

A Equação de Poisson

Potenciais elétricos e distribuições de carga

GUILHERME DAMACENA DE CASTRO, 11224150

Instituto de Física, Universidade de São Paulo - SP

Maio, 2022

Resumo

A Equação de Poisson tem aplicações notáveis na física. Ela surge naturalmente no estudo de uma variedade de sistemas, como no estudo do calor que flui em uma barra sólida ou no estudo de potenciais elétricos gerados por cargas estáticas. Esse segundo caso é de particular interesse, pois é muito útil para a montagem de equipamentos científicos que exigem controle fino de campos elétricos e magnéticos. Neste trabalho, é apresentado e discutido um método computacional de solução da equação de Poisson aplicada à eletrostática. Ele é testado e validado para aplicações científicas e didáticas com inúmeros exemplos.

I. INTRODUÇÃO

O Eletromagnetismo de Maxwell é um dos campos da física mais amplamente estudados e bem estabelecidos da física. As chamadas **Leis de Maxwell** descrevem o comportamento dos campos eletromagnéticos. Elas são descritas nas equações (1), onde \vec{E} e \vec{B} são os campos elétrico e magnético, respectivamente, ρ é a densidade de cargas elétricas, \vec{J} é a densidade de correntes elétricas, ϵ_0 é a constante de permissividade do vácuo, μ_0 é a constante de permeabilidade do vácuo, \vec{r} é a posição no espaço onde se mede os campos e t é o tempo a partir de $t = 0s$.

$$\begin{cases} \vec{\nabla} \cdot \vec{E}(\vec{r}, t) = \frac{\rho(\vec{r}, t)}{\epsilon_0} \\ \vec{\nabla} \cdot \vec{B}(\vec{r}, t) = 0 \\ \vec{\nabla} \times \vec{E}(\vec{r}, t) = -\frac{\partial}{\partial t} \vec{B}(\vec{r}, t) \\ \vec{\nabla} \times \vec{B}(\vec{r}, t) = \mu_0 \vec{J}(\vec{r}, t) + \mu_0 \epsilon_0 \frac{\partial}{\partial t} \vec{E}(\vec{r}, t) \end{cases} \quad (1)$$

O teorema de Helmholtz garante que os campos elétrico e magnético podem ser descritos em termos de potenciais $V(\vec{r}, t)$ e $\vec{A}(\vec{r}, t)$ através das equações (2) [1]. V é o chamado potencial elétrico, enquanto \vec{A} é o potencial vetor.

$$\begin{cases} \vec{E} = -\vec{\nabla}V - \frac{\partial}{\partial t} \vec{A} \\ \vec{B} = \vec{\nabla} \times \vec{A} \end{cases} \quad (2)$$

Para campos estáticos, a derivada temporal na expressão para o campo elétrico \vec{E} na equação (2) se anula. Ainda, na ausência de cargas em movimento (correntes), o campo magnético é nulo. Sendo assim, usando o divergente do campo magnético (dado nas equações (1)) e a expressão potencial do campo elétrico, temos a equação (3), que é

uma equação de poisson que descreve o potencial em termos da distribuição de cargas no espaço.

$$\nabla^2 V = -\frac{\rho}{\epsilon_0} \quad (3)$$

Note que a equação (2) só é válida para casos em que $\vec{A}(\vec{r}, t)$ é nulo ou estático $A(\vec{r}, t) = \vec{A}(\vec{r})$. Solucionar a equação de poisson com uma distribuição de cargas analiticamente pode ser muito complicado para um diferentes condições de contorno. Dadas as muitas aplicações na física de sistemas em que é necessário conhecer a forma dos campos eletromagnéticos (aceleradores de partículas, por exemplo), é importante se capaz de solucionar a equação (3) numericamente.

Uma técnica que utiliza o Método de Relaxação de Jacobi é apresentada e testada através da comparação com soluções analíticas conhecidas. Além disso, são mostradas as soluções para casos interessantes: Um capacitor de placas paralelas (em que são visíveis os efeitos de borda evitados em cursos de graduação usuais) e diferentes situações com cargas pontuais.

II. DESCRIÇÃO DO MÉTODO NUMÉRICO

Por razões de simplicidade, o trabalho desenvolvido calculou o potencial em um plano (2D). No entanto, o processo pode ser facilmente estendido para a situação tridimensional.

I. Matriz das posições

O primeiro passo é preparar o espaço onde o potencial é calculado. Isso é feito em uma malha (matriz) de compri-

mento L_x no eixo x e L_y no eixo y . Cada eixo é dividido em passos de comprimento Δx ou Δy . Uma figura ilustrativa da malha computacional é apresentada na figura 1.

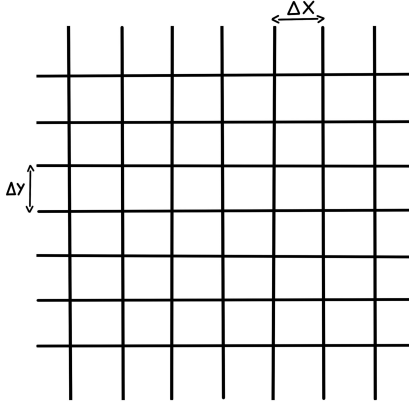


Figura 1: Malha espacial onde é calculado o potencial V .

Cada nodo (encontro entre retas horizontais e verticais) na matriz das posições é uma posição efetiva de coordenadas (x,y) a partir da origem (colocada no canto inferior esquerdo por conveniência). Note que a maneira como cada nodo é referenciado depende da linguagem de programação escolhida. Como os eixos x e y são discretizados e igualmente espaçados, é possível referenciar cada posição x através de um índice i . Equivalentemente, cada posição y é referenciada com um índice j . Assim, definimos a origem como o ponto tal que $i = 0$ e $j = 0$. Sendo assim, vale que as coordenadas são determinadas pelas equações (4). Isso permite que o valor do potencial elétrico seja determinado pelos índices: $V(x, y) \rightarrow V(i, j)$.

$$\begin{cases} x_i = i \cdot \Delta x \\ y_j = j \cdot \Delta y \\ 0 \leq i \leq \frac{L_x}{\Delta x} \\ 0 \leq j \leq \frac{L_y}{\Delta y} \end{cases} \quad (4)$$

É importante notar que, além do potencial, a densidade de cargas ρ também é descrita sobre a malha. Assim, também vale que $\rho(x, y) = \rho(i, j)$. Na prática, isso acontece com qualquer grandeza que dependa de (x, y) . Por conveniência, é assim que os potenciais e campos são indexados neste trabalho. Vale ainda notar que as notações de variável seguinte como $x + \Delta x$ podem ser substituídas pelo índice $i + 1$. O mesmo vale para y .

