

Métodos numéricos para la Ciencia e Ingeniería

Tarea 5: Integración de Ecuación de Poisson 2D

Felipe Toledo Bittner

October 28, 2015

1 Introducción

En electrostática la Ecuación de Poisson (1) describe el potencial eléctrico V a partir de la distribución espacial de carga libre ρ en el sistema, con ϵ una constante del medio físico:

$$\nabla^2 V(\vec{x}) = \frac{-\rho(\vec{x})}{\epsilon} \quad (1)$$

Si la distribución de las cargas eléctricas es muy compleja, resolver esta ecuación de forma analítica puede ser imposible. Debido a que en experimentos reales la carga no necesariamente se ordena de forma sencilla, se vuelve necesario el estudiar e implementar métodos numéricos que permitan resolver esta ecuación para mayor diversidad de geometrías.

En este trabajo se implementa un programa que permite resolver la ecuación de Poisson en dos dimensiones para una geometría de difícil integración analítica utilizando el algoritmo de relajación.

1.1 Descripción del problema

Se debe integrar la Ecuación de Poisson en 2D para una caja rectangular con los siguientes requerimientos:

1. Dimensiones de 10 [cm] x 15 [cm]
2. Condiciones de borde:
 - (a) Potencial $V = 0$ en todo el perímetro.
 - (b) C.B. derivativa: $\frac{dV}{dy} = 1$ en la línea $x = [-3 : 3]$; $y = -5.5$ [cm] respecto al centro de la caja.
3. Letra "F" de carga total $1[C]$ encerrada en un rectángulo centrado de 5 [cm] x 7 [cm].

2 Implementación

2.1 Software

La solución del problema se realizó implementando los archivos *main.py* y *box.py*.

El archivo *box.py* es una clase que contiene todos los métodos necesarios para operar y simular el problema solicitado, incluyendo elementos que permiten operar con coordenadas en centímetros, dibujar cargas en la caja, escribir condiciones de borde incluyendo derivativas e integrar la Ecuación de Poisson.

Por otro lado el archivo *main.py* es el archivo ejecutable. Se utiliza para crear objetos caja, definir sus parámetros, iniciar las operaciones de cálculo y presentar los resultados.

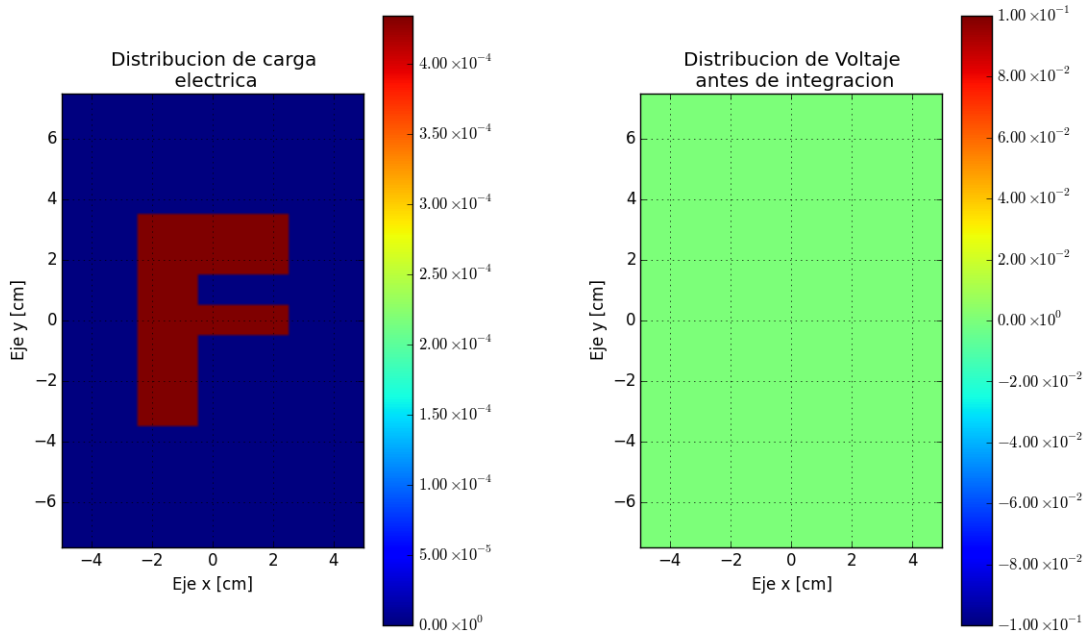


Figura 1: Situacion inicial de las cajas de carga y voltaje. Se observa la letra 'F' dibujada en la caja de carga y las condición $V = 0$ en todo el borde de la caja de potencial. Las unidades de carga son Coulomb y las de potencial, Voltaje.

La caja se representa usando dos matrices con resolución $h = 0.1$ [cm]. Una matriz se utiliza para almacenar los valores de voltaje y condiciones de borde y la otra almacena la distribución de carga en el espacio. En la Figura (1)

Esta solución posee la ventaja de almacenar la carga de inmediato en memoria. Así, para calcular el potencial en cada punto basta leer su casilla correspondiente de la matriz de carga para obtener su valor local. Con ellos se simplifica la implementación del programa y se adquiere más velocidad al privilegiar un acceso directo a memoria sobre otras alternativas como consulta de índices o árboles de decisión.

