

Tarea 3 EM

Alejandro Borda

Marzo 2024

1 Marco teórico

En este documento buscamos estudiar el método de relajación para la ecuación de Laplace desde un punto de vista teórico y mas adelante con una correspondiente implementación computacional. En particular, considere la ecuación de Laplace del potencial V , dada por

$$\Delta V(\vec{r}) = 0 \quad (1)$$

donde $\vec{r} = (x, y)$ y estudiamos la región rectangular limitada por $0 \leq x \leq a$ y $0 \leq y \leq b$ con condiciones de frontera de Dirichlet $V(\vec{r}) = V_0(x, y)$ para elementos de la frontera. Ahora consideremos la discretización de la región en donde esta región se convierte en una red de $N \times M$ sitios separados por las constantes (dx, dy) . De esta forma podