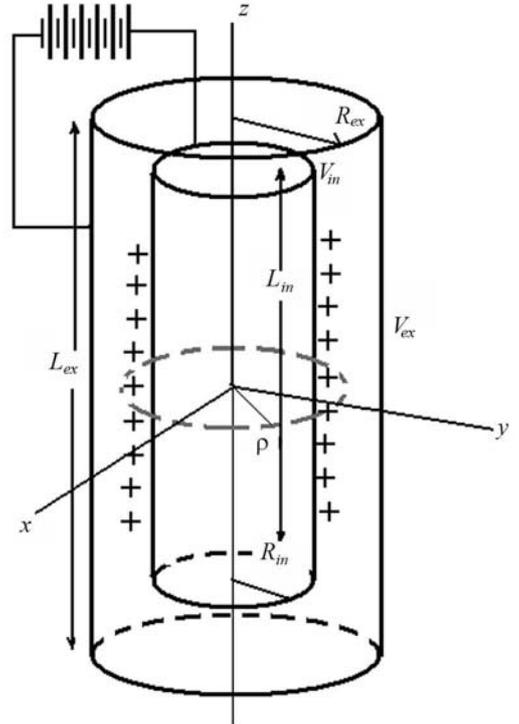


Figura 1 - Geometria do capacitor cilíndrico finito.

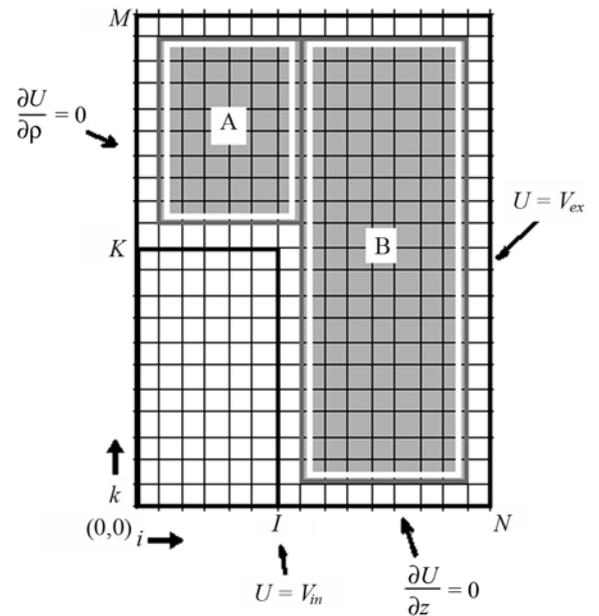


Figura 2 - Ilustração da malha para solução por diferenças finitas do potencial e condições de contorno.

As dimensões espaciais agora são parametrizadas por um conjunto de números inteiros (i, k) de forma que

$$\begin{aligned} \rho &\rightarrow i\Delta_\rho, & i &\in \{0, N\}, \\ z &\rightarrow k\Delta_z, & k &\in \{0, M\}, \end{aligned} \quad (7)$$

como mostra a Fig. 2. Como admitimos simetria na disposição dos cilindros não é necessária uma rede para todo o espaço da Fig. 1, a dependência de ϕ foi eliminada (porque não existe) e a malha está definida apenas para o quadrante superior ($z > 0, \rho > 0$). Nesse contexto, aproximações de diferenças finitas para as derivadas de primeira e segunda ordem que aparecem na Eq. (6) podem ser escritas como

$$\begin{aligned} \frac{\partial U}{\partial \rho} &\approx \frac{U_{i+1,k} - U_{i-1,k}}{2\Delta_\rho} + O(\Delta_\rho^2), \\ \frac{\partial^2 U}{\partial \rho^2} &\approx \frac{U_{i+1,k} - 2U_{i,k} + U_{i-1,k}}{\Delta_\rho^2} + O(\Delta_\rho^4), \\ \frac{\partial^2 U}{\partial z^2} &\approx \frac{U_{i,k+1} - 2U_{i,k} + U_{i,k-1}}{\Delta_z^2} + O(\Delta_z^4), \end{aligned} \quad (8)$$

onde se escreve explicitamente que o erro é da ordem Δ^2 em ρ ou z . Outras aproximações para as derivadas existem com maior ou menor erro. Entretanto, nosso tratamento se limitará às aproximações de ordem 2 em Δ . Podemos então escrever a versão em diferenças finitas para a equação de Laplace (6)