2.2 Algoritmo de relajación en 2D

El algoritmo de relajación consiste en calcular el valor de cada punto de una matriz usando los adyacentes durante sucesivas iteraciones, deteniéndose cuando se alcanza convergencia. Por convergencia se entiende el instante en que la diferencia entre dos iteraciones es menor a una determinada tolerancia para todos los puntos. En el caso de la ecuación de poisson, cada iteración del algoritmo de relajación calcula el voltaje de la celda (i, j) usando la ecuación (2). Para resaltar el efecto de las cargas se escoge $\epsilon = 5 \cdot 10^{-4}$ [F/m].

$$V_{i,j}^{next} = \frac{1}{4}(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} + h^2 \frac{\rho_{i,j}}{\epsilon}) \quad (2)$$

En el programa se implementa una modificación a este algoritmo, que se puede observar en la fórmula (3). Con ésta se aprovecha que el cálculo de dos de los términos de la siguiente iteración se realizaron con anterioridad, que junto con el nuevo parámetro w permiten acelerar la convergencia.

$$V_{i,j}^{next} = (1 - w)V_{i,j} + \frac{w}{4}(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} + h^2 \rho_{i,j}) \quad (3)$$

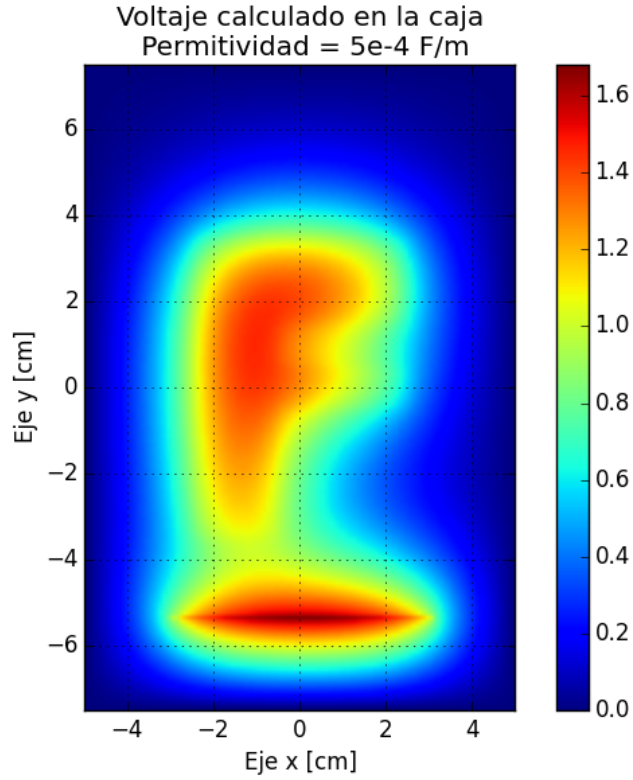


Figura 2: Resultado del algoritmo de integración. Se observa claramente el efecto de la distribución de las cargas y las condiciones de borde en el potencial.

Se escoge una tolerancia de 10^{-7} para determinar la finalización del algoritmo. Si se realizan más de 800 iteraciones se asume que no hay convergencia bajo los parámetros entregados y se interrumpe la ejecución.

2.3 Condición de borde derivativa

Para implementar la condición de borde derivativa se discretiza la derivada $\frac{dV}{dy} = 1$, quedando:

$$\frac{(y_{i,j_0+1} - y_{i,j_0})}{h} = 1 \rightarrow y_{i,j_0+1} = h + y_{i,j_0} \quad (4)$$

Donde j_0 es el índice de la casilla inmediatamente inferior a la línea $y = -5.5$ [cm], y los i corresponden a los elementos ubicados en $x = [-3 : 3]$ [cm]. Luego de cada iteración del algoritmo de relajación se impone nuevamente la condición de borde, recalculando y_{i,j_0+1} . Como $h = 0.1$, la línea queda justo entre dos casillas, evitándose que queden puntos sobre ella. Esto simplifica la implementación ya que no hay que calcular potencial justo encima de la condición de borde.

3 Resultados

Los resultados del algoritmo de cálculo pueden verse en la Figura (2). Como se indicó en la sección 2.2, se escoge un ϵ tal que el potencial causado por las cargas pueda apreciarse junto a la condición de borde.

4 Conclusiones

El algoritmo de relajación ha mostrado ser una herramienta simple y bastante eficaz para resolver el problema de integración de la ecuación de Poisson. Se debe notar que si se utilizase una distribución de carga con dependencia temporal $\rho(\vec{x}, t)$, el algoritmo de relajación puede ser aplicado para calcular el potencial de la caja en cada instante de la malla de tiempos, permitiendo simular sistemas con comportamiento dinámico en régimen electro cuasiestático.

Además se puede intuir que implementar este algoritmo en sistemas de tres dimensiones debiese ser análogo, lo que abre posibilidades de aplicación de versiones mejoradas de este programa en áreas como la ingeniería, en particular para asistir el diseño de condensadores y elementos similares que operen en bajas frecuencias.

En el área de software, tanto el algoritmo como el uso de la caja de cargas permiten concluir que una buena forma de acelerar los cálculos es evitar el cálculo redundante aprovechando siempre que se pueda la posibilidad de utilizar datos almacenados en la memoria.