

Métodos computacionais em física

Projeto2-Relatório

Pedro Mariano Marques Mendes, 9865398

Maio 2020

1 Introdução

O objetivo deste projeto é resolver a Equação de Poisson bidimensional numericamente. Em especial para dois casos: 1. Uma carga pontual numa caixa com as paredes aterradas. 2. Uma carga pontual entre um capacitor.

A Equação de Poisson:

$$\nabla^2 V(\vec{r}) = -\frac{\rho(\vec{r})}{\epsilon_0} \quad (1)$$

Onde ϵ_0 é a constante de permissividade do vácuo, ρ é a densidade volumétrica de carga, V é o potencial elétrico e ∇^2 é o laplaciano.

Em seguida o campo elétrico pode ser obtido da seguinte forma:

$$\vec{E} = -\vec{\nabla} V \quad (2)$$

Assim é possível observar o comportamento do campo elétrico nos casos citados. Em especial será verificado se o comportamento do campo elétrico numa condição bidimensional é equivalente a uma situação tridimensional. No caso tridimensional, o campo elétrico gerado por uma carga pontual é dado por:

$$\vec{E} = \frac{1}{4\pi\epsilon_0} \frac{\hat{r}}{r^2} \quad (3)$$

Para resolver esse problema numericamente iremos utilizar o método de relaxação de Jacobi no programa Octave [1].

2 Descrição da simulação numérica e resultados obtidos.

A simulação é feita por meio do método de relaxação de Jacobi. O método consiste em calcular o potencial num determinado ponto como uma média simples do potencial nos pontos ao redor mais a contribuição da carga a cada passo.

Retomando a equação 1:

$$\nabla^2 V(\vec{r}) = -\frac{\rho(\vec{r})}{\epsilon_0} \quad (4)$$

Como aproximação para a segunda derivada:

$$\frac{\partial^2 V(x, y)}{\partial x^2} \simeq \frac{V(x + \Delta x, y) - 2V(x, y) + V(x - \Delta x, y)}{(\Delta x)^2} \quad (5)$$

$$\frac{\partial^2 V(x, y)}{\partial y^2} \simeq \frac{V(x, y + \Delta y) - 2V(x, y) + V(x, y - \Delta y)}{(\Delta y)^2} \quad (6)$$

A soma de 5 e 6 deve ser igual ao lado direito da equação 1. Assumindo $\Delta x = \Delta y$ e passando o termo a direita da 1 para a esquerda ficamos com:

$$\frac{1}{(\Delta x)^2} \left[-4V(x, y) + V(x + \Delta x, y) + V(x - \Delta x, y) + V(x, y + \Delta y) + V(x, y - \Delta y) \right] + \frac{\rho(x, y)}{\epsilon_0} = 0 \quad (7)$$

Passando o termo com $V(x, y)$ para o outro lado, ficamos com:

$$\frac{4V(x, y)}{(\Delta x)^2} = \frac{1}{(\Delta x)^2} \left[V(x + \Delta x, y) + V(x - \Delta x, y) + V(x, y + \Delta y) + V(x, y - \Delta y) \right] + \frac{\rho(x, y)}{\epsilon_0} \quad (8)$$

Por fim:

$$V(x, y) = \frac{1}{4} \left[V(x + \Delta x, y) + V(x - \Delta x, y) + V(x, y + \Delta y) + V(x, y - \Delta y) + \frac{\rho(x, y) \Delta x^2}{\epsilon_0} \right] \quad (9)$$

Discretizando isso para utilizar num problema computacional:

$$V(i, j) = \frac{1}{4} \left[V(i + 1, j) + V(i - 1, j) + V(i, j + 1) + V(i, j - 1) + \frac{\rho(i, j) \Delta x^2}{\epsilon_0} \right] \quad (10)$$

No método de relaxação de Jacobi é feita uma nova aproximação $(n + 1)$ a partir da antiga (n) segundo a equação 10. Dessa forma a nova aproximação para o potencial em função do potencial anterior é dada por:

$$V_{n+1}(i, j) = \frac{1}{4} \left[V_n(i + 1, j) + V_n(i - 1, j) + V_n(i, j + 1) + V_n(i, j - 1) + \frac{\rho(i, j) \Delta x^2}{\epsilon_0} \right] \quad (11)$$

Essa é a fórmula que será utilizada para calcular o potencial em todo o espaço.

