Métodos Numéricos para la Ciencia e Ingeniería
Informe Tarea 5

Benjamín Guerrero
18.862.370-0
27 de Octubre, 2015

1. Introducción

Se busca integrar la ecuación de Poisson en 2D para el potencial electrostático:

$$\nabla^2 V(x,y) = -\rho(x,y)$$

Aquí, V es el potencial, y ρ es la densidad de carga libre. Se debe integrar dentro de una caja de 10 cm x 15 cm, con potencial nulo en los bordes. Se define el origen $(x,y) = (0{,}0)$ en el centro de la caja. Dentro de la caja, hay una línea en $y = -5.5$, $x = [-3{,}3]$, en la que se cumple la siguiente condición de borde:

$$\frac{dV}{dn} = \pm 1$$

Aquí, la derivada perpendicular a la línea es positiva si $y > -5.5$, y negativo si $y < -5.5$, según el vector normal (así, la derivada es continua)

Además en la caja hay una letra, la primera letra del nombre de la persona (en este caso es B). Dicha letra se encuentra en un rectángulo centrado de lados 5 cm x 7 cm. El espesor de las líneas de la letra es de 1 cm. La letra contiene una densidad de carga constante, con una carga total de 1 Coulomb, de la siguiente forma:

$$Q = \int \rho(x,y) dx dy = 1[C]$$

Se pide usar el método de sobre relajación sucesiva.

2. Procedimiento

Para resolver el problema, se crea un archivo llamado *Poisson.py*. Todas las funciones y el código principal se ubicarán aquí.

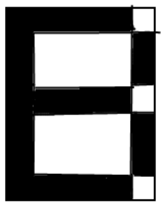
Primero, se definen Lx y Ly como los largos totales de x e y, o sea, 10 y 15 centímetros. Luego, se define h como 0.2, lo que determinará el número de pasos (*N_pasos)* en x e y (dividiendo los largos por h).

Luego, usando la función auxiliar *crear_caja*, se crean tres *np.array* de 2 dimensiones llamados V_actual, V_next, y *carga_caja*, llenados con ceros. La primera dimensión (x) tiene *N_pasos_x* ceros, y la segunda dimensión (y) tiene *N_pasos_y* ceros. Esto termina la inicialización.

Ahora, se puede definir la densidad de carga como:

$$\rho(x,y) = \begin{cases} \frac{1}{A_B} & dentro\ de\ la\ letra \\ 0 & fuera\ de\ la\ letra \end{cases}$$

Donde $A_B$es el área que ocupa la letra B dentro de la caja. Dicha letra se va a definir más o menos así:



Así, si el espesor de las líneas es de 1cm, entonces $A_B = 20\ cm^2 = 2 * 10^{-3}\ m^2$.

De esta forma, primero se define la función *trazo*, que permite dibujar una línea recta con el ancho indicado tanto vertical como horizontalmente, apoyándose de la función auxiliar *es_horizontal*.

Luego, se define la función *armar_letra_B*, que, usando la función *trazos*, crea la letra B, tal como se pide. Finalmente, usando la función *poner_carga*, se redefine *carga_caja* como las cargas en cada punto de la letra.

Ahora, en *V_actual* se usa la función llamada *poner_condiciones_borde*, que define la condición de borde señalada (el potencial es 0 en el borde de la caja).

Luego, para calcular todos los valores de V, se crea una función llamada *Iteracion*, que itera la función según el método de sobre relajación sucesiva, en este caso:

$$V_{i,j}^{n+1} = (1-\omega)V_{i,j}^{n} + \frac{\omega}{4}\left(V_{i+1,j}^{n} + V_{i-1,j}^{n+1} + V_{i,j+1}^{n} + V_{i,j-1}^{n+1} + h^2\rho(x,y)\right)$$

Aquí, ω puede ser un número entre 1 y 2. En este caso, será 1.2.

Ahora, la ecuación superior sirve para la mayoría de los casos, pero en la línea de la condición derivativa, se debe ocupar una ecuación modificada para la condición de borde.

