Electrostática Computacional
Generación eficiente de mallas con `linspace` y `meshgrid`

1. Motivación

En las secciones anteriores se construyó explícitamente una malla bidimensional mediante ciclos `for`. Aunque este procedimiento es pedagógicamente útil, resulta innecesariamente largo y poco eficiente desde el punto de vista computacional.

En esta sección se muestra cómo las funciones `linspace` y `meshgrid` permiten generar automáticamente una malla bidimensional, produciendo un código más compacto, más legible y más cercano a la formulación matemática del problema físico.

2. Discretización del dominio con `linspace`

El dominio espacial bidimensional se define como

$$x \in [-1, 1], \qquad y \in [-1, 1].$$

La instrucción `linspace` permite generar directamente los vectores de coordenadas físicas:

$$x_i = -1 + i\,\Delta x, \qquad y_j = -1 + j\,\Delta y, \tag{1}$$

con

$$N = 51$$

puntos uniformemente distribuidos en cada dirección.

3. Construcción automática de la malla

La función `meshgrid` toma los vectores unidimensionales x y y y construye dos matrices bidimensionales:

$$X_{i,j} = x_j, \qquad Y_{i,j} = y_i. \tag{2}$$

Estas matrices representan explícitamente la malla bidimensional del dominio físico, donde cada par $(X_{i,j}, Y_{i,j})$ corresponde a un punto espacial.

Este procedimiento reemplaza completamente la construcción manual mediante ciclos anidados.

4. Modelo físico

Se considera nuevamente el potencial eléctrico bidimensional de una carga puntual:

$$\phi(x, y) = \frac{q}{\sqrt{x^2 + y^2}}, \tag{3}$$

donde q es la carga eléctrica. Esta expresión se evalúa simultáneamente en todos los puntos de la malla.

5. Código completo en Python

El siguiente script implementa la discretización del dominio, la construcción de la malla y la visualización del potencial eléctrico.

5.1. Script

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

# N mero de puntos por eje
N = 51

# Dominio espacial
x = np.linspace(-1, 1, N)
y = np.linspace(-1, 1, N)

# Malla bidimensional
X, Y = np.meshgrid(x, y)

# Par metro f sico
q = 1.0

# Potencial el ctrico
R = np.sqrt(X**2 + Y**2)
phi = q / R

# Gr fica del potencial
plt.figure(figsize=(6,5))
plt.pcolormesh(X, Y, phi, shading='auto')
plt.colorbar(label='Potencial $\phi$')
plt.xlabel('x')
plt.ylabel('y')
plt.title(r'Potencial $\phi(x,y) = q / \sqrt{x^2 + y^2}$')
plt.axis('equal')
plt.show()
```

6. Interpretación de la gráfica

La Figura 1 muestra el potencial eléctrico evaluado sobre una malla bidimensional generada automáticamente.

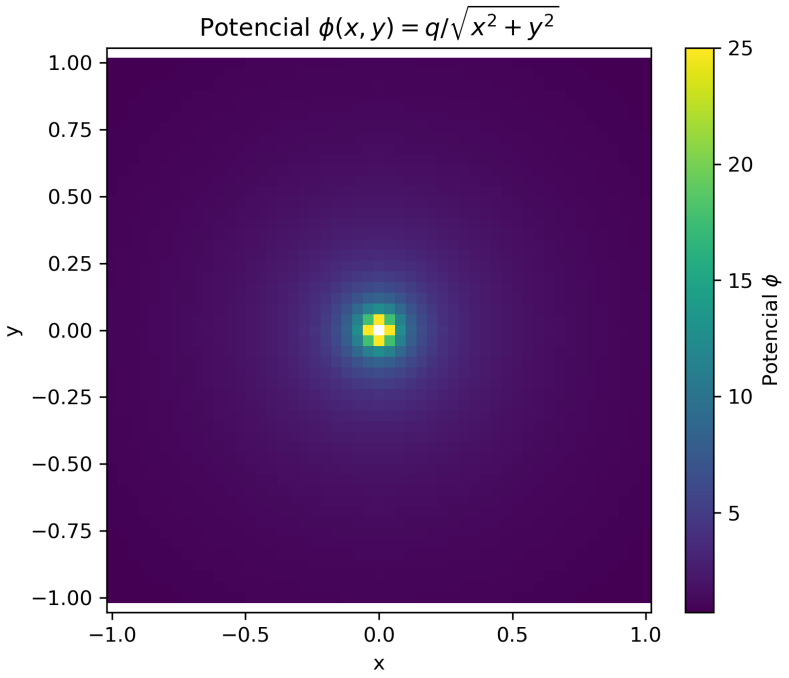

Figura 1: Potencial eléctrico bidimensional calculado usando `linspace` y `meshgrid`. Los ejes corresponden a coordenadas físicas.

A diferencia de la visualización directa de una matriz:

- los ejes están correctamente escalados en unidades de longitud,
- cada punto del color corresponde a una posición física (x, y),
- la simetría radial del potencial se aprecia claramente.

7. Ventajas del enfoque con `meshgrid`

El uso de `linspace` y `meshgrid` presenta ventajas claras:

- elimina ciclos explícitos,
- reduce significativamente la longitud del código,
- facilita la vectorización de operaciones,
- refleja directamente la estructura matemática del problema.

Este enfoque es el estándar en electrostática computacional y métodos numéricos en dos dimensiones.

8. Preparación para métodos numéricos

Una vez definida la malla bidimensional, el siguiente paso natural es utilizarla para:

- resolver la ecuación de Laplace,
- resolver la ecuación de Poisson con fuentes distribuidas,
- implementar esquemas de diferencias finitas en 2D.

La estructura (X, Y, ϕ) es precisamente la que se utiliza en estos métodos.

9. Conclusión

El uso de `linspace` y `meshgrid` permite pasar de una construcción manual de la malla a una formulación compacta, eficiente y conceptualmente clara, constituyendo la base de prácticamente todos los esquemas modernos de electrostática computacional en dos dimensiones.