Na solução computacional o espaço é discretizado e modelado como um grid de pontos. A carga pontual é modelada colocando-a num ponto do grid. Dessa forma $\rho(i, j) \neq 0$ apenas no ponto onde a carga está.

Importante ressaltar que os pontos onde estão as placas do capacitor não são atualizados para que estejam sempre com o mesmo potencial predeterminado.

A cada passo o programa avalia a diferença nos valores calculados entre o passo n e o passo $n + 1$, se a diferença for abaixo de uma certa precisão o cálculo é dado como pronto, ou seja, se estiver sendo calculado um novo potencial ($n + 1$) muito semelhante com o anterior (n) não há porque continuar calculando $n + 2, 3, 4, \dots$. A diferença entre ($n + 1$) e (n) é dada pela soma:

$$S = \sum_{x,y} |V_{n+1} - V_n| \quad (12)$$

Onde $\sum_{x,y}$ representa a soma sobre todo o espaço. Se a soma S for menor do que a precisão o programa encerra o cálculo.

Um exemplo dos gráficos gerados, no caso com $Q_{eps0} = 3$ e uma definição espacial de $5cm$ num espaço total de $1m^2$, uma precisão desejada de 10^{-3} e um número máximo de 1000 iterações:

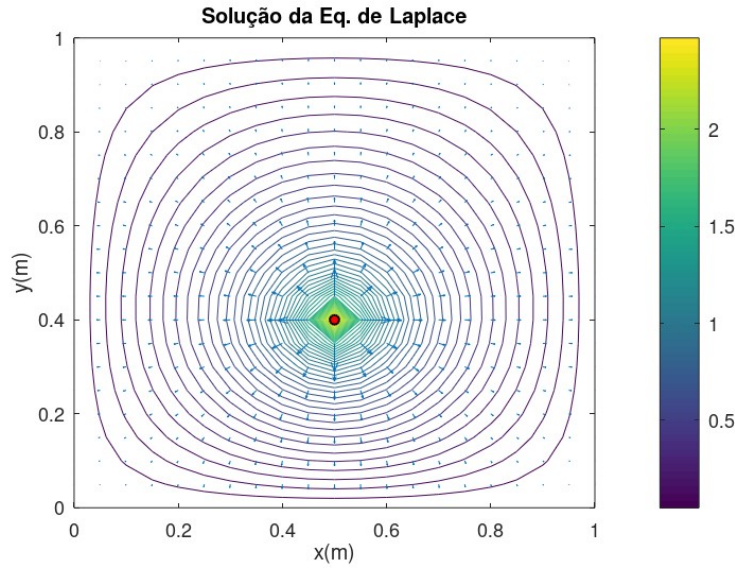


Figure 1: Curvas de nível do potencial e campo vetorial elétrico para a carga pontual.

