

Tutorial de Electrodinámica Computacional

Ejemplo 2: Discretización del espacio en dos dimensiones

1. Objetivo del ejemplo

El objetivo de este ejemplo es construir de manera explícita una **mallla discreta de puntos** en el plano xy , la cual será utilizada posteriormente para evaluar el campo eléctrico generado por una carga puntual.

Este paso es fundamental en electrodinámica computacional, ya que permite pasar de un espacio continuo a un conjunto finito de puntos donde se realizan los cálculos numéricos.

2. Contexto físico y numérico

En la formulación continua, el campo eléctrico es una función vectorial definida en todo el espacio:

$$\mathbf{E}(\mathbf{r}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad (1)$$

En una simulación numérica, el espacio se discretiza y el campo se evalúa únicamente en un conjunto finito de puntos (x_i, y_j) . Este ejemplo construye explícitamente dichos puntos.

3. Implementación en Python

A continuación se analiza el código paso a paso.

3.1. Importación de librerías

```
import numpy as np
import matplotlib.pyplot as plt
```

Se utiliza `numpy` para el manejo de arreglos y `matplotlib` para la visualización de la malla de puntos.

3.2. Parámetros físicos

```
# Parámetros físicos
k = 1.0
q = 1.0
```

En este ejemplo, los valores de la constante de Coulomb k y de la carga q se fijan en unidades arbitrarias. Aunque no se utilizan aún, se incluyen para mantener continuidad con los siguientes ejemplos.

3.3. Posición de la carga

```
# Posición de la carga
x_q = 0.0
y_q = 0.0
```

La carga puntual se encuentra en el origen del sistema de coordenadas. Esta información será relevante cuando se calcule el campo eléctrico.

3.4. Definición del dominio espacial

```
# Dominio espacial
xmin = -2.0
xmax = 2.0
Nx = 10

ymin = -2.0
ymax = 2.0
Ny = 10
```

Se define una región cuadrada del plano xy :

$$x \in [x_{\text{mín}}, x_{\text{máx}}], \quad y \in [y_{\text{mín}}, y_{\text{máx}}] \quad (2)$$

La región se discretiza en $N_x \times N_y$ puntos.

3.5. Espaciamientos

```
dx = (xmax - xmin) / (Nx - 1)
dy = (ymax - ymin) / (Ny - 1)
```

Los parámetros dx y dy representan la distancia entre puntos consecutivos en cada dirección. Esto define la resolución espacial de la simulación.

4. Construcción de la malla

4.1. Prealocación de arreglos

```
X = np.zeros(Nx * Ny)
Y = np.zeros(Nx * Ny)
```

Se crean arreglos unidimensionales para almacenar las coordenadas de todos los puntos de la malla. Cada índice corresponde a un punto del espacio.

4.2. Llenado de la malla

```
n = 0
for j in range(Ny):
    y = ymin + j * dy
    for i in range(Nx):
        x = xmin + i * dx
```

```
X[n] = x
Y[n] = y

n = n + 1
```

Este doble ciclo recorre sistemáticamente el dominio espacial:

- El índice j recorre la dirección y .
- El índice i recorre la dirección x .

Cada par (x, y) representa un punto donde posteriormente se evaluará el campo eléctrico.

5. Visualización de la malla

```
plt.scatter(X, Y)
plt.gca().set_aspect('equal')

# Guardar figura
plt.savefig('a6_01.png', dpi=300, bbox_inches='tight')
plt.show()
```

Cada punto representa una posición del espacio donde se realizarán los cálculos. La proporción igual asegura que la geometría no se distorsione.

6. Resultado gráfico

La malla discreta generada se muestra en la Figura 1.

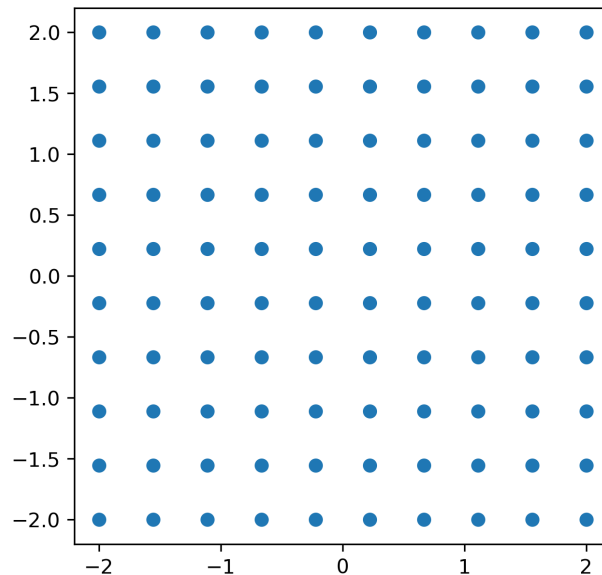


Figura 1: Malla discreta de puntos en el plano xy utilizada como dominio de cálculo.

7. Comentario final

Este ejemplo introduce el concepto fundamental de **discretización del espacio**, indispensable en cualquier simulación de electrodinámica computacional.

En el siguiente paso, cada punto de esta malla se utilizará para calcular el campo eléctrico generado por la carga puntual mediante la ley de Coulomb.