II. Condições de contorno

II.1 Bordas

Em geral, sempre será necessário escolher os potenciais (uma explicação mais detalhada é disposta na subseção **O Método de Jacobi**).

Sendo assim, os potenciais nas bordas $V(0, j)$ (borda da esquerda), $V(\frac{L_x}{\Delta x}, j)$ (borda da direita), $V(i, 0)$ (borda de baixo) e $V(i, \frac{L_y}{\Delta y})$ (borda de cima) devem ser determinados.

II.2 Interior

Em muitos casos, é interessante, também, adicionar condições de contorno no interior da matriz de posições. Há uma peculiaridade importante para isso. Como os eixos são discretos, condições de contorno só podem ser adicionadas em coordenadas (x, y) tais que x e y cumpram as condições dadas nas equações (4).

III. O Método de Jacobi

Partindo da equação (3), é possível escrever o laplaciano 2D em coordenadas cartesianas como $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Assim, vale:

$$\frac{\partial^2}{\partial x^2} V(x, y) + \frac{\partial^2}{\partial y^2} V(x, y) = -\frac{\rho(x, y)}{\epsilon_0}$$

As derivadas parciais de segunda ordem em uma variável u de uma função $f(u, \vec{x})$, onde o vetor \vec{x} é um vetor de várias possíveis variáveis genéricas $x \in [x_1, x_2, \dots, x_n]$, pode ser aproximada como dito pela equação (5) que leva em conta diferenças finitas [2].

$$\frac{\partial^2}{\partial u^2} f(u, \vec{x}) \approx \frac{f(u + \Delta u, \vec{x}) - 2f(u, \vec{x}) + f(u - \Delta u, \vec{x})}{(\Delta u)^2} \quad (5)$$

Ao aplicar a aproximação de segunda derivada na equação de poisson, obtemos a solução aproximada para o potencial dada pela equação (6).

$$V(x, y) = L(x, y) + D(x, y) \quad (6)$$

Onde

$$\begin{cases} L(x, y) = \gamma((\Delta y)^2 \Delta V_x + (\Delta x)^2 \Delta V_y) \\ \Delta V_x = V(x + \Delta x, y) + V(x - \Delta x, y) \\ \Delta V_y = V(x, y + \Delta y) + V(x, y - \Delta y) \end{cases}$$

$$D(x, y) = \gamma(\Delta x \Delta y)^2 \frac{\rho(x, y)}{\epsilon_0}$$

$$\gamma = \frac{1}{2 \cdot ((\Delta x)^2 + (\Delta y)^2)}$$

Algo interessante acontece aqui. A solução pode ser dividida entre $L(x, y)$, que é a solução da equação de laplace (quando $\rho = 0$) e $D(x, y)$. Vale notar que a solução $D(x, y)$ é estática! Isso significa que ela pode ser tratada como uma condição de contorno. Já a solução $L(x, y)$ mostra que o valor do potencial em um ponto (x, y) da malha é dado por uma espécie de média ponderada dos valores do potencial ao redor de x, y . Isso significa que, para que o valor do potencial em um ponto da malha seja calculado corretamente através da expressão de $V(x, y)$, os valores dos potenciais ao redor devem estar corretos. Portanto, é necessário chutar uma solução inicial V_0 (normalmente $V_0 = 0V$) e calcular V_1 usando $V(x, y)$. Calculando V_1 , que é uma melhor aproximação do potencial real, é possível utilizá-lo para calcular uma solução V_2 . Usando esse raciocínio, é possível obter uma solução iterativa em termos de i e j dada pela equação (7).

$$V_{k+1}(i, j) = L_k(i, j) + D(i, j) \quad (7)$$

Onde

$$\begin{cases} L_k(i, j) = \gamma((\Delta y)^2(\Delta V_k)_x + (\Delta x)^2(\Delta V_k)_y) \\ (\Delta V_k)_x = V_k(i+1, j) + V_k(i-1, j) \\ (\Delta V_k)_y = V_k(i, j+1) + V_k(i, j-1) \end{cases}$$

$$D(i, j) = \gamma(\Delta x \Delta y)^2 \frac{\rho(i, j)}{\epsilon_0}$$

Com a solução iterativa, as soluções V_k devem ser calculadas tomando o cuidado de manter as condições de contorno inalteradas. Isso inclui a solução estática $D(i, j)$.

Ainda é possível definir um critério de convergência para o potencial, afinal, é esperado que para um determinado número de iterações, as mudanças no potencial passem a ser irrelevantes. Para isso, é definida a grandeza δ , que é a soma das mudanças absolutas para o potencial em cada ponto da malha. Assim, definido um erro máximo ϵ , é possível dizer que o potencial V converge com erro ϵ na repetição N se $\delta_N < \epsilon$.

$$\delta_n = \sum_{i,j} |V_n(i, j) - V_{n-1}(i, j)| \quad (8)$$

A escolha de ϵ adequada é um passo importante da simulação. Uma discussão sobre isso é feita na seção III.

Após a obtenção do potencial, é possível obter, através da equação (2), o campo elétrico. Em coordenadas cartesianas (em 2D, como o problema que está sendo tratado), o vetor gradiente é escrito como $\vec{\nabla} = \hat{x} \frac{\partial}{\partial x} + \hat{y} \frac{\partial}{\partial y}$. Uma aproximação para as primeiras derivadas parciais discretas, novamente utilizando o método das diferenças finitas, é dada na equação (9) [2].

$$\frac{\partial}{\partial u} f(u, \vec{x}) \approx \frac{f(u + \Delta u, \vec{x}) - f(u - \Delta u, \vec{x})}{2\Delta u} \quad (9)$$

Assim, é possível descrever as componentes E_x e E_y do campo elétrico $\vec{E} = E_x \hat{x} + E_y \hat{y}$ através das equações (10). O leitor pode ter notado que a forma como as derivadas primeiras foram aproximadas assumem a existência de pontos anteriores e seguintes na variável de derivação. Isso não é um problema para quaisquer pontos interiores à malha, mas é um problema para os pontos de borda. Note que, não é possível calcular as componentes x do campo elétrico nas bordas esquerda e direita da malha, assim como não é possível calcular as componentes y do campo nas bordas superior e inferior da malha através da equação (9). Nas bordas, é necessário usar diferentes aproximações de derivada que permitem obter as equações (11).

$$\begin{cases} E_x(i, j) \approx -\frac{V(i+1, j) - V(i-1, j)}{2\Delta x} \\ E_y(i, j) \approx -\frac{V(i, j+1) - V(i, j-1)}{2\Delta y} \end{cases} \quad (10)$$

$$\begin{cases} E_x^e(i, j) \approx -\frac{V(i+1, j) - V(i, j)}{\Delta x} \\ E_x^d(i, j) \approx -\frac{V(i, j) - V(i-1, j)}{\Delta x} \\ E_y^c(i, j) \approx -\frac{V(i, j) - V(i, j-1)}{\Delta y} \\ E_y^b(i, j) \approx -\frac{V(i, j+1) - V(i, j)}{\Delta y} \end{cases} \quad (11)$$

As letras sobrescritas nas componentes do campo elétrico nas bordas representam e: esquerda; d: direita; c: cima; b: baixo.