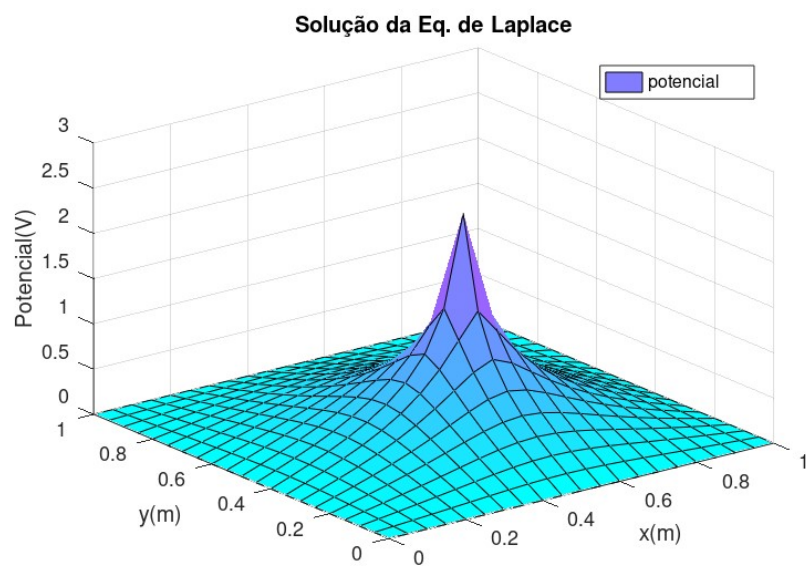


Figure 2: Superfície do potencial para a carga pontual.

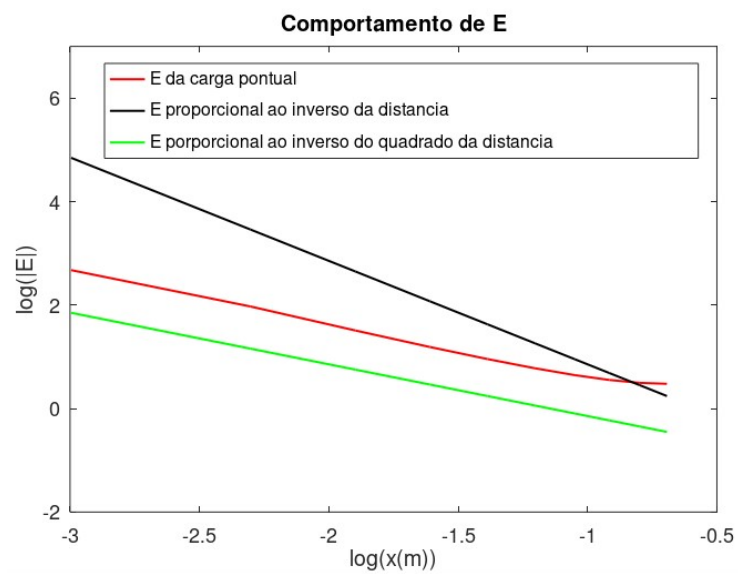


Figure 3: Decaimento do campo elétrico com a distância.

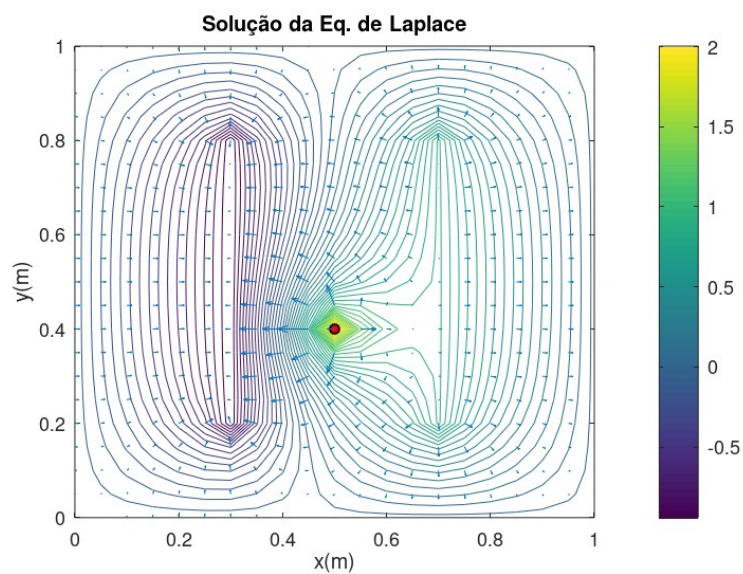


Figure 4: Curvas de nível do potencial e campo vetorial elétrico para a carga pontual e o capacitor.