Sobre la línea:

$$V_{i,j}^{n+1} = (1-\omega)V_{i,j}^{n} + \frac{\omega}{3}\left(V_{i+1,j}^{n} + V_{i-1,j}^{n+1} + V_{i,j+1}^{n+1} - h\frac{dV}{dy}\right)$$

Bajo la línea:

$$V_{i,j}^{n+1} = (1-\omega)V_{i,j}^{n} + \frac{\omega}{3}\left(V_{i+1,j}^{n} + V_{i-1,j}^{n+1} + V_{i,j-1}^{n+1} + h\frac{dV}{dy} \right)$$

Y en la línea:

$$V_{i,j}^{n+1} = \left(V_{i,j-1}^{n+1} + h\frac{dV}{dy} \right)$$

En este caso, $\frac{dV}{dy} = 1$.

Se crea una función auxiliar para cada caso.

La iteración se realiza hasta que hayan 500 iteraciones (se usa un contador), o hasta que la función converja. Para esto, se crea una función llamada *no_converge*, que retorna True si la función no converge, y False si la función converge, con una tolerancia de 0.1.

Una vez hecha la iteración, es hora de graficar los resultados. Para no tener que escribir el código 2 veces, se crea una función llamada *mostrar*, que crea los gráficos que muestran la distribución de carga y el valor del potencial en el espacio, y los guardan en archivos jpg. También se imprime la cantidad de iteraciones hechas.

3. Resultados

Se ve que la función converge luego de 299 iteraciones.

Ahora, los gráficos creados por la función *mostrar* son los siguientes:

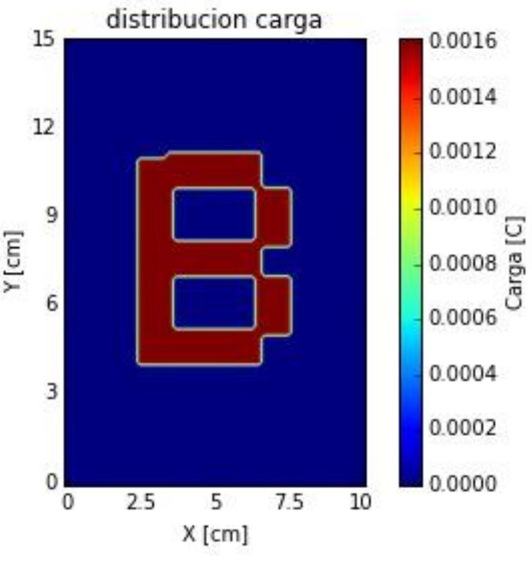


Fig 1: Distribución de carga en el espacio.

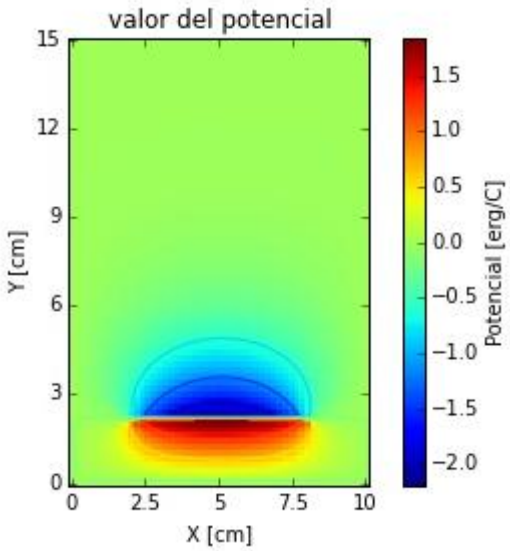


Fig 2: Valor del potencial en el espacio.

4. Conclusión

Primero, se ve que la función converge luego de casi 300 pasos. Esto es debido al valor relativamente pequeño de ω. Si fuera mayor, la función convergería más rápido.

Se aprecia que la carga se pudo distribuir relativamente bien en la letra. No es 100% exacta (pareciera que le falta un pedazo en la parte superior izquierda), pero es una buena aproximación. Es posible que el error se deba a la definición de *trazo* hecha en el programa.

Finalmente, y lo más importante, se nota que el potencial varía mucho en la vecindad de la línea en $y = -5{,}5$, la condición derivativa, como era de esperarse. No se aprecia una variación de potencial debido a la letra, pero esto se debe a que la carga distribuida en la letra es relativamente pequeña. Con una carga mayor, tal vez sea posible apreciar una variación en la letra.