$$\begin{aligned} U_{i+1,k} \left(1 + \frac{1}{2i}\right) + U_{i-1,k} \left(1 - \frac{1}{2i}\right) - \\ 2U_{i,k} \left(1 + \frac{\Delta_\rho^2}{\Delta_z^2}\right) + \frac{\Delta_\rho^2}{\Delta_z^2} (U_{i,k+1} + U_{i,k-1}) = 0, \end{aligned} \quad (9)$$

$$\begin{aligned} 1 \leq i \leq N-1, \\ 1 \leq k \leq M-1. \end{aligned}$$

O potencial para todos os pontos discretos do espaço deve obedecer a um vínculo entre vizinhos dados pela Eq. (9). O potencial ainda tem valores bem determinados em dados pontos na malha, que são as condições de potencial impostas. Além disso, a simetria imposta pelo arranjo da Fig. 1 implica na existência de condições de contorno sobre as derivadas do potencial em certos pontos. Existem dois tipos de condições de fronteira:

Condição de von Neumann: vínculos de derivadas de potencial fixas sobre determinados pontos da malha;

Condição de Dirichlet: vínculos de potencial fixo sobre determinados pontos da malha.

Para a condição de Dirichlet, de acordo com a Fig. 2, existem dois inteiros I_1 e N_{in} tais que

$$\begin{aligned} U_{I,k} &= V_{in}, & 0 \leq k \leq K, \\ U_{N,k} &= V_{ex}, & 0 \leq k \leq N, \end{aligned} \quad (10)$$

I e K são números inteiros tais que

$$\begin{aligned} K &\approx \frac{L_{in}}{2\Delta_z}, \\ I &\approx \frac{R_{in}}{\Delta_\rho}. \end{aligned} \quad (11)$$

Pode-se tomar tais números como determinados pelos inteiros mais próximos, resultado da razão descrita na Eq. (11). Para capacitores com tampas fechadas, devemos ainda impor

$$\begin{aligned} U_{i,K} &= V_{in}, & 0 \leq i \leq I, \\ U_{i,M} &= V_{ex}, & 0 \leq i \leq N. \end{aligned} \quad (12)$$

Para as condições de von Neumann, impomos

$$\begin{aligned} U_{0,k} &= U_{1,k}, & K+1 \leq k \leq M-1, \\ U_{i,0} &= U_{i,1}, & I+1 \leq i \leq N-1. \end{aligned} \quad (13)$$

A segunda condição é necessária por causa da simetria em z (a dependência de todo potencial com z é uma função par por causa do arranjo dos cilindros). O mesmo vale para a primeira Eq. (13) que representa simetria cilíndrica.

4. Resolvendo a equação

O problema se resolve encontrando os valores do potencial U que satisfaçam a Eq. (9). Para tanto, fazemos uso do método das relaxações sucessivas, [3]. De acordo com esse método, um chute inicial para o potencial U é definido e, por iterações sucessivas, valores novos para a iteração $(n+1)$ do campo são atualizados conforme a iteração (n) via

$$U_{i,k}^{(n+1)} = U_{i,k}^{(n)} + \omega R_{i,k}^{(n)}, \quad (14)$$

onde

$$\begin{aligned} R_{i,k}^{(n)} &= U_{i+1,k} \left(1 + \frac{1}{2i}\right) + \\ &U_{i-1,k} \left(1 - \frac{1}{2i}\right) - 2U_{i,k} \left(1 + \frac{\Delta_\rho^2}{\Delta_z^2}\right) + \\ &\frac{\Delta_\rho^2}{\Delta_z^2} (U_{i,k+1} + U_{i,k-1}), \end{aligned} \quad (15)$$

é o resíduo da (n) -ésima iteração. Nesta equação ω é um parâmetro de convergência entre 1 e 2. A convergência deve ser ajustada testando-se valores de ω nesse intervalo. A convergência significa que

$$n \rightarrow \infty \quad R_{i,k}^{(n)} \rightarrow 0,$$

para todo (i, k) no domínio da solução procurada, ver Eq. (9). Como chute inicial para U podemos utilizar a média dos potenciais, admitindo o mesmo para todos os pontos

$$U_{i,k}^{(0)} = \frac{V_{ex} + V_{in}}{2}. \quad (16)$$

A principal vantagem do método das relaxações sucessivas é a simplicidade de implementação de código. Uma desvantagem é a velocidade de convergência, o método pode ser lento dependendo do problema a ser resolvido e do parâmetro de convergência ω escolhido. A Fig. 3 traz um fluxograma para um código geral para resolver o problema. Inicialmente, são estabelecidas as condições de contorno fixas que excluem os pontos da malha para os quais o potencial é procurado.