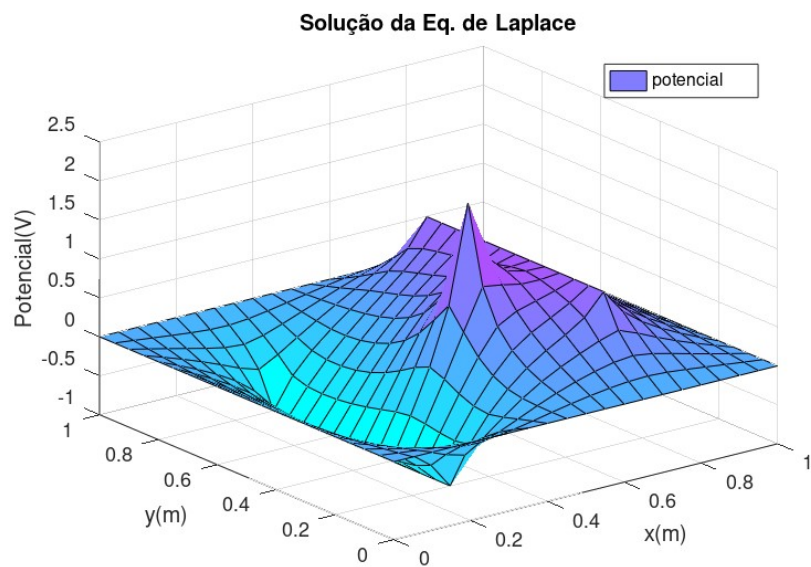


Figure 5: Superfície do potencial para a carga pontual e o capacitor.

No gráfico 3 a curva em vermelho representa o módulo do campo elétrico

obtido para a carga pontual. É o campo logo a direita da carga até a parede a direita como mostrado na figura 6.

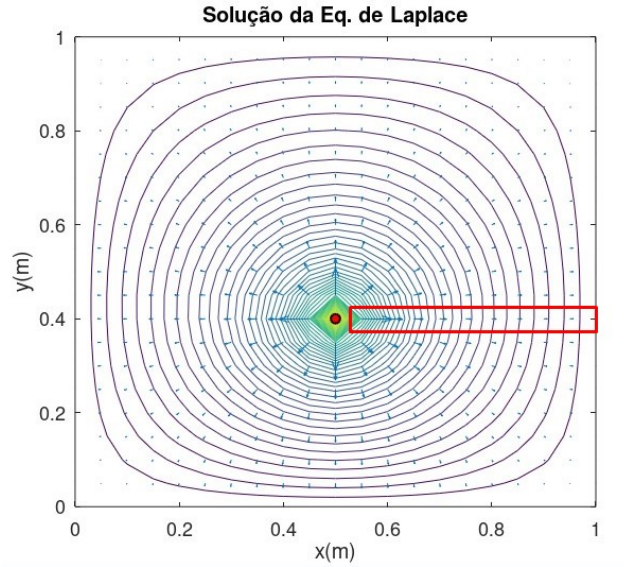


Figure 6: Região que estamos considerando do campo elétrico.

A curva preta representa um campo que cai com r^{-2} como esperado para o campo elétrico segundo a equação 3 e a curva verde um campo que cai com r^{-1} . Ambas são para comparação com a curva vermelha obtida. Aplicando \log nos dois lados da equação 3:

$$\log(E) = -\log(4\pi\epsilon_0) - 2 \cdot \log(r) \quad (13)$$

E podemos ver que o coeficiente angular representa a potência com que o campo elétrico cai. O coeficiente linear é dado por constantes apenas e não é do interesse desse projeto. Pensando nisso é possível observar que a inclinação na curva do campo gerado pela carga pontual se assemelha muito mais a curva verde. Por se tratar de um caso bidimensional, a carga não ser realmente pontual e sim estar "distribuída" numa área Δx^2 e as paredes serem aterradas, o campo elétrico da carga cai com r^{-1} ao invés de r^{-2} como se era esperado de um campo elétrico. Em especial é possível notar o efeito da parede aterrada na extremidade direita da curva gerada pela carga.

