

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

Modelación Computacional de Sistemas Eléctricos(Gpo 1)

Profesora:

Eloína Rodríguez González

M4T4

Integrantes:

Armando Gutiérrez Rojo - A01702748

Oscar Eduardo Nieto Espitia - A01705090

Cristian Rogelio Espinosa Díaz - A01702752

Fermín Méndez García- A01703366

Pablo César Jiménez Villeda - A01703517

Conclusiones individuales

El estudiante redactará la funcionalidad, metodología de programación de la práctica, experiencia de aprendizaje y el avance en el desarrollo de la simulación de la situación problema

Armando: Se realizó una animación utilizando funciones y ciclos for. Esta animación muestra el flujo de glóbulos infectados y no infectados, donde los infectados se representan con color negro y se desplazan hacia el lado positivo y los no infectados aparecen con color blanco y se mueven hacia la placa de cargas negativas. Para el campo eléctrico se eligió un color morado para los vectores, un color verde para las líneas que representan el flujo de sangre, negro para la carga negativa y rojo para la carga positiva. En resumen, se realizó una simulación en la que podemos ver el comportamiento de los glóbulos para así poder hacer un diagnóstico.

Oscar: Como podemos observar en la simulación, los glóbulos infectados están representados por partículas negras y los glóbulos blancos no infectados por partículas blancas, graficamos las placas electroforéticas y mediante vectores representamos el campo electromagnético, ya que los glóbulos blancos infectados tienen una carga positiva estos se ven atraídos por la placa negativa, y así es como logramos separar glóbulos infectados y no infectados para poder hacer un diagnóstico.

Cristian: En esta entrega tuvimos que utilizar los conocimientos que hemos recolectado en estas sesiones para poder desarrollar el producto completo, el cual es la simulación del comportamiento que tendrían los glóbulos durante el proceso de una dielectroforesis. Dentro de esta simulación se pudo observar que los glóbulos sanos se mueven hacia una placa de carga negativa mientras que los glóbulos infectados se mueven hacia una placa de carga positiva

César: En conclusión, aprender los componentes que conforman las funciones para graficar en matlab, así como el uso de palabras reservadas como hold on, nos permiten representar la información compleja de una forma entendible para el usuario. Al implementar dichos elementos de diseño los gráficos se hacen más legibles, por tanto, en el reto usaremos estos conocimientos para experimentar con las gráficas de formas distintas para presentar la información de una forma más legible.

Fermín: Por último, pulimos y completamos la programación añadiendo a lo que teníamos la animación de los glóbulos y simulando el comportamiento de los glóbulos, cómo es que se separan los sanos de los infectados. Confirmando cómo es que las cargas distintas se separan en un campo eléctrico de electrodos de placas paralelas con cargas opuestas.

Reto. Cálculo y graficación de campos eléctricos no uniformes, como los usados en dielectroforesis para el diagnóstico de malaria

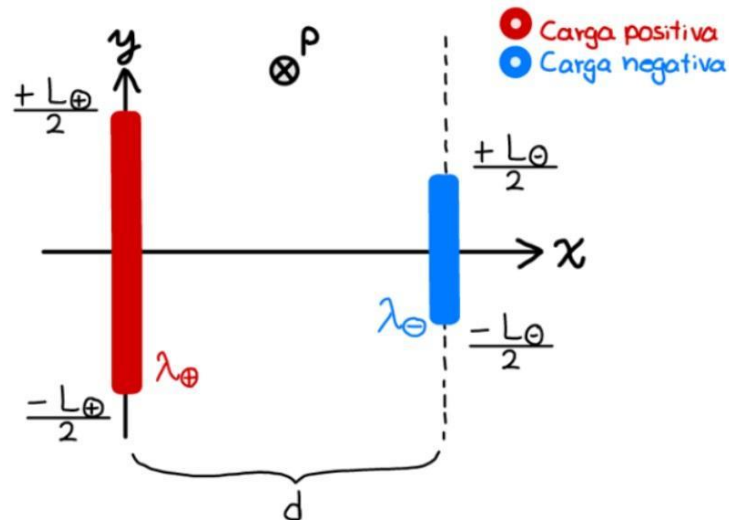
Objetivo

Desarrollar una **simulación computacional 2D del campo eléctrico**, en MATLAB, producido por dos electrodos de placas planas, paralelas, de carga opuesta y el usuario podrá modificar el tamaño de una de las placas. En un recuadro de la simulación, el usuario podrá especificar las

