

Resolución Numérica de la Ecuación de Laplace en 2D usando Diferencias Finitas y el Método Gauss

Facultad de ciencia naturales y matematicas

11 de diciembre de 2025

1. Formulación Matemática y Metodología Numérica

1.1. Ecuación de Laplace Bidimensional

El problema físico consiste en resolver la ecuación de Laplace en dos dimensiones para el potencial electrostático $\phi(x, y)$ en un dominio rectangular $[0, L] \times [0, H]$:

$$\nabla^2 \phi(x, y) = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (1)$$

definida en un dominio rectangular:

$$0 \leq x \leq L, \quad 0 \leq y \leq H.$$

sujeta a condiciones de frontera de Dirichlet, donde tres bordes tienen potencial constante y un cuarto borde presenta una variación espacial específica, según uno de los ocho casos propuestos en el enunciado.

1.2. Discretización por Diferencias Finitas

Se discretiza el dominio con una malla uniforme de $N_x \times N_y$ puntos:

$$x_i = i\Delta x, \quad \Delta x = \frac{L}{N_x - 1}, \quad i = 0, 1, \dots, N_x - 1 \quad (2)$$

$$y_j = j\Delta y, \quad \Delta y = \frac{H}{N_y - 1}, \quad j = 0, 1, \dots, N_y - 1 \quad (3)$$

Usando diferencias finitas centradas de segundo orden:

$$\frac{\partial^2 \phi}{\partial x^2} \approx \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta x)^2} \quad (4)$$

$$\frac{\partial^2 \phi}{\partial y^2} \approx \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{(\Delta y)^2} \quad (5)$$

Para simplificación y eficiencia computacional, se considera $\Delta x = \Delta y = h$, obteniendo la ecuación discreta:

$$\phi_{i,j} = \frac{1}{4} (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1})$$

Esta ecuación es válida únicamente para los puntos interiores del dominio.

1.3. Implementación del Método SOR (Successive Over-Relaxation)

El método Gauss–Seidel actualiza iterativamente, pero su convergencia puede ser lenta. Para mejorarla, se añade un parámetro de sobre-relajación ω :

$$\phi_{i,j}^{(k+1)} = (1 - \omega) \phi_{i,j}^{(k)} + \omega \cdot \frac{1}{4} \left(\phi_{i+1,j}^{(k+1)} + \phi_{i-1,j}^{(k+1)} + \phi_{i,j+1}^{(k)} + \phi_{i,j-1}^{(k)} \right). \quad (6)$$

Para $1 < \omega < 2$ la convergencia puede acelerarse notablemente. En este proyecto se utilizó $\omega = 1,85$, adecuado para mallas rectangulares.

El error global se evalúa como:

$$E^{(k)} = \sum_{i,j} \left| \phi_{i,j}^{(k+1)} - \phi_{i,j}^{(k)} \right|,$$

y la iteración se detiene cuando:

$$E^{(k)} < 10^{-6}.$$

1.4. Condiciones de Frontera Implementadas

Según el caso 8 del enunciado, se implementaron las siguientes condiciones:

- **Borde izquierdo** ($x = 0$): $\phi(0, y) = V_{\text{left}} = 0,0$
- **Borde derecho** ($x = L$): $\phi(L, y) = V_{\text{right}} = 10,0$
- **Borde superior** ($y = H$): $\phi(x, H) = V_{\text{top}} = 5,0$
- **Borde inferior** ($y = 0$): $\phi(x, 0) = \cosh(x)$

La función $\cosh(x)$ se evalúa en cada nodo del borde inferior mediante:

$$\phi_{0,i} = \cosh(x_i) = \cosh(i\Delta x), \quad i = 0, \dots, N_x - 1 \quad (7)$$

1.5. Criterios de Convergencia y Parámetros

- **Tolerancia:** $\epsilon_{\text{tol}} = 10^{-6}$
- **Error calculado:** Suma absoluta de diferencias entre iteraciones sucesivas
- **Máximo de iteraciones:** 5×10^5 para evitar bucles infinitos
- **Dimensión de malla:** $N_x = N_y = 51$ puntos en cada dirección
- **Dominio:** $L = H = 5,0$ unidades de longitud

1.6. Implementación Computacional

La solución se implementó en C++ usando:

- Matriz bidimensional de tamaño $(N_y \times N_x)$ para almacenar ϕ
- Inicialización con ceros en todos los puntos interiores
- Aplicación explícita de condiciones de frontera antes de iterar
- Almacenamiento de resultados en archivo de texto para posterior visualización

El método SOR garantiza convergencia para la ecuación de Laplace discretizada, siendo particularmente eficiente cuando se selecciona un valor óptimo de ω .