As definições utilizadas para gerar as figuras apresentadas (uma definição espacial de $5cm$ num espaço total de $1m^2$, uma precisão desejada de 10^{-3} e um número máximo de 1000 iterações) são adequadas tanto no aspecto visual informativo quanto no quesito de tempo de execução. Mostram claramente o comportamento do potencial e do campo elétrico de forma suave, curvas sem

bicos e demora por volta de um minuto de execução. Alterando a definição espacial para $2cm$ o tempo de execução aumenta consideravelmente para 15 : 30 minutos gerando figuras não muito diferentes do caso com $5cm$. Além disso nessa configuração com $2cm$ de definição espacial os vetores de campo elétrico aparecem muito pequenos nos gráficos 1 e 4 tornando difícil sua visualização.

3 Manual do usuário

O programa tem como objetivo resolver a Equação de Poisson e calcular o potencial e o campo elétrico dada uma carga pontual num determinado local do espaço.

A primeira parte do programa consiste nas condições iniciais do problema que podem ser alteradas pelo usuário. Entre as linhas 13 e 21 é possível alterar as condições de posição da carga, valor da carga sobre ϵ_0 , tamanho do quadrado, definição espacial, critério de convergência, número máximo de iterações e o potencial nas placas do capacitor.

As configurações de tamanho do quadrado, definição espacial, critério de convergência e número máximo de iterações alteram diretamente o tempo de execução do programa, as outras variáveis não afetam o tempo de execução.

Como o espaço é discretizado o programa procurará o ponto espacial mais próximo do esperado para posicionar a carga.

O critério de convergência estabelece uma condição para considerar que o resultado está próximo o suficiente do resultado final, significa essencialmente a diferença no potencial entre a iteração n e a iteração $n + 1$ calculada de acordo com a equação 12. Quanto menor o critério mais próximas elas terão de ser para o algoritmo considerar resolvido o problema. Caso seja escolhida uma precisão muito pequena temos uma segunda medida que é o número máximo de iterações para o programa considerar o problema resolvido.

Em seguida o algoritmo primeiro faz a parte sem o capacitor: Ajusta as condições do problema, calcula o potencial e plota três gráficos. Um gráfico em duas dimensões que apresenta curvas de nível do potencial elétrico e o campo elétrico por meio de um campo vetorial (figura 1), um segundo gráfico, esse tridimensional do potencial como eixo z em função de x e y (figura 2) e por último um gráfico log-log do módulo do campo elétrico em função da distância (figura 3). Faz os cálculos novamente para o caso com as placas e plota os dois primeiros gráficos para esse caso (figuras 4 e 5).

Além disso o programa também calcula o tempo total de execução e o apresenta na área de comando ao final. Como dito, para as configurações de precisão espacial igual a $0.05m$, um tamanho total de $1m$, uma precisão desejada de 10^{-3} e um número máximo de 1000 iterações o tempo médio para realizar o cálculo e plotar os gráficos é de 1 minuto.

4 Conclusão

O método é adequado para resolver os problemas desejados, chegando ao resultado esperado sem um tempo de execução demasiadamente alto. Além disso apresenta gráficos claros que permitem analisar o potencial e o campo elétrico em todo o espaço bidimensional.

Qualitativamente o campo elétrico se comporta como esperado para as placas e para a carga. Em especial é interessante notar que o programa nos fornece o campo elétrico na região de borda do capacitor, que apresenta uma grande dificuldade de ser calculado analiticamente. Quantitativamente vemos que o campo elétrico gerado pela carga decai proporcionalmente a r^{-1} que é substancialmente diferente do esperado para uma carga pontual num espaço tridimensional.

References

- [1] John W. Eaton, David Bateman, Søren Hauberg, and Rik Wehbring. *GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations*, 2020. URL <https://www.gnu.org/software/octave/doc/v5.2.0/>.