coordenadas (x,y,z) de un punto entre las placas, y se debe mostrar en ese mismo recuadro los valores de las componentes (Ex,Ey,Ez) del campo eléctrico en ese punto.

Entregable ③

Distribución de cargas lineales (alambres).



$$|Q_{\oplus \text{ total}}| = |Q_{\ominus \text{ total}}|$$

λ : Densidad de carga
"Uniforme"

$$\lambda_{\oplus} = \frac{Q_{\oplus}}{L_{\oplus}}$$

$$\lambda_{\ominus} = \frac{Q_{\ominus}}{L_{\ominus}}$$

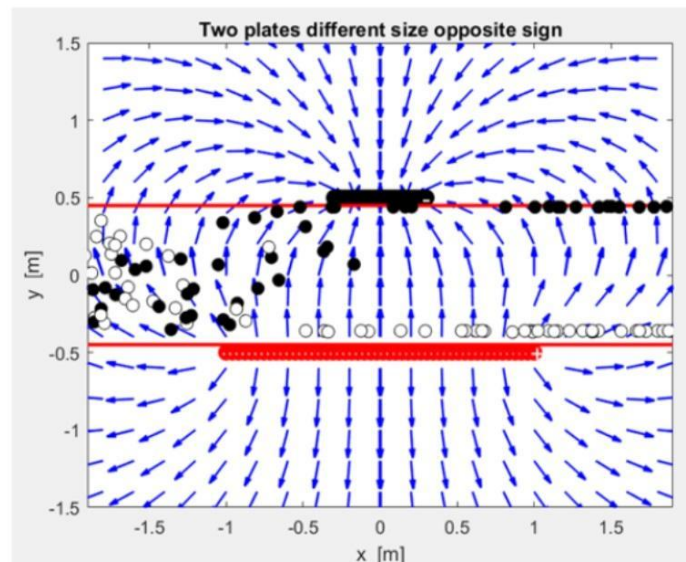


Figura 1. Distribución de cargas lineales (alambres).

Simulación Distribución de cargas lineales

- Limpiar las variables y consola

```
clear;
```

- Longitud de la pista, $L = 20$

$L = 20;$

- Números de puntos a cuadrricular, $N = 1001$

$N = 1001;$

- Asignar el 100 cargas, N_Q

$N_Q = 100;$

- Carga, $Q_0 = 10 \times 10^{-6}$

$Q_0 = 10e-6;$

- Radio de la carga, $r = 0.05$

$r = 0.05;$

$$x_1 = -1; \quad x_2 = 1;$$

$$x_3 = -0.3; \quad x_4 = 0.3;$$

$$y_1 = -0.3; \quad y_2 = 0.5;$$

- Componentes x,y de la posición horizontal de carga,

$x_1 = -1; \quad x_2 = 1;$

$x_3 = -0.3; \quad x_4 = 0.3;$

$y_1 = -0.5; \quad y_2 = 0.5;$

- Asignar la constante dieléctrica, $\epsilon_0 = 8.854 \times 10^{-12};$

$\epsilon_0 = 8.854e-12;$

- Asignar la constante de Coulomb, $k_C = \frac{1}{4 * \pi * \epsilon_0};$

$k_C = 1/(4 * \pi * \epsilon_0);$

Dimensiones de región:

$$\min X = -2; \quad \max X = 2;$$

$$\min Y = -2; \quad \max Y = 2;$$

$$\min R = 1 \times 10^{-6};$$

$$\min R_x = 1 \times 10^{-6};$$

- $\min R_y = 1 \times 10^{-6};$

```

minX = -2;
maxX = 2;
minY = -2;
maxY = 2;
minR = 1e-6;
minRx = 1e-6;
minRy = 1e-6;

```

$$V_{\text{sat}} = 1 \times 10^7;$$

- Saturación de niveles, $E_{\text{sat}} = 1 \times 10^8;$

```

Vsat = 10000000;
Esat = 100000000;

```

- Inicialización de Carga, $Q = -Q_0 * [1, 1, 1, \dots]_{1 \times N_Q}$

```

Q = -Q0 .* ones(1,NQ);

```

- Asignar el vector de cargas iniciales $Q\left(1 + \frac{N_Q}{2} : N_Q\right) = Q_0$

```

Q(1+NQ/2:NQ) = Q0;

```

Componentes x,y de la posición de carga,

- longitud límite inferior, $x_C\left(1 : \frac{N_Q}{2}\right) \in [x_1, x_2]$ en intervalos de $\frac{N_Q}{2}$

```

xC(1:NQ/2) = linspace(x1,x2,NQ/2);

```

- **longitud límite superior** $x_C\left(1 + \frac{N_Q}{2} : N_Q\right) \in [x_3, x_4]$ en intervalos de $\frac{N_Q}{2}$

```

xC(1+NQ/2:NQ) = linspace(x3,x4,NQ/2);

```

- altura límite inferior, $y_C\left(1 : \frac{N_Q}{2}\right) = y_1$

```

yC(1:NQ/2) = y1;

```

- altura límite superior, $y_C\left(1 + \frac{N_Q}{2} : N_Q\right) = y_2;$

```
yC(1+NQ/2:NQ) = y2;
```

$$V = 0_{N \times N};$$
$$E_x = 0_{N \times N};$$
$$E_y = 0_{N \times N};$$

- Inicialización de Voltaje y Campos en componentes x,y

```
V = zeros(N,N);
```

```
Ex = zeros(N,N);
```

```
Ey = zeros(N,N);
```

$$x \in [\text{minX}, \text{maxX}] \text{ en } N \text{ elementos}$$

- Región en 2D, $y \in [\text{minY}, \text{maxY}]$ en N elementos

```
x = linspace(minX,maxX,N);
```

```
y = linspace(minY, maxY,N);
```

Color de la carga

- Asignar a colorR , color rojo en vector RGB
- Asignar a colorN , color negro en vector RGB

```
colorR = [1 0 0];
```

```
colorN = [0 0 0];
```

- Posiciones de la cuadrícula para dibujar quiver, $[xG, yG] = \text{meshgrid}(x, y)$

```
[xG, yG] = meshgrid(x,y);
```

Calcular el campo potencial y eléctrico por un ciclo

- Para $n = 1 : N_Q$

```
for n = 1 : NQ
```

$$R_x = xG - xC(n);$$

- Componentes x,y para desplazar la carga al origen, $R_y = yG - yC(n);$

```
Rx = xG - xC(n);
```

```
Ry = yG - yC(n);
```

- Asignar el elemento del vector de R_x , $\text{elem} = \text{find}(|R_x| + |R_y| == 0);$

```
elem = find(abs(Rx)+ abs(Ry) == 0);
```