Com os passos apresentados até então, é possível resolver a equação de Poisson para uma conjunto imenso de densidades de carga e condições de contorno. Infelizmente não é possível abordar muitos casos específicos aqui, mas uma porção deles é discutida na seção a seguir.

III. DISCUSSÃO E RESULTADOS

Todo o desenvolvimento até aqui descrito foi usado para construir um algoritmo em Python 3 capaz de solucionar e gerar figuras das soluções da equação de Poisson para diversos sistemas. O algoritmo, assim como interessantes animações da relaxação de Jacobi estão disponíveis neste link. Além disso, no apêndice A, há um manual do usuário para o citado algoritmo.

Uma boa maneira de começar é testando um caso bastante simples.

I. Potencial Constante

No caso em que a equação de laplace é resolvida para um conjunto de condições de contorno completamente constantes e iguais em todas as bordas, o potencial é completamente constante [3].

As características gerais da malha escolhida são, por conveniência, $L_x = L_y = 1m$ e os passos $\Delta x = \Delta y = 0.05m$.

Para o caso do potencial constante, consideramos $\rho = 0 \frac{C}{m^3}$. Isso significa que estamos no caso em que a equação de Poisson se reduz à equação de Laplace.

As condições de contorno para este caso são bastante simples. Basta inserir $V = 1V$ nas bordas da malha. Para a condição inicial do potencial no interior da malha, o palpite usual é utilizado $V_0 = 0V$.

Uma pergunta importante surge naturalmente na hora de fazer a simulação. Qual deve ser a escolha do erro máximo ϵ ? A resposta depende do nível de precisão que é necessária para que se realize o que deseja com o resultado da simulação. Por exemplo, se o que se deseja é observar o comportamento geral do potencial na malha, erros maiores podem ser permitidos. No entanto, se os valores do potencial ponto a ponto são necessários para realizar simulações numéricas mais profundas, como a previsão do deslocamento de uma partícula carregada em um plano, o erro deve ser menor. É claro que deve haver um limite de tolerância para que não sejam aceitas simulações com erros gritantes (que fazem com que o resultado obtido não seja físico). Para lidar com essas decisões, foi criado um procedimento de escolha para os ϵ em cada situação.

1.1 A escolha de ϵ

Na primeira vez que é realizada a simulação, escolhe-se um número máximo de iterações muito grande e um epsilon muito pequeno. O objetivo é garantir a convergência da simulação, afinal, desejamos saber quão rápido uma situação quase estática (que não muda muito com cada iteração) é atingida. Durante a simulação, os valores dos δ_n , assim como as matrizes V , são armazenados em listas. Isso permite observar o comportamento da amplitude dos δ em função do número da iteração.

Assim, a escolha de epsilon pode ser feita de forma que sejam garantidas iterações suficientes para que o comportamento seja quase estático (que pode ser comprovado observando a evolução do perfil de V) e de forma que a velocidade com a qual δ diminui a cada iteração seja aproximadamente nula (δ seja quase constante).

Esse procedimento é demonstrado a seguir com o caso do potencial constante

Os primeiros parâmetros escolhidos foram $N_{max} = 10000$ e $\epsilon = 10^{-6}V$, onde N_{max} representa o número máximo de iterações realizadas (caso δ nunca seja inferior a ϵ).

Ao fazer isso, o que é observado é que a simulação para na iteração 2683. Isso significa que ela atingiu $\delta_{2683} < \epsilon$. O primeiro passo é visualizar a solução, mostrada na figura 2, e julgar se ela é coerente.

É fácil ver que, nessa situação, o resultado é coerente com o esperado: um potencial completamente constante. O próximo passo é verificar se o resultado realmente converge apenas para uma iteração grande. Analisando cuidadosamente iteração por iteração, é possível observar que até a iteração 2672, o resultado ainda não havia convergido completamente. Isso é evidenciado pela imagem na figura 3.

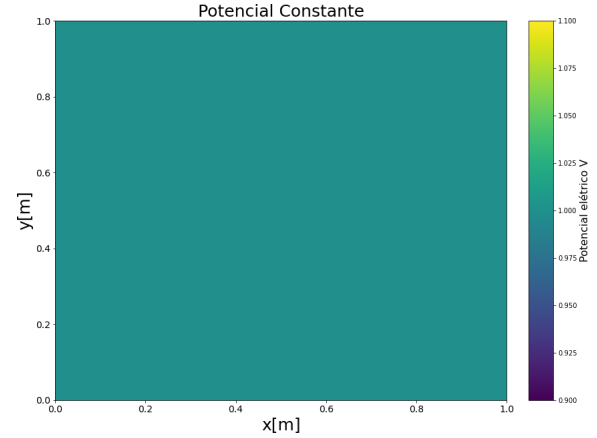


Figura 2: Simulação do potencial elétrico 2D com nenhuma distribuição de cargas para uma malha com lados $L_x = L_y = 1m$ dividida em passos $dx = dy = 0.05m$ cujas condições de contorno foram $V_{bordas} = 1V$ e o chute inicial para o potencial no interior foi $V_0 = 0V$.