Um *loop* principal atualiza o campo escalar U segundo a Eq. (15) onde também são incluídas as condições de von Neumann e Dirichlet dadas pelas Eqs. (10) e (13). Uma condição de parada sobre o resíduo, Eq. (15), é estabelecida

$$\max |R_{i,k}^{(n)}| < \varepsilon. \quad (17)$$

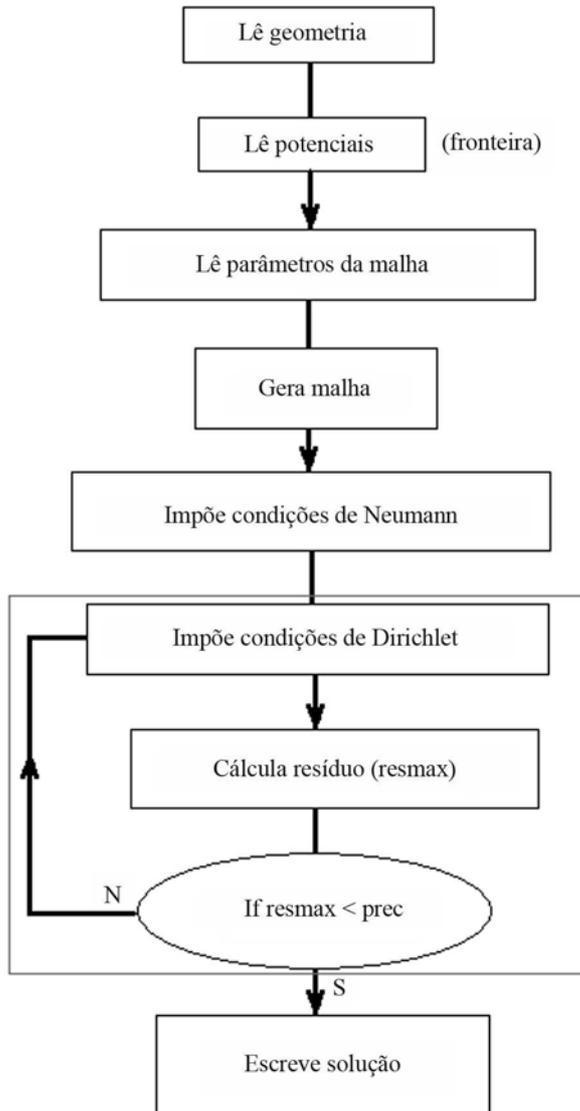


Figura 3 - Fluxograma esquemático do código mínimo para resolver o problema do capacitor.

Se o maior valor encontrado na malha para o resíduo for menos que um número $\varepsilon < 1$, então a iteração é terminada.

5. Implementação de código

Verifica-se que a convergência é conseguida se

$$\Delta_\rho = \Delta_z, \quad (18)$$

o que simplifica a equação do resíduo (15). Para gerar a malha, admitimos que existem N_ρ pontos ao longo do eixo ρ , com os quais podemos calcular a densidade de malha

$$\Delta = \frac{R_{ext}}{N_\rho}.$$

A partir disso, determinamos o número de pontos ao longo de z por

$$N_z = \frac{L_{ext}}{2\Delta}.$$

Os números K e I são calculados pelas Eqs. (11) levando em consideração a Eq. (18). Um fluxograma para o algoritmo de cálculo da solução de potencial é mostrado na Fig. 3.

Os dados iniciais para a geometria dos cilindros, potenciais aplicados em cada armadura e variáveis da malha são lidos inicialmente a partir de funções únicas designadas para isso. O Apêndice traz a implementação de um código mínimo em C++ (que pode ser gerado por compiladores gratuitos tanto em Windows como Unix, Ref. [4]) onde criamos duas estruturas específicas em `defs.h`, o arquivo de cabeçalho que contém também os protótipos das funções. Duas novas classes, `MALHA` e `CILINDRO` são criadas para facilitar a troca de parâmetros (evitando-se assim a declaração de funções com longas listas de parâmetros de entrada). O potencial é uma matriz de números criada utilizando o *template* `<vector>` na forma

`vector < vector <double >> U;`

Utilizando a função `resize()`, é possível criar dinamicamente memória e todas as tarefas de criação de funções construtoras e destrutoras ficam a cargo do *template*.

É importante impor as condições de fronteira sobre potenciais das armaduras fora de um laço principal para convergência da solução SOR. As condições de fronteira estão assim em lugares separados no código (ver função `GeraMalha()` e `SORLaplace()`, membros de `CILINDRO`). O “miolo” onde se procura a solução é soma das duas regiões A e B mostradas na Fig. 2. O método de super-relaxação é aplicado nesta região obedecendo às condições de Dirichlet para as derivadas. Nessa parte, é também calculado o maior resíduo encontrado no miolo que é então utilizado subsequente para terminar a finalização do processo iterativo, caso ele seja menor que a precisão inicialmente requerida. Para outras geometrias (por exemplo, simulação de dois cilindros coaxiais internos) deve-se modificar o código das condições de Neumann e Dirichlet nessas subrotinas. A equação de Laplace (9) entra através da função `CalculaResiduo()` que calcula o resíduo, já levando em consideração a densidade isotrópica de malha (18):

```
double CILINDRO::CalculaResiduo
(int i, int k) {
    double gm = 1. + .5 / (double)i;
    double gn = 1. - .5/ (double)i;
    return U[i+1][k]*gm + U[i-1][k]*gn
        - 4.*U[i][k] + U[i][k+1] + U[i][k-1];
}
```

A função `EscreveU()` escreve a solução no arquivo externo "U.dat". Como dissemos, esse é um código mínimo a partir do qual podemos obter o campo elétrico e calcular quantidades tais como a densidade de carga, a carga total no cilindro interno, a densidade de energia elétrica etc.

6. Algumas soluções e cálculo de quantidades elétricas

Utilizando as estruturas do código fornecido no apêndice, podemos resolver o problema através das seguintes chamadas na função principal:

```
CILINDRO ex;
ex.LeGeometria(5.,10.,10.,20.);
ex.LePotenciais(10.,0.0);
ex.Grid.LeParametrosDaMalha(50,1.e-5);
ex.GeraMalha();
ex.SORLaplace(1.5);
```

Um objeto da classe `CILINDRO` é gerado com raios interno e externo 5,0 cm e 10,0 cm, respectivamente, e comprimento interno e externo 10,0 cm e 20,0 cm, respectivamente. Nessas chamadas a unidade não está explicitada. No cálculo de campo final é que o usuário deve definir a unidade específica (o mesmo vale para os potenciais). Os potenciais interno e externo são iguais a 10,0 V e 0,0 V respectivamente. Os parâmetros da malha são estabelecidos por `LeParametrosDaMalha()` e iguais a 50 pontos ao longo de ρ com tolerância mínima de solução igual a 10^{-5} . A solução é obtida gerando uma malha para os dados fornecidos e invocando a função `SORLaplace()` com parâmetro ω de relaxação igual a 1,5.

A solução obtida é mostrada na forma de equipotenciais (não numeradas) desde o potencial interno (em branco, equivalente a 10 V) até a armadura externa (em preto, equivalente a 0 V) na Fig. 4. Essas curvas permitem imediatamente o cálculo do campo elétrico através da relação

$$\mathbf{E} = -\nabla U = \left(\frac{\partial U}{\partial \rho}, \frac{1}{\rho} \frac{\partial U}{\partial \varphi}, \frac{\partial U}{\partial z} \right). \quad (19)$$

Obviamente não há campo na direção ϕ . Para obter o campo elétrico utilizamos as derivadas numéricas

$$\frac{\partial U}{\partial \rho} \approx \frac{U_{i+1,k} - U_{i-1,k}}{\Delta} + O(\Delta^2), \quad (20)$$

$$\frac{\partial U}{\partial z} \approx \frac{U_{i,k+1} - U_{i,k-1}}{\Delta} + O(\Delta^2).$$

Para a solução de potencial mostrada na Fig. 4, uma superfície com o módulo do campo elétrico pode ser construída, que é mostrada na Fig. 5.

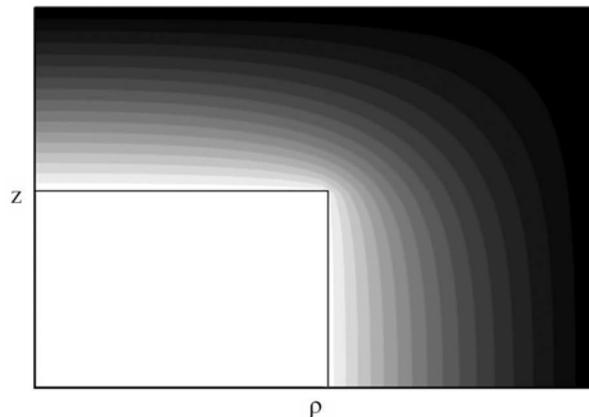


Figura 4 - Equipotenciais para os cilindros. O potencial da armadura interna (branco) é 10 V, a armadura externa aterrada (0 V - preto). O quadrado interno só delimita a geometria.