$$R_x(\text{elem}) = \min R_x;$$

- Asignar el menor de los R_x, R_y , $R_y(\text{elem}) = \min R_y;$

$$R_x(\text{elem}) = \min R_x;$$

$$R_y(\text{elem}) = \min R_y;$$

- Calcular la magnitud del vector, $R = \sqrt{R_x.^2 + R_y.^2};$

$$R = \text{sqrt}(R_x.^2 + R_y.^2);$$

- Colocar el origen del radio, $R(R == 0) = \min R;$

$$R(R == 0) = \min R;$$

$$V = V + kC \cdot \frac{Q(n)}{R}$$

- Calcular el Voltaje

$$V = V + kC \cdot Q(n) ./ R;$$

$$R3 = R.^3;$$

$$E_x = E_x + \frac{kC \cdot Q(n) \cdot R_x}{R3};$$

$$E_y = E_y + \frac{kC \cdot Q(n) \cdot R_y}{R3};$$

- Calcular el Campo eléctrico,

$$R3 = R.^3;$$

$$E_x = E_x + kC \cdot Q(n) \cdot R_x ./ R3;$$

$$E_y = E_y + kC \cdot Q(n) \cdot R_y ./ R3;$$

- Fin de ciclo n

end

$$\max(\max(V)) \geq V_{\text{sat}};$$

- Condicional para el vector de valores máximos de V , $V(V > V_{\text{sat}}) = V_{\text{sat}};$

if max(max(V)) >= Vsat; V(V > Vsat) = Vsat; end

$$\min(\min(V)) \leq -V_{\text{sat}};$$

- Condicional para el vector de valores mínimos de V , $V(V < -V_{\text{sat}}) = -V_{\text{sat}};$

if min(min(V)) <= -Vsat; V(V < -Vsat) = -Vsat; end

- Calculo de magnitud del campo eléctrico, $E = \sqrt{E_x.^2 + E_y.^2};$

```
E = sqrt(Ex.^2 + Ey.^2);
```

- Condicional para el vector de valores máximos de E ,
 $\max(\max(E)) \geq \text{Esat}; E(E > \text{Esat}) = \text{Esat};$

```
if max(max(E)) >= Esat; E(E > Esat) = Esat; end
```

- Condicional para el vector de valores mínimos de E ,
 $\min(\min(E)) \leq -\text{Esat}; E(E < -\text{Esat}) = -\text{Esat};$

```
if min(min(E)) <= -Esat; E(E < -Esat) = -Esat; end
```

- Condicional para el vector de valores máximos de E_x ,
 $\max(\max(\text{Ex})) \geq \text{Esat}; \text{Ex}(\text{Ex} > \text{Esat}) = \text{Esat};$

```
if max(max(Ex)) >= Esat; Ex(Ex > Esat) = Esat; end
```

- Condicional para el vector de valores mínimos de E_x ,
 $\min(\min(\text{Ex})) \leq -\text{Esat}; \text{Ex}(\text{Ex} < -\text{Esat}) = -\text{Esat};$

```
if min(min(Ex)) <= -Esat; Ex(Ex < -Esat) = -Esat; end
```

- Condicional para el vector de valores máximos de E_y ,
 $\max(\max(\text{Ey})) \geq \text{Esat}; \text{Ey}(\text{Ey} > \text{Esat}) = \text{Esat};$

```
if max(max(Ey)) >= Esat; Ey(Ey > Esat) = Esat; end
```

- Condicional para el vector de valores mínimos de E_y ,
 $\min(\min(\text{Ey})) \leq -\text{Esat}; \text{Ey}(\text{Ey} < -\text{Esat}) = -\text{Esat};$

```
if min(min(Ey)) <= -Esat; Ey(Ey < -Esat) = -Esat; end
```

Grafica para el comportamiento del múltiples cargas

```
clf; figure
```

- Conservar elementos gráficos

```
hold on
```

```
index1 = 51 : 50 : 951;
```

```
index1 = [index 1500 502];
```

Asignar el indice para x y para y, `index2 = index1;`


```
index1 = 51 : 50 : 951;
index1 = [index1 500 502];
index2 = index1;
```

Graficar las líneas de campo eléctrico de quiver (representación por flechas)

- Coordenadas de origen (p_{xo}, p_{yo}) , $pxo = xG(index1, index2);$
 $pyo = yG(index1, index2);$

```
pxo = xG(index1, index2);
pyo = yG(index1, index2);
```