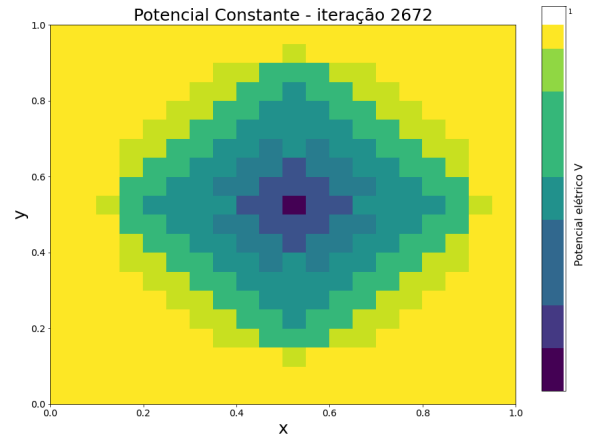


Figura 3: Iteração 2672 da simulação do potencial elétrico 2D com nenhuma distribuição de cargas para uma malha com lados $L_x = L_y = 1m$ dividida em passos $dx = dy = 0.05m$ cujas condições de contorno foram $V_{bordas} = 1V$ e o chute inicial para o potencial no interior foi $V_0 = 0V$.

Isso significa que é necessário passar da iteração $n = 2672$ para garantir a convergência. Em seguida, é ne-

cessário analisar a evolução de δ com as iterações (figura 4). É possível identificar a aproximação assintótica do delta de uma constante próxima de 0 para um número de iterações entre 2000 e 3000, o que é coerente com o resultado obtido apenas buscando a coerência do resultado da simulação com o previsto pela teoria.

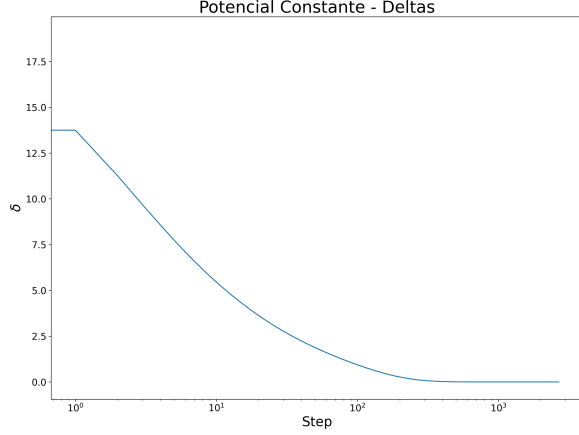


Figura 4: Evolução dos deltas com o aumento do número de iterações para a simulação do potencial constante.

Sendo assim, o ϵ ideal pode ser escolhido como o delta da iteração em que a simulação convergiu.

$$\epsilon_{optimal} = \delta_{2673} \approx 1.9e - 14V$$

Esse procedimento para a escolha de $\epsilon_{optimal}$ foi descrito extensivamente para o caso do potencial constante. Como o procedimento é padronizado, é desnecessário que seja repetido em futuras análises, mas é importante que esteja claro que escolher um bom conjunto ϵ e N_{max} é essencial para garantir o melhor funcionamento da simulação.

II. Capacitor de Placas Paralelas

Aqui, ainda não há distribuição de cargas, mas é relevante falar sobre esse problema para que sejam observados os efeitos de borda gerados no campo elétrico.

Para esse problema, tomam-se as bordas da malha aterradas $V_{bordas} = 0V$. Além disso, ao longo da malha são depositadas 2 placas de potencial constante. A placa 1, em $x = 0.2m$ tem potencial $V_1 = 1V$, enquanto a placa 2, em $x = 0.8m$ tem potencial $V_2 = -1V$. Ambas as placas tem comprimento $h = 0.8m$. É esperado, portanto, que o campo elétrico entre as placas seja aproximadamente constante.

Na figura 5 está representada a evolução dos δ para um número grande de repetições. Ali, é claro o início da convergência na iteração $n = 40$. Apesar disso, ao observar a evolução do perfil de V iteração por iteração, o sistema

parece se tornar aproximadamente estático para $n \rightarrow 300$. Assim, o erro escolhido foi $\epsilon_{opt} = 7.2 \cdot 10^{-10}V$.

O resultado da simulação é mostrado na figura 6, que ilustra não apenas o potencial, mas também o campo elétrico. A amplitude do campo elétrico é ilustrada pelo tamanho das setas de campo. Nota-se que é possível ver os efeitos de borda do campo elétrico criado pelos capacitores! Apesar disso, no interior dos capacitores (na região mais afastada das bordas), as linhas de campo apresentam comprimentos muito semelhantes, o que indica um campo elétrico constante, como esperado.

Uma análise interessante pode ser feita para verificar a validade da simulação do campo de capacitor. Escolhendo $y = 0.5$, é possível traçar um perfil da amplitude do campo elétrico $|\vec{E}|$ em função de x . A figura 7 mostra que a amplitude do campo elétrico das placas paralelas, em $x = 0.5m$ é aproximadamente constante em 3 regiões. Isso é coerente com o esperado pois além do capacitor principal, as placas centrais formam sistemas capacitivos com as bordas da malha.

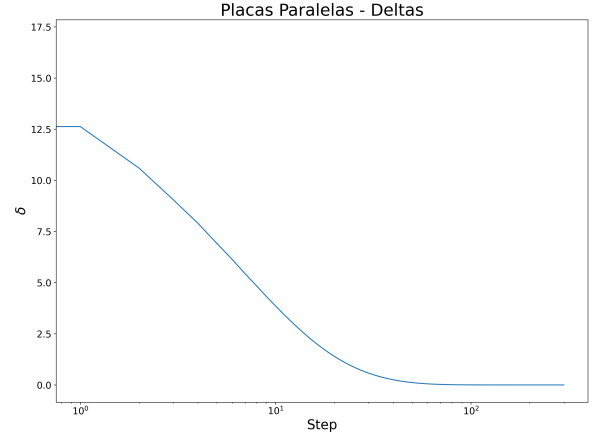


Figura 5: Evolução dos δ com o aumento de iterações no caso das placas paralelas. Os parâmetros iniciais utilizados foram $L_x = L_y = 1m$, $dx = dy = 0.05m$. As placas paralelas têm comprimento $h = 0.8m$ e foram colocadas sobre as posições $x_1 = 0.2m$ ($1V$) e $x_2 = 0.8m$. O chute inicial utilizado foi o usual $V_0 = 0V$.

Além disso, é interessante pensar sobre a validade da aproximação de um campo de capacitor como um campo elétrico uniforme. Afinal, na figura 7, é possível ver que a amplitude do campo entre as placas para um y fixo não é perfeitamente constante. Para sermos coerentes, é necessário levar em conta que, nas aproximações usuais, garante-se que as dimensões das placas são muito superiores à separação entre elas. Portanto, é necessário realizar a simulação para placas mais próximas. Na figura 8, é fácil ver que a região de campo constante é muito mais extensa

(em y) e a aproximação é mais coerente dadas as dimensões do problema. Isso é confirmado pela análise do módulo do campo elétrico para um y fixo na figura 9

Até agora, apesar de terem sido estudados sistemas muito interessantes e úteis, o método computacional não está sendo completamente usado. Afinal, ainda não foram inseridas cargas! Isso significa que todo o trabalho até aqui poderia ter sido realizado apenas com a solução computacional da equação de laplace. É hora de inserir cargas elétricas nos sistemas para que seja estudado o comportamento de seu campo elétrico.