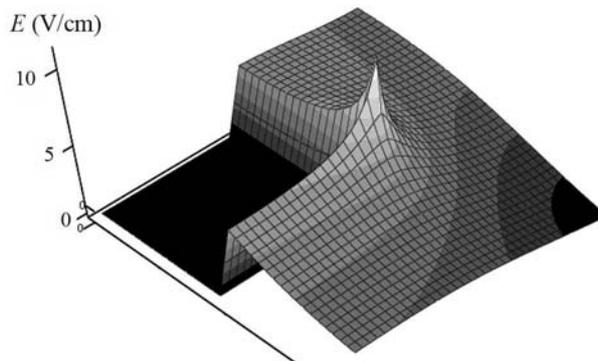


Figura 5 - Superfície 3D para o módulo do campo elétrico. Obviamente, o campo este é nulo no interior do cilindro interno.

Notemos a alta concentração de campo (e, portanto, carga elétrica) no canto do cilindro interno (Fig. 6) onde há descontinuidade (por causa da geometria) entre a tampa e a superfície. Esse é o famoso "efeito das pontas" provocado pela repulsão de cargas de mesmo sinal. Notemos ainda que o campo elétrico é quase constante entre as tampas dos cilindros (superfícies planas), enquanto que cai ao longo de ρ . Obviamente, o campo elétrico é nulo no interior do cilindro interno.

A densidade de cargas pode ser calculada a partir do campo elétrico e utilizando a lei de Gauss

$$\oiint_{\Sigma} \mathbf{E} \cdot d\mathbf{s} = \frac{Q}{\epsilon_0}, \quad (21)$$

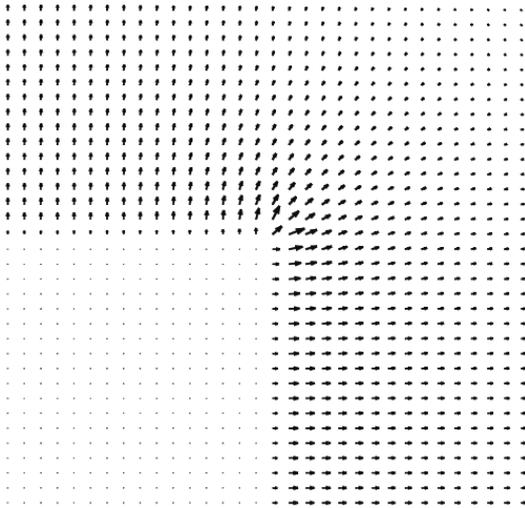


Figura 6 - Gráfico vetorial do campo elétrico (zoom) mostrando o efeito das pontas nas proximidades da junção da tampa com o lado do cilindro interno.

onde escolhemos uma superfície como na Fig. 7. Nessa situação, considerando o cilindro interno, o campo elétrico é quase constante sobre a superfície (exceto nas proximidades da tampa que, como vimos existe uma descontinuidades na derivada do campo). A integral (21) aplicada sobre a caixa com dimensão em ρ tendendo a zero torna-se

$$\sigma = \varepsilon_0 \mathbf{E} \cdot \hat{n}, \quad (22)$$

ou seja, a densidade de cargas é diretamente proporcional à componente normal do campo elétrico. Na Fig. 8 é exibida a densidade de carga calculada a partir das normais para o lado do cilindro interno e um raio da tampa para a tensão previamente estabelecida. No canto (encontro do lado com a tampa) existe uma descontinuidade por causa da discretização da rede utilizada. Obviamente, não existem “gaps” de carga no cilindro. Por método semelhante podemos encontrar a densidade de carga no cilindro interno. Notamos que, ao longo da tampa, há uma região de distribuição de carga aproximadamente constante (desde 0 até $\sim 1,5$ cm). A configuração geométrica entre as tampas gera um “capacitor de placas paralelas” onde a distribuição de cargas é bastante uniforme.