- coordenadas de final (p_{xf}, p_{yf}) , $pxf = \frac{Ex(index1, index2).}{(E(index1, index2))};$
 $pyf = \frac{Ey(index1, index2).}{(E(index1, index2))};$

```
pxf = Ex(index1, index2)./(E(index1, index2));
pyf = Ey(index1, index2)./(E(index1, index2));
```

- Asignar los parámetros del objeto gráfico,
 $h = \text{quiver}(px, pyo, pxf, pyf, 'autoscalefactor', 0.8);$

```
h = quiver(pxo, pyo, pxf, pyf, 'autoscalefactor', 0.8);
```

- Definir las propiedades del gráfico $\text{set}(h, 'color', [0 \ 0 \ 1], 'linewidth', 1.2)$

```
set(h, 'color', [1 0 1], 'linewidth', 1.2)
```

- Conservar elementos graficos

```
hold on
```

Graficar conductores cargados negativos:

- **Para** $n = 1 : \frac{N_Q}{2} : N_Q$

```
for n = 1:NQ/2:NQ
```

- Vector de posiciones, $\text{pos} = [-r + xC(n), -r + yC(n), 2 * r, 2 * r];$

```
pos = [-r+xC(n), -r+yC(n), 2*r, 2*r];
```

- Asignar los parámetros del objeto gráfico, $h = \text{rectangle}('Position', \text{pos}, 'Curvature', [1, 1]);$

```
h = rectangle('Position', pos, 'Curvature', [1, 1]);
```

- Definir las propiedades del gráfico $\text{set}(h, 'FaceColor', \text{colorN}, 'EdgeColor', \text{colorN});$

```
set(h, 'FaceColor', colorN, 'EdgeColor', colorN);
```

- Texto para signo $-$, ubicación en $(\text{pos}(1) + 0.02, \text{pos}(2) + 0.05)$ **y tamaño de letra en 14 puntos**

```
text(pos(1)+0.02, pos(2)+0.05, '-', 'Color', 'white', 'FontSize', 14) % - sign
```

- Fin ciclo n

```
end
```

Graficar conductores cargados positivos:

- Para $n = 1 + \frac{N_Q}{2}$

```
for n = 1+NQ/2
```

- Vector de posiciones, $\text{pos} = [-r + xC(n), -r + yC(n), 2 * r, 2 * r];$

```
pos = [-r+xC(n), -r+yC(n), 2*r, 2*r];
```

- Asignar los parámetros del objeto gráfico, $h = \text{rectangle}('Position', \text{pos}, 'Curvature', [1, 1]);$

```
h = rectangle('Position', pos, 'Curvature', [1, 1]);
```

- Definir las propiedades del gráfico $\text{set}(h, 'FaceColor', \text{colorR}, 'EdgeColor', \text{colorR});$

```
set(h, 'FaceColor', colorR, 'EdgeColor', colorR);
```

- Texto para signo $+$, ubicación en $(\text{pos}(1) + 0.02, \text{pos}(2) + 0.05)$ y tamaño de letra en 14 puntos

```
text(pos(1)+0.02, pos(2)+0.05, '+', 'Color', 'white', 'FontSize', 14) % + sign
```

- Fin ciclo n

```
end
```

- Etiquetas de los ejes $x[m]$, $y[m]$

```
xlabel('x [m]');  
ylabel('y [m]');
```

- **Título** Two plates different size opposite sign

```
title('Simulación dielectroforesis: Placas paralelas de carga opuesta');
```

- Definir las propiedades del gráfico, $x \in [-2, 2]$ y $y \in [-2, 2]$

```
set(gca, 'xLim', [-2, 2]); % gca devuelve los ejes actuales  
set(gca, 'yLim', [-2, 2]);
```

- Límites de presentación de gráfico, $x \in [-1.5, 1.5]$ y $y \in [-1.5, 1.5]$ y eliminar el marco de quiver

```
axis([-1.5 1.5 -1.5 1.5]);  
axis equal  
box on
```