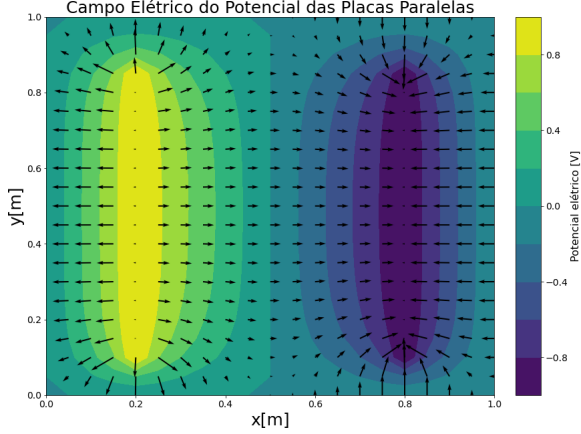


Figura 6: Potencial e campo elétricos simulados no caso das placas paralelas. Os parâmetros iniciais utilizados foram $L_x = L_y = 1m$, $dx = dy = 0.05m$. As placas paralelas têm comprimento $h = 0.8m$ e foram colocadas sobre as posições $x_1 = 0.2m$ ($1V$) e $x_2 = 0.8m$. O chute inicial utilizado foi o usual $V_0 = 0V$. A amplitude do campo elétrico é representada pelo comprimento das setas de campo em preto.

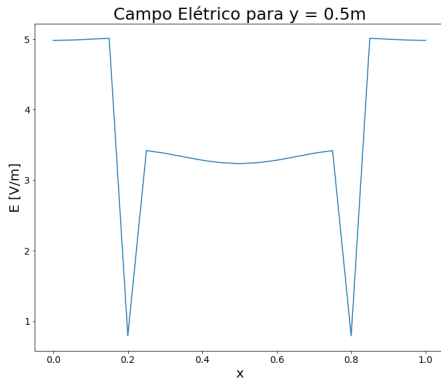


Figura 7: Amplitude do campo elétrico em $y=0.5m$ em função de x .

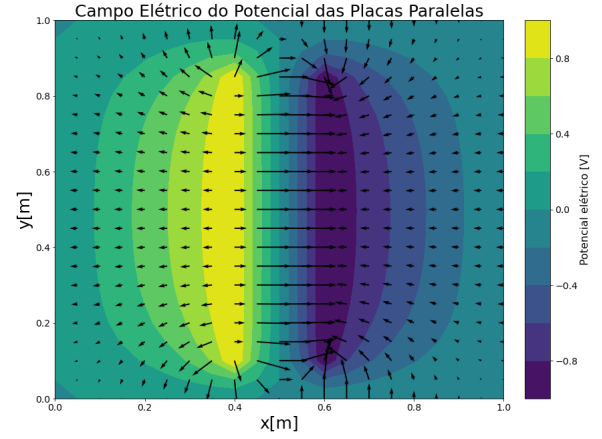


Figura 8: Campo elétrico gerado por um capacitor de placas paralelas de altura $h = 0.8m$. As placas foram posicionadas em $x_1 = 0.4m$ e $x_2 = 0.6m$ e têm potenciais de $V_1 = 1V$ e $V_2 = -1V$.

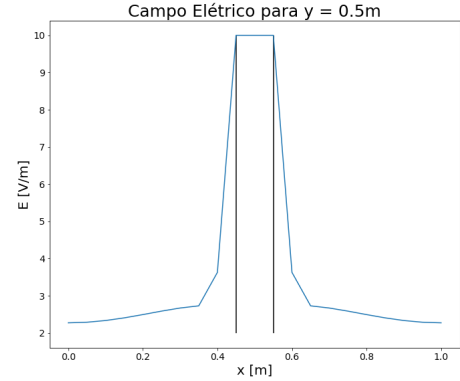


Figura 9: Amplitude do campo elétrico gerado por um capacitor de placas paralelas "ideal" na coordenada $y=0.5m$ como função de x . As placas foram posicionadas em $x_1 = 0.4m$ e $x_2 = 0.6m$ e têm potenciais de $V_1 = 1V$ e $V_2 = -1V$.

III. Carga Isolada

Podemos, agora, explorar um sistema diferente. Aqui, além das condições de contorno, há uma distribuição de cargas. Definimos, por conveniência, $\epsilon_0 = 1C^2N^{-1}m^{-2}$. No caso, é estudada apenas uma carga $q = 1C$ centralizada na malha. As paredes são mantidas aterradas $V_{Bordas} = 0V$. As dimensões da malha são mantidas em $L_x = L_y = 1m$ com passo $dx = dy = 0.05m$.

A figura 10 mostra a evolução dos δ para o caso da carga isolada. É notável que, em comparação aos casos sem distribuição de cargas, o sistema apresenta maior dificuldade de convergência. O valor de δ começa a apresentar um comportamento assintótico próximo à iteração 300.

O erro escolhido, analisando o comportamento da evolução do perfil do potencial com as iterações foi $\epsilon_{optimal} = 4.7 \cdot 10^{-7} V$.

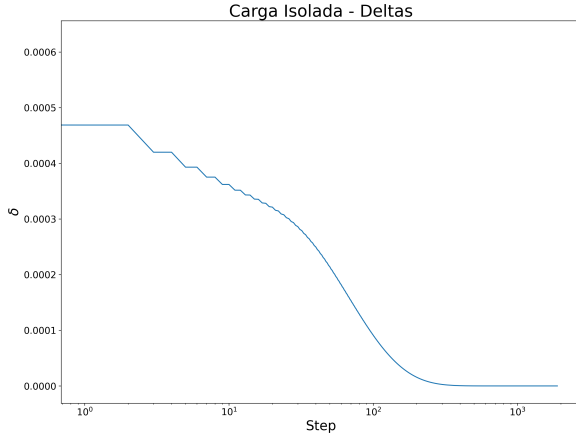


Figura 10: Evolução dos δ para o potencial elétrico de uma carga isolada $q = 1C$ posicionada em $x = 0.5m$ e $y = 0.5m$ numa malha de dimensões $L_x = L_y = 1m$ com passos $dx = dy = 0.05m$. O potencial nas bordas foi mantido nulo, enquanto o chute inicial foi o usual V_0 .