Para calcular a capacitância e avaliar a influência do campo nas bordas como função do tamanho dos cilindros é preciso avaliar a integral (21) que dá a carga total nos cilindros. Só precisamos calcular a carga total no cilindro interno e, a partir da relação (2) calculamos a capacitância. Em livros-texto que explicam a lei de Gauss [1], o mais simples é assumir uma densidade de carga elétrica uniforme no cilindro (infinito) interno e calcular a diferença de potencial. Desprezando efeitos das bordas, o campo elétrico é totalmente normal ao lado do cilindro interno, isto é radial, e igual a

$$\mathbf{E} = \frac{\sigma}{\varepsilon_0 r} \hat{\rho}. \quad (23)$$

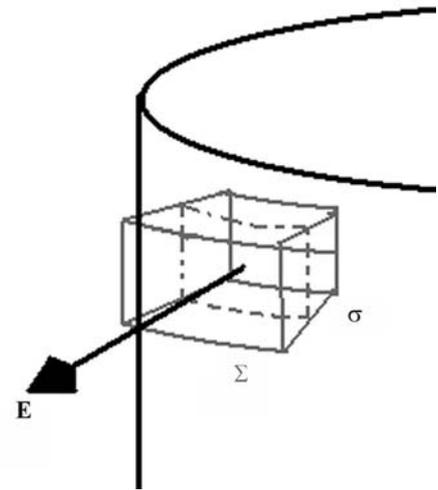


Figura 7 - Superfície Σ para integração utilizando a lei de Gauss e obtenção da densidade de carga σ sobre uma armadura.

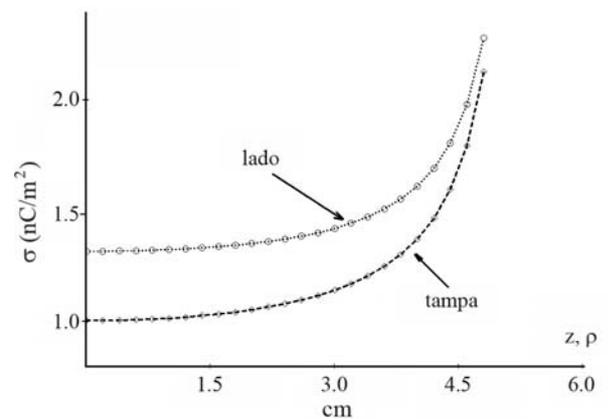


Figura 8 - Densidade de carga ao longo do lado do cilindro e ao longo de um raio da tampa.

Calculando-se a diferença de potencial e inserindo os resultados na Eq. (2) chega-se à capacitância na Eq. (1) para o comprimento L . No caso presente, a carga total é desconhecida enquanto que o potencial é dado. A carga total no cilindro interno é proporcional à área sob a curva na Fig. 7, ou mais exatamente

$$Q = 2\pi \left[\int_0^{R_{int}} \sigma_t(\rho) \rho d\rho + R_{int} \int_0^{L_{int}/2} \sigma_L(z) dz \right] \times 2 \quad (24)$$

onde σ_t e σ_L são as densidades de carga na tampa e nos lados respectivamente. A multiplicação por 2 vem do fato que o termo entre parênteses só fornece a carga em metade do cilindro interno.

Não é possível obter a Eq. (1) a partir da Eq. (24) por nenhum processo limite uma vez que no cálculo da Eq. (1) não existem tampas (os cilindros são infinitos). Isso se vê claramente na Eq. (1) desde que, se a razão R_{ex}/R_{in} for mantida constante, a capacitância

é a mesma. Podemos, entretanto, calcular as capacitâncias para as duas relações como função do comprimento externo (assumindo que $L_{ex} - L_{in}$ é uma constante). Se fizermos $R_{ex} = 10$ cm, $R_{in} = 5$ cm, $L_{ex} - L_{in} = 1$ cm o gráfico da Fig. 9 fornece as capacitâncias calculadas pela Eq. (1) e pela Eq. (24) como função de L_{ex} . O mesmo gráfico contém o caso $R_{ex} = 2$ cm, $R_{in} = 1$ cm. A curva cheia representa o resultado para a capacitância do caso infinito. Vemos que, para cada valor de raio, em apenas um valor de L_{ex} o resultado coincide com o caso infinito, e que há uma grande dependência (como se espera no caso finito) com o tamanho dos raios. Como o caso simulado tem tampa, é natural esperar esse tipo de dependência.

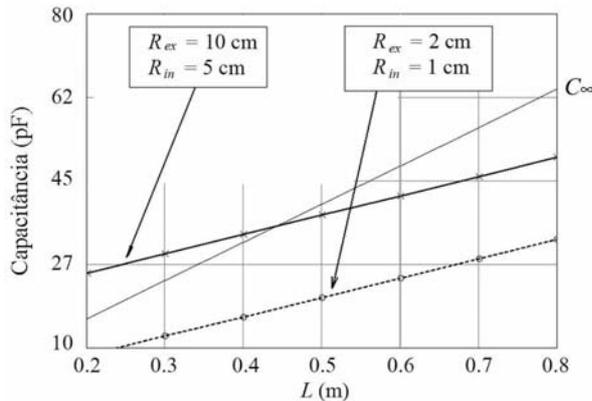


Figura 9 - Curva de capacitância como função do comprimento maior (o menor é inferior em 1 cm) calculada numericamente para a mesma razão entre raios. O caso do capacitor infinito é “degenerado” para essa razão. Para qualquer configuração de raio interno e externo, existe um ponto em que a capacitância coincide com o valor dado pelo modelo infinito.