Dielectroforesis

- Asignar el número de celdas (espacios), $nCells = 500$

```
nCells = 500;
```

- Crear el objeto, $Cells = gobjects(nCells, 1)$

```
Cells = gobjects(nCells, 1);
```

- Asignar la posición de las celdas, $Positions = zeros(nCells, 5)$, con 5 parámetros

```
Positions = zeros(nCells, 5);
```

- Para $i \in [1, nCells]$

```
for i=1:nCells
```

Asignar ubicaciones de cada elemento de la globulo (infectada o no):

- Posición en componente X, $Positions(i, 1) = -2$

```
Positions(i, 1) = -2;
```

- Posición en componente Y, $Positions(i, 2) = -0.4 + 0.8 * rand(1)$

```
Positions(i,2) = -0.4+0.8*rand(1);
```

- Posición para evaluar infectado o no, $\text{Positions}(i,3) = \text{rand}(1)$

```
Positions(i,3) = rand(1);
```

- Posición de Velocidad en componente X, $\text{Positions}(i,4) = \text{rand}(1) * 0.1$

```
Positions(i,4) = rand(1)*0.1;
```

- Posición de Velocidad en componente Y, $\text{Positions}(i,5) = 0$

```
Positions(i,5) = 0;
```

- Fin de ciclo i

```
end
```

Animación

- Conservar gráficos

```
hold on
```

- Asignar $n = 1$ para animar una celda a la vez

```
n = 1;
```

- Dibujar el área (flujo de sangre), rectangulo

```
rectangle('Position',[-2 -0.4 4 0.9], 'EdgeColor','g', 'linewidth',2); %Sangre
```

- **Mientras** hay globulos por dibujar, $\min(\text{Positions}(:,1)) < 1.9$

```
while min(Positions(:,1)) < 1.9
```

- Ciclo **para** $i \in [1,n]$

```
for i=1:n
```

- Actualizar la posición X, $\text{Positions}(i,1) = \text{Positions}(i,1) + \text{Positions}(i,4)$

```
Positions(i,1) = Positions(i,1) + Positions(i,4);
```

- Actualizar la posición Y, $\text{Positions}(i,2) = \text{Positions}(i,2) + \text{Positions}(i,5)$

```
Positions(i,2) = Positions(i,2) + Positions(i,5);
```

- Si el globo es visible en la gráfica

```
if Positions(i,1) < 1.9
```

- Asignar la posición y tamaño de cada globo,
pos = [Positions(i,1) Positions(i,2) 0.08 0.08]

```
pos = [Positions(i,1) Positions(i,2) 0.08 0.08];
```

- Dibujar el globo, Cells(i) = rectangle('Position',pos,'Curvature',[1 1])

```
Cells(i) = rectangle('Position',pos,'Curvature',[1 1]);
```

- Si el globo es infectado, Positions(i,3) <= 0.5

```
if Positions(i,3) <= 0.5
```

- Asignar color negro, Cells(i).FaceColor = 'k'

```
Cells(i).FaceColor = 'k';
```

- Si Positions(i,1) >= -1

```
if Positions(i,1) >= -1
```

- Si Positions(i,2) <= 0.4

```
if Positions(i,2) <= 0.4
```

- entonces, Positions(i,5) = Positions(i,5) + 0.1 * Positions(i,4)

```
Positions(i,5) = Positions(i,5) + 0.1*Positions(i,4);
```

```
Positions(i,2) = 0.4;
```

- Si no Positions(i,5) = 0;

```
else
```

```
Positions(i,2) = 0.4;
```

```
Positions(i,5) = 0;
```