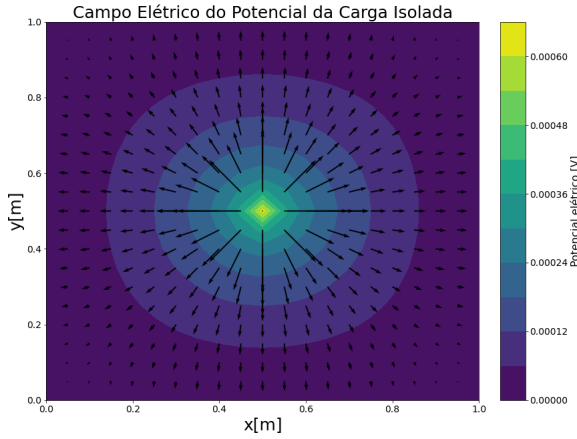


Figura 11: Campo e Potencial elétricos de uma carga isolada $q = 1C$ posicionada em $x = 0.5m$ e $y = 0.5m$ numa malha de dimensões $L_x = L_y = 1m$ com passos $dx = dy = 0.05m$. O potencial nas bordas foi mantido nulo, enquanto o chute inicial foi o usual V_0 .

Nota-se, na figura 11, que o campo é radial e sua amplitude diminui com a distância até o centro, como esperado.

Uma análise interessante pode ser feita aqui. É esperado que a amplitude do campo de uma carga isolada caia com inverso do quadrado da distância até a carga (assim como o potencial deve cair com o inverso da distância).

Uma figura com ajustes das dependências citadas é encontrado na figura 12.

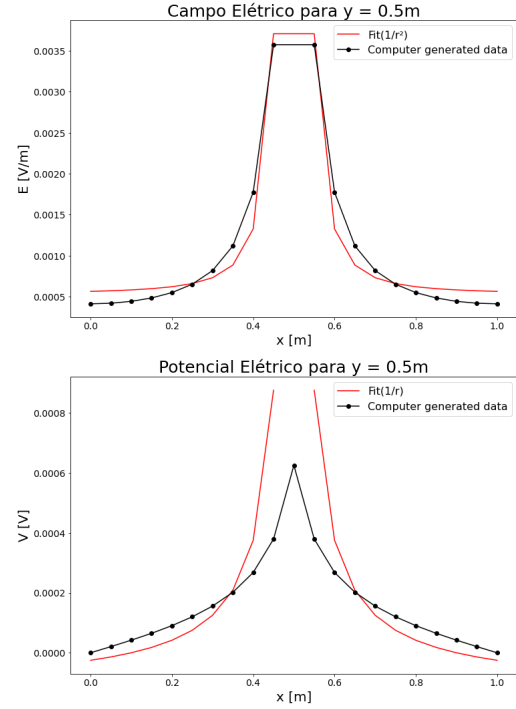


Figura 12: Dependência da amplitude do campo e do potencial elétricos na distância até a carga (localizada em $x = 0.5m$).

É notável que o comportamento diverge rapidamente. E isso faz sentido! Afinal, a dependência da amplitude do potencial e do campo só é em $\frac{1}{r^2}$ e em $\frac{1}{r}$ respectivamente quando o potencial é nulo no infinito! Não numa borda a uma distância finita (como é o caso) [1]. Para realmente estudar tal dependência, é necessário deixar o potencial "livre" nas bordas da malha (equivalente a mandar a condição de contorno para o infinito). Isso pode ser feito utilizando diferentes tipos de condição de contorno, como condições de contorno de Neumann, que designam o valor da derivada do potencial nas bordas, não do potencial em si [4].

IV. Dipolo

Um caso particularmente interessante que pode ser estudado é o dipolo elétrico! Esse sistema é estudado em duas situações.

IV.1 Cargas afastadas

Esse é o caso em que é possível ver as linhas de campo bastante características de dipolo (ligando ambas as cargas). Após o limite de estabilização de 300 iterações, o resultado da simulação obtido é disposto na figura 14.

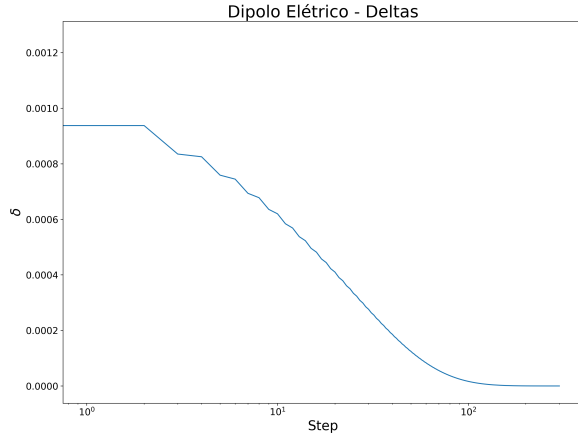


Figura 13: Evolução dos δ para o potencial elétrico de um dipolo elétrico com cargas posicionadas em $(0.5m, 0.3m)$, $q_1 = 1C$ e $(0.5m, 0.7m)$, $q_2 = -1C$ numa malha de dimensões $L_x = L_y = 1m$ com passos $dx = dy = 0.05m$. O potencial nas bordas foi mantido nulo, enquanto o chute inicial foi o usual V_0 .

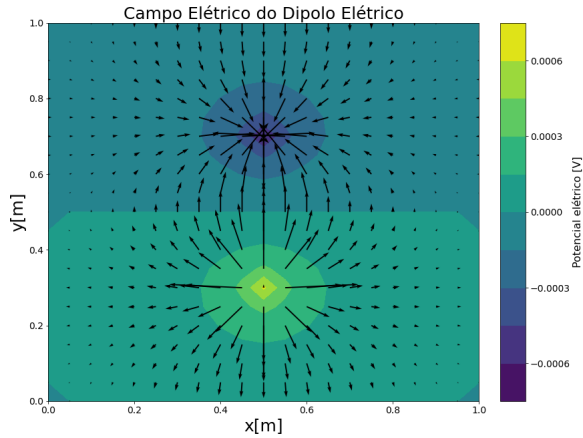


Figura 14: Campo e Potencial elétricos de um dipolo elétrico com cargas em $(0.5m, 0.3m)$, $q_1 = 1C$ e $(0.5m, 0.7m)$, $q_2 = -1C$ numa malha de dimensões $L_x = L_y = 1m$ com passos $dx = dy = 0.05m$. O potencial nas bordas foi mantido nulo, enquanto o chute inicial foi o usual V_0 .