7. Discussão final

Apresentamos aqui alguns resultados numéricos, juntamente com um código mínimo, que permite determinar várias quantidades elétricas no sistema de capacitores cilíndricos finitos. O algoritmo permite determinar a distribuição de potencial elétrico entre as armaduras e a partir dela o aluno interessado pode calcular diversas outras quantidades tais como o campo elétrico, a distribuição de carga elétrica, energia elétrica etc. Outras quantidades derivadas dessas primitivas podem igualmente ser determinadas. De posse de um *software* gráfico, o aluno pode também examinar a característica das soluções obtidas, bem como modificar o algoritmo utilizado para outras geometrias. Por exemplo, o caso de cilindros concêntricos mas serem iguais pode ser facilmente tratado com esse mesmo algoritmo através de algumas modificações simples. Já o caso excêntrico necessita de modificações maiores uma vez que a simetria cilíndrica é perdida.

Pretendemos aqui ilustrar o processo de tratamento numérico de casos reais, livres de aproximações analíticas ou outras simplificações. No estado atual de

desenvolvimento de recursos de computação numérica e diante da possibilidade de se utilizar *softwares* gratuitos para exibição de resultados gráficos, o tempo gasto na busca de uma solução numérica mais geral é bastante compensador se comparada aos esforços que muitas vezes se fazem na tentativa de obtenção de soluções analíticas, ainda que aproximadas. As soluções numéricas podem ilustrar vividamente também casos limites para os quais soluções analíticas sejam disponíveis ou demonstrar a limitação dessas últimas quando existe alguma quebra de simetria. Por meio desse exemplo particular de diferenças finitas, outras questões não abordadas no texto podem ser tratadas, tais como a otimização da memória disponível para armazenamento de soluções, as taxas de convergência da solução como função da quantidade de memória disponível e velocidade do processador e outras questões de interesse em física computacional.

Apêndice

Código em C++ que implementa solução para equação de Laplace para o problema do capacitor cilíndrico finito - modelo com tampas.