- Fin de Si Positions(i,2) <= 0.4

end

- Fin de **Si** $\text{Positions}(i, 1) \geq -1$

end

- **Si no** de el globulo es infectado

else

- Asignar color blanco para no infectado

$\text{Cells}(i).\text{FaceColor} = 'w';$

- **Si** $\text{Positions}(i, 1) \geq -1$

if $\text{Positions}(i, 1) \geq -1$

- **Si** $\text{Positions}(i, 2) \leq -0.4$

if $\text{Positions}(i, 2) \leq -0.4$

- entonces, $\text{Positions}(i, 5) = \text{Positions}(i, 5) - 0.1 * \text{Positions}(i, 4)$

$\text{Positions}(i, 5) = \text{Positions}(i, 5) + 0.1 * \text{Positions}(i, 4);$

$\text{Positions}(i, 2) = -0.4;$

- **Si no** $\text{Positions}(i, 5) = 0;$

else

$\text{Positions}(i, 2) = -0.4;$

$\text{Positions}(i, 5) = 0;$

- Fin de **Si** $\text{Positions}(i, 2) \leq 0.4$

end

- Fin de **Si** $\text{Positions}(i, 1) \geq -1$

end

- Fin de Si el globulo es infectado, $\text{Positions}(i, 3) \leq 0.5$

end

- Fin de Si el globulo es visible en la gráfica

end

- Fin del ciclo i

end

- Asignar el nuevo globulo a dibujar

$n = n + 1;$

- Si los globulos son dibujados, guardar $n \geq nCells$, $n = nCells$;

if $n \geq nCells$, $n = nCells$; end

drawnow

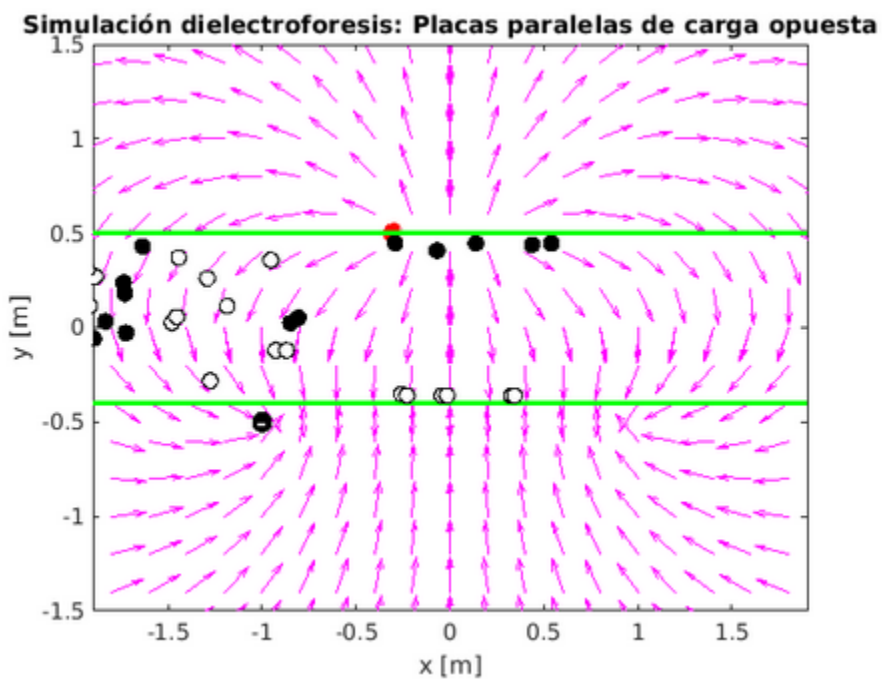
- Pausar para mostrar los globulos delete(Cells)

drawnow

delete(Cells)

- Fin del ciclo infinito Positions

end



Recursos

T. B. Jones, *Electromechanics of particles* (Cambridge University Press, Cambridge, 1995).

Dielectroforesis

Ozuna, Lapizco, (2007)

Principios de Circuitos Eléctricos

Floyd

Pearson, Prentice Hall

Desarrollar apps mediante App Designer — Ejemplos

Help, MathWorks

Drawnow

Help, MathWorks