IV.2 Cargas próximas

Aqui, o estudo é focado em observar o comportamento do campo longe do dipolo elétrico. Na figura 15 é possível ver as linhas de campo do dipolo de cargas próximas. As linhas de campo foram mudadas pois a amplitude do campo é muito pequena longe das cargas e, então, o sistema de setas utilizado anteriormente não seria visível.

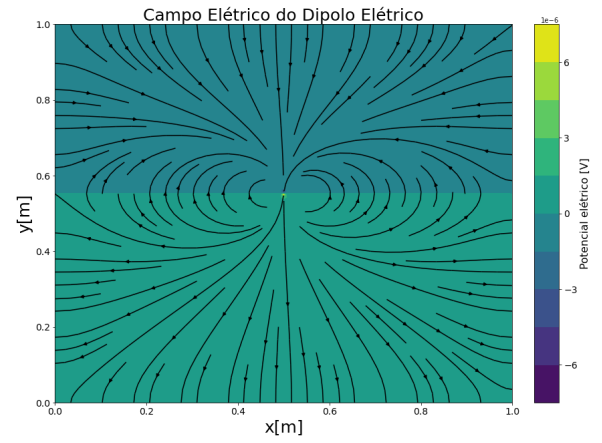


Figura 15: Campo e Potencial elétricos de um dipolo elétrico com cargas em $(0.5m, 0.55m)$, $q_1 = 1C$ e $(0.5m, 0.555m)$, $q_2 = -1C$ numa malha de dimensões $L_x = L_y = 1m$ com passos $dx = dy = 0.005m$. O potencial nas bordas foi mantido nulo, enquanto o chute inicial foi o usual V_0 .

V. Carga Isolada Entre um Capacitor de Placas Paralelas

É interessante ver o efeito criado pela presença de uma carga sobre o campo elétrico aproximadamente constante. Na figura 16 é possível ver que o campo envolve a carga e, então, perde suas propriedades de uniformidade.

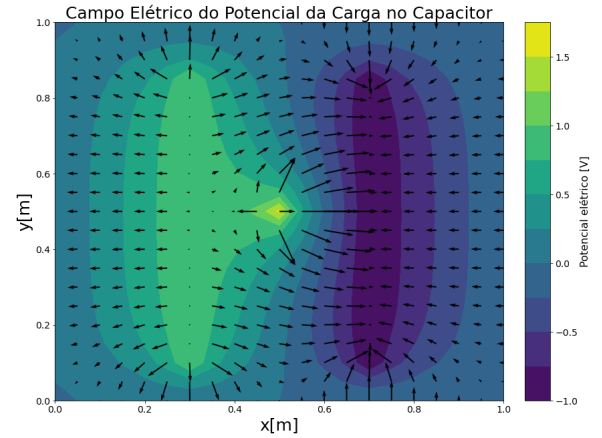


Figura 16: Campo e Potencial elétricos de uma carga inserida em um capacitor de placas paralelas. A carga é posicionada em $(0.5m, 0.5m)$ e tem carga $q = 2500C$ numa malha de dimensões $L_x = L_y = 1m$ com passos $dx = dy = 0.05m$. O potencial nas bordas foi mantido nulo, enquanto o chute inicial foi o usual V_0 .

Uma pergunta interessante que a simulação levanta é: Quão grandes as placas devem ser para que o campo em

seu interior volte a ser constante, ou seja, para que a carga seja "irrelevante"? . Isso pode ser analisado em um trabalho futuro.

Mas é importante deixar claro que, para que a carga fosse relevante nessa simulação, foi necessário fazer com que a carga pontual tivesse valor muito alto. Isso acontece pois a intensidade do campo elétrico dos capacitores é muito grande, principalmente devido à proximidade.

IV. CONCLUSÃO

Um método numérico foi descrito para a resolução e o estudo da equação de Poisson para sistemas eletrostáticos. O método utilizado e testado foi o método de relaxação de Jacobi, que mostrou eficiência ao analisar a velocidade com a qual os sistemas convergem às soluções, mas o que é realmente interessante é a amplitude dos casos que esse método computacional razoavelmente simples permite que sejam explorados. Há muitas outras distribuições de cargas interessantes que podem ser estudadas utilizando os métodos aqui descritos.

Foi evidenciado que o comportamento dos potenciais e dos campos de uma carga isolada não vai como o esperado pela literatura $E \propto \frac{1}{r^2}$ e $V \propto \frac{1}{r}$ para uma carga livre. Isso ocorre pois a carga está efetivamente em uma caixa de potencial constante. Para obter o resultado desejado, é necessário fazer as bordas estarem no infinito, o que deve ser atingido com outros tipos de condição de contorno, como as de Neumann. Mais estudos precisam ser realizados para melhorar a rapidez e a abrangência do método computacional, mas ele já é bastante eficaz para o estudo de sistemas simples e principalmente para o uso didático.

REFERÊNCIAS

- [1] David J Griffiths. Introduction to electrodynamics, 2005.
- [2] Robert Johansson. Introduction to scientific computing in python, 2016.
- [3] Luís Abramo. Notas de aula eletromagnetismo 1, 2021. Acessado em 30 de Maio de 2022.
- [4] Gabriel Landi. Notas de aula - física matemática 1, 2020. Acessado em 30 de Maio de 2022.

A. MANUAL DO USUÁRIO

O Código disponível no endereço a seguir **Pasta no Google Drive** é capaz de solucionar computacionalmente a equação de Poisson para uma gama de diferentes condições de contorno e distribuições de carga. Abaixo, há uma breve descrição das funções criadas nesse código, além de uma dica para a simulação de um sistema físico e criação de gráficos/animações.

GuilhermeDamacenaDeCastro_Projeto2.py

1. create_grid(Lx, Ly, dx, dy)

Essa função cria o ambiente onde o potencial e o campo serão calculados. O ambiente sempre é um retângulo de lados Lx e Ly divididos em passos de dx e dy.

Essa função retorna uma tupla com 4 itens: (x,y,xsize,ysize), onde x e y são listas das posições x e y existentes da malha espacial 2D, xsize e ysize são os tamanhos das listas x e y respectivamente.

2. initiate_matrices(xsize, ysize)

Essa função efetivamente cria as matrizes que serão utilizadas na simulação. Ela só precisa saber as dimensões da malha (xsize,ysize).

