Métodos Numéricos para la Ciencia e Ingeniería

Benjamín Oliva

27 de Octubre, 2015

1 Pregunta 1

Se pide integrar la ecuación de Poisson en 2 Dimensiones para el potencial electrostático, dada por:

$$\nabla^2 V(x,y) = -\rho(x,y) \tag{1}$$

se pide que esta integración se haga dentro de una caga de $10[cm]$ x $15[cm]$ conectada a tierra. En la caja hay una linea que cumple con las condiciones de borde derivativas dadas por:

$$\frac{dV}{dn} = \pm 1 \tag{2}$$

tomando el signo + cuando la derivada perpendicular a la linea es calculada sobre ella y el signo negativo cuando la derivada es calculada bajo ella. Aparte de todo esto, dentro de la caja hay una letra 'B' (la cual proviene del primer nombre del alumno) contenida dentro del rectángulo centrado con lados $5[cm]$ x $7[cm]$. Dentro del área que contiene la letra, la densidad de carga es constante y con una carga total de $1[C]$. La carga total viene dada por:

$$Q = \int \rho(x,y)dxdy = 1[C] \tag{3}$$

de la ecuacion 3 se puede despegar la densidad en funcion de la carga total, llegando a la siguiente ecuación:

$$Q = \rho A \tag{4}$$

donde A es el área del sector ocupado por la letra.

1.1 Procedimiento

Para realizar la integración de la ecuación 1 con las condiciones antes explicadas, con un reticulado de $h = 0.2$, se procedió de la siguiente manera (detalles en el archivo *codigo.py*, también se puede observar el avanze del código en los distintos commits):

1. Se creó la grilla con las dimensiones solicitadas, y se le otorgaron los valores correspondientes de densidad de carga a cada punto de la grilla, dandoles un valor de $\rho = 0$ fuera de la letra, y un valor de $\rho = \frac{1}{23}$ dentro de la letra.

2. Se creó una función que realiza una iteración sobre la grilla, aplicando el método de *sobre-relajación*. Teniendo en cuenta que habían condiciones de borde derivativas, fueron dos las fórmulas iteradas:

$$V_{i,j} = (1-w)V_{i,j} + \frac{w(V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} + h^2\rho(x,y))}{4} \tag{5}$$

para el área fuera de la condición derivativa, y:

$$V_{i,j} = (1-w) * V_{i,j} + \frac{w(V_{i+1,j} + v_{i-1,j} + v_{i,j-1} + h^2\rho(x,y) + hg_i)}{3} \tag{6}$$

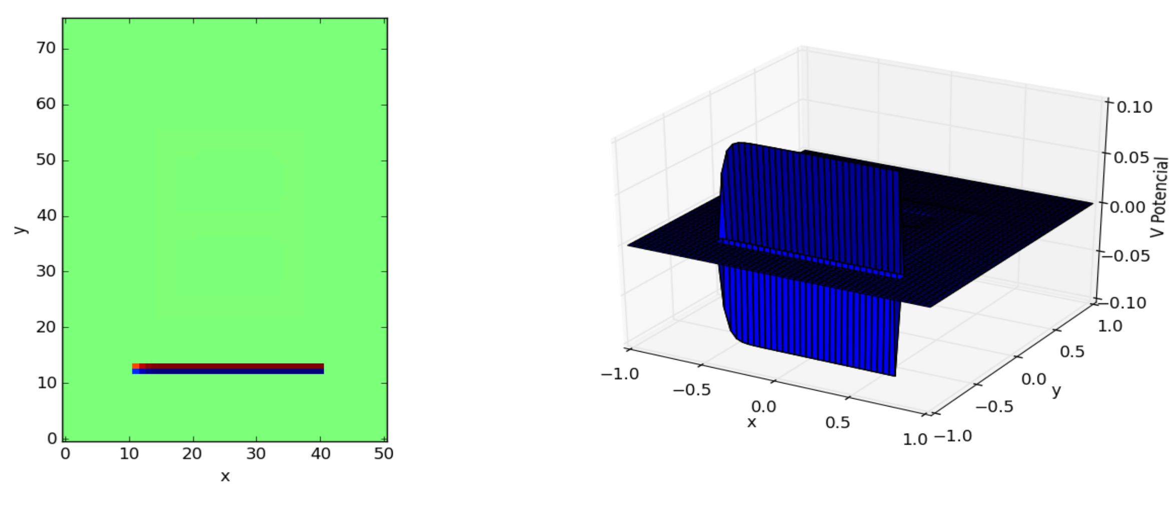
para dentro de la condición derivativa, donde $g_i = \pm 1$ dado por la ecuación 2.

3. Se implementó un algoritmo para iterar la función del punto anterior una cantidad limitada de veces o considerando un cirterio de convergencia, el cual consistió en ver la diferencia entre un paso y el paso siguiente. Si esta diferencia era mayor que la variable *tolerancia*, entonces se consideraba que aun no convergía . Luego se grafican los resultados para distintos valores de w y de tolerancia (detalles en la Tabla 1).

1.2 Resultados

Inicialmente probó con una tolerancia muy alta ($tolereancia = 1$) y un valor de $w = 1$ para probar que el criterio de convergencia funcionaba bien. Éste convergió en una sola iteración y se obtuvieron los gráficos de la Figura 1.