defs.h (Protótipo de funções em definição de classe::)

```
#include <math.h>
#include <stdio.h>
#include <iostream>
#include <vector>
using namespace std;
#define Pi 3.14159265358979323846e0

struct MALHA{
int Ngrid; //numero de pt.s ao longo de rho
double prec; // tolerância de cálculo
// (determina saída do código)
int Nrho, Nz; //numero de pontos em rho e em z
double del; //dens. radial e vert. da malha
int K,I; // Números K e I (ver eqs. 10)
void LeParametrosDaMalha
(int NN, double precisao);
};

//dimensões do cilindro
struct CILINDRO {
double Rext, Rint; // raio ext. e int.
double Lext, Lint; // comprimentos ext. e int.
double Vext, Vint; //Potenciais ext. e int.
MALHA Grid;
vector <vector <double>> U; //campo de pot.
void LePotenciais(double V1, double V2);
void LeGeometria (double R1, double R2,
double L1, double L2);
void GeraMalha(); //Gera malha para o pot. e
```

```

// aplica condições de fronteira fixas
double CalculaResiduo(int i, int k);
void SORLaplace(double w);
// w parâm. de relax.
// de acordo com o potencial
void EscreveU();
}

Subs.cpp (funções membras de MALHA e CILINDRO)::

#include "defs.h"

//Le parâmetros externos da malha
//N numero de pontos na malha
//prec :: precisão esperada para o cálculo
//      da solução
void MALHA::LeParametrosDaMalha (int NN,
double precisao) {
    Ngrid=NN;
    prec=precisao;
}

//Le parâmetros externos (potenciais)
void CILINDRO::LePotenciais
(double V1, double V2 ) {
    Vext=V2;
    Vint=V1;
}

//Le parâmetros geométricos do cilindro
//R1 é o raio interno
//R2 é o raio interno (R2 >R1)
//L é o comprimento (L>0)
void CILINDRO::LeGeometria (double R1,
double R2, double L1, double L2) {
    Rint=R1;
    Rext=R2;
    Lint=L1;
    Lext=L2;
}

void CILINDRO::GeraMalha() {
    int k, i, I, K;
    Grid.del = Rext/(double)Grid.Ngrid;
    //Densidade da malha
    //calculo de inteiros aproximados
    //para dimensionamento da malha
    Grid.Nrho = Grid.Ngrid;
    Grid.Nz = (int)(Lext/(2.*Grid.del));
    // aloca memória para a matriz do
    // campo de pot. na malha
    U.resize(Grid.Nrho);

    for(i=0;i<Grid.Nrho;i++)
        U[i].resize(Grid.Nz);
    //Condições de Neumann para o cilindro ext.
    for(k=0;k<Grid.Nz;k++)
        U[Grid.Nrho-1][k]=Vext;
    //Condições de Neumann para a tampa ext.
    for(i=0;i<Grid.Nrho;i++)
        U[i][Grid.Nz-1]=Vext;
    //Condições para o cilindro interno
    //Primeiro calcula os números K e I
    Grid.I=(int)(Rint/Grid.del);
    Grid.K=(int)(Lint/(2.*Grid.del));
    I=Grid.I;
    K=Grid.K;
    //condição de Neumann para o cilindro int.
    for(k=0;k<K;k++)
        U[I][k]=Vint;
    //condição de Neumann para a tampa int.
    for(i=0;i<I;i++)
        U[i][K]=Vint;

    //Gera vals. iniciais para o campo de pot.
    // baseado nos potencias das armaduras
    //região A
    for(i=0;i<I;i++)
        for(k=K+1;k<Grid.Nz-1;k++)
            U[i][k]=.5*(Vext+Vint);
    //região B
    for(i=I+1;i<Grid.Nrho-1;i++)
        for(k=0;k<Grid.Nz-1;k++)
            U[i][k]=.5*(Vext+Vint);
    //região interior ao cilindro int.
    for(i=0;i<I;i++)
        for(k=0;k<K;k++)
            U[i][k]=Vint;
}

//Calcula residuo da eq. de Laplace
// no ponto (i,k) da malha, Eq. (??)
double CILINDRO::CalculaResiduo
(int i, int k) {
    double gm=1.+5/(double)i;
    double gn=1.-5/(double)i;
    return U[i+1][k]*gm + U[i-1][k]*gn-4.*U[i][k]
        + U[i][k+1] + U[i][k-1];
}

void CILINDRO::EscreveU() {
    int i,k;
    FILE *fp;
    fp = fopen("U.dat", "w"); //open file to write
    for(i=0;i<Grid.Nrho;i++) {
        for(k=0;k<Grid.Nz;k++) {
            fprintf(fp, "%5.3f ",U[i][k]);
        }
        fprintf(fp, "\n");
    }
}

```

```

fclose(fp);
}

//Resolve a eq. de Laplace pelo método SOR
//res: variável auxiliar para o resíduo
//resmax: máximo resíduo
void CILINDRO::SORLaplace(double w) {
int i,k;
double res, resmax;
do {
    resmax=0.;
    // aplica condição de Dirichlet dU/dRho=0
    // para simetria axial (Eq. 12)
    for(k=Grid.K+1;k<Grid.Nz-1;k++)
        U[0][k]=U[1][k];
    // aplica condição de Dirichlet para
    // fronteira dU/dz=0 (Eq. 12)
    for(i=Grid.I+1;i<Grid.Nrho-1;i++)
        U[i][0]=U[i][1];
    //Iteração para Região A
    for(i=1;i<=Grid.I;i++)
    for(k=Grid.K+1;k<Grid.Nz-1;k++) {
        res=CalculaResiduo(i,k);
        U[i][k]=U[i][k] + .25*w*res;
        if (fabs(res)>resmax) resmax=fabs(res);
    }

    //Iteração para Região B
    for(i=Grid.I+1;i<Grid.Nrho-1;i++)
    for(k=1;k<Grid.Nz-1;k++) {
        res=CalculaResiduo(i,k);
        U[i][k]=U[i][k]+.25*w*res;
        if (fabs(res)>resmax) resmax=fabs(res);
    }
}
while (fabs(resmax)>Grid.prec);
cout << "Terminou..." << endl;
EscreveU();
}

```

Referências

- [1] D. Halliday and R. Resnick, *Fundamentals of Physics* (J. Wiley & Sons, Nova Iorque, 1997), 4^a ed., cap. 27, p. 743.
- [2] J.D. Jackson, *Classical Electrodynamics* (J. Wiley & Sons, Nova Iorque, 1999), 3^a ed., cap. 3, p. 95.
- [3] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C, the Art of Scientific Computing* (Cambridge University Press, Cambridge, 1997), 2^a ed., p. 863.
- [4] <http://gcc.gnu.org/>.