A função retorna uma tupla com 5 itens: (V, V_old, Ex, Ey, rho)

V é a matriz do potencial;

V_old é a matriz onde é armazenado o potencial anterior (útil para cálculo de erros);

Ex é a matriz onde são armazenados os valores de campo elétrico na direção x. O mesmo vale para Ey;

rho é a matriz onde é montada a distribuição de cargas.

3. create_rho(rho, x_position, y_position, charge)

Essa função adiciona as cargas desejadas à matriz rho.

Caso não deseje inserir cargas, basta não inserir os parâmetros _position e charge (são opcionais).

Caso deseje inserir cargas, adicione listas nas posições x_position, y_position e charge contendo as coordenadas x de cada carga, as coordenadas y de cada carga e a carga efetiva de cada carga (com sinal).

Essa função não retorna.

4. E(V, Ex, Ey)

É a função que calcula as componentes do campo elétrico. Ela precisa receber 3 matrizes. Uma com o potencial calculado e duas vazias para a inserção dos valores de Ex e Ey.

Essa função não retorna.

5. delta(V, V_old)

Essa função recebe duas matrizes de potencial e calcula o parâmetro δ entre elas.

Essa função retorna δ calculado.

6. add_border_conditions(V, xsize, ysize, value)

Essa função adiciona as condições de contorno nas bordas da malha. Efetivamente, ela precisa receber a matriz V, os valores xsize e ysize. O valor do potencial nas bordas é imposto como 0 caso não seja inserido no uso da função (value é um parâmetro opcional).

Essa função não retorna.

7. add_capacitor(V, xs, ys_d, ys_u, cs)

Essa função adiciona capacitores paralelos à malha. Ela precisa receber a matriz de potencial V, uma lista xs com as posições x onde cada placa paralela será inserida, uma lista ys_d com as coordenadas y inferiores de cada placa,

uma lista `ys_u` com as coordenadas `y` superiores de cada placa e uma lista `cs` com os valores dos potenciais constantes de cada placa.

Importante: Todas as listas devem ter tamanho 2. Não é possível inserir um número de placas paralelas diferente de 2.

essa função não retorna.

8. `iterate(V, V_old, eps, nmax, rho, xsize, ysize, capacitor, xs, ys_d, ys_u, cs)`

Essa função efetivamente itera a matriz `V` até uma repetição máxima `nmax` ou até que o delta calculado entre `V` e `V_old` (que é inserida na função apenas uma vez. Depois ela é calculada automaticamente) seja inferior ao valor `eps` escolhido. A função ainda deve receber a matriz `rho` e os valores `xsize` e `ysize`. Os parâmetros a partir de **capacitor** são opcionais. `capacitor` é uma variável booleana que recebe "False" ou "True" para que o programa saiba se há ou não um capacitor de placas paralelas no problema. Se houver, as coordenadas das placas são `xs`, `ys_u`, `ys_d` e `cs` como na função `add_capacitor`.

essa função retorna uma tupla de 3 itens: (`Vs`, `k`, `ds`)

`Vs` é um objeto que armazena todas as versões iteradas de `V`. Isso significa que o chute inicial de `V` pode ser encontrado fazendo `Vs[0]`.

`k` é o índice da última iteração realizada.

`ds` é a lista de todos os delta calculados (iteração a iteração).

9. `create_delta_plot(k, ds, title)`

Essa função cria um plot simples da evolução dos delta com a iteração. Ela recebe a lista dos delta, a iteração máxima e um título para o gráfico.

A função não retorna, mas salva uma figura em png. O nome da figura repete o título.

10. `create_basic_potential_plot(V, x, y, title)`

Essa função cria um gráfico de contorno do potencial dado pela matriz `V`. Ela recebe, além de `V`, as listas `x` e `y` e um título.

A função não retorna, mas salva uma figura em png. O nome da figura repete o título.

11. `create_field_plot(V, Ex, Ey, x, y, title)`

Essa função cria um gráfico de contorno do potencial subposto às setas do campo elétrico. Ela recebe as matrizes `V`, `Ex`, `Ey`, as listas `x` e `y`, além de um título para o gráfico.

Há uma peculiaridade sobre essa função. Em sua sintaxe há uma linha comentada

```
"# SP = ax2.streamplot(...)"
```

Você pode descomentar essa linha e comentar a linha seguinte

```
"ax2.quiver(...)"
```

Se quiser mudar o estilo com o qual os campos são apresentados. O primeiro mostra linhas de campo, melhor para uma visualização fácil do comportamento do campo vetorial. O segundo, setas de campo, melhor para observar a magnitude do campo.

Essa função não retorna, mas salva uma figura em png. O nome da figura repete o título.

12. `create_animation(vs, x, y, title)`

Essa função cria uma animação (filme) do relaxamento feito pelo método de jacobi. Isso significa que você é capaz de ver a evolução da matriz `V` iteração por iteração.

Ela recebe o objeto `Vs` (advindo da função `iterate`), as listas `x` e `y`, além de um título.

A função não retorna, mas salva um filme no formato .gif com o nome igual ao título.

13. debug de exemplo. Ao final do código, há um exemplo de como uma simulação deve ser construída usando as funções descritas aqui. Uma cópia desse exemplo é mostrada a seguir.

```
# Example: Parallel Plates Capacitor + Single Charge

# Choose the grid dimensions
Lx = 1
Ly = Lx

#Choose the grid steps
dx = 0.05
dy = dx

# Choose your stop parameters
eps = 7.4e-16
Nmax = 600

# Create grid
x, y, xsize, ysize = create_grid(Lx, Ly, dx, dy)

# Initiate matrices
V, V_old, Ex, Ey, rho = initiate_matrices(xsize, ysize)

# Add charge distribution
create_rho(rho, [0.5], [0.5], [2500])

# add border conditions
add_border_conditions(V, xsize, ysize, 0)

# Iterate and save vs, k and ds
vs, k, ds = iterate(V, V_old, eps, Nmax, rho, xsize,
                    ysize, capacitor = True, xs = [0.3, 0.7])

# calculate the resulting electric field
E(V, Ex, Ey)

# plot everything
create_delta_plot(k, ds, 'deltas-evolution')
create_basic_potential_plot(V, x, y, 'potential')
create_field_plot(V, Ex, Ey, x, y, 'electric-field')

# The animation is commented because it could take a while.
# It is worth it, though
# create_animation(vs, x, y, 'animation')
```