También se puso una tolerancia muy baja pero con un límite de iteraciones muy bajo



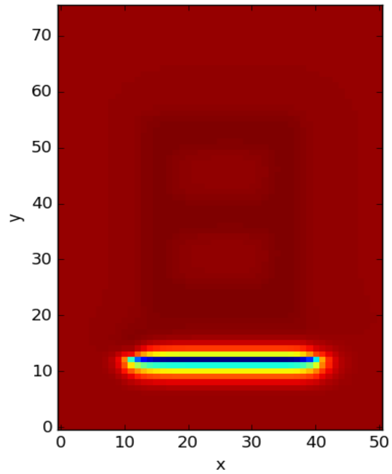
(a) Gráfico de profundidad para $w = 1$

(b) Gráfico en 3D para $w = 1$

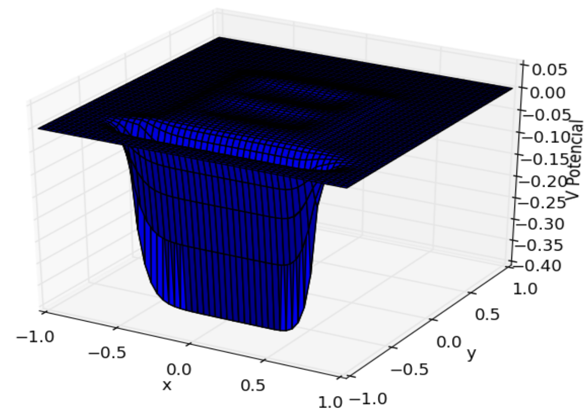
Figure 1: Gráficos obtenidos luego de 1 iteración.

también, obteniéndose los gráficos de la figura 2.

Luego se fueron cambiando los valores de w y de la tolerancia, obteniéndose los siguientes números de iteraciones para distintos valores de tolerancia (Tabla 1).



(a) Gráfico de profundidad para $w = 1$



(b) Gráfico en 3D para $w = 1$

Figure 2: Gráficos obtenidos luego de 10 iteraciones y sin converger.

Table 1: Número de iteraciones para distintos valores de tolerancia y de w.

	Iteraciones (Tolerancia = 10^{-1})	Iteraciones (Tolerancia = 10^{-2})	Iteraciones (Tolerancia = 10^{-3})
$w = 1.0$	1616	1693	2054
$w = 1.2$	1080	1152	1436
$w = 1.4$	696	762	972

Luego se graficó la solución para $w = 1.4$ con una $tolerancia = 10^{-5}$ (figura 3).

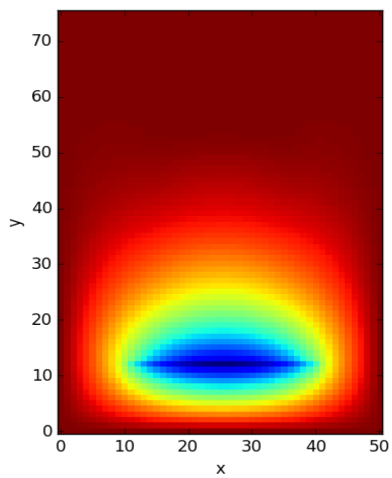
La figura 3 es muy similar a los gráficos obtenidos con las distintas variaciones que se encuentran en la Tabla 1, dado que en todas ellas la solución convergió.

1.3 Conclusiones

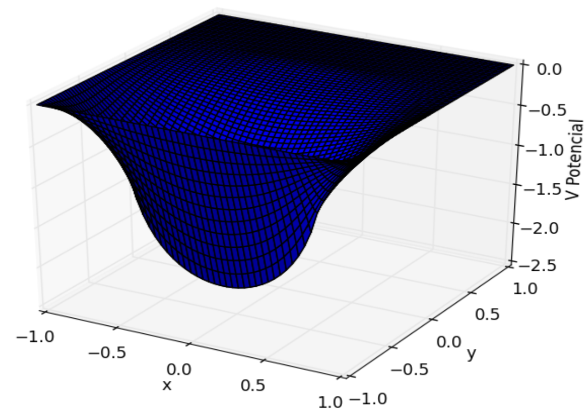
En primer lugar, se puede concluir que el método de sobre-relajación es un muy buen método al momento de resolver Ecuaciones de Derivadas Parciales, ya que, como se observó en la experiencia, se comportó de muy buena manera al hacer variar los valores de w y *tolerancia*. En particular, se comporta de manera excelente para un $w = 1.4$, ya que es el que converge más rápido, sin perder precisión, como se observa en la Tabla 1 y la figura 3.

También se puede destacar la importancia de escribir un código ordenado (según la convención) y estructurado, ya que esto facilita lo que a futuro puede ser la optimización del mismo programa.

Cabe destacar que el criterio de convergencia aplicado (que fue el visto en clases) es muy eficiente ya que permitió obtener el número de iteraciones necesarias para que la solución convergiera para los distintos valores de w y *tolerancia*.



(a) Gráfico de profundidad para $w = 1.4$



(b) Gráfico en 3D para $w = 1.4$

Figure 3: Gráficos obtenidos al congerger luego de 1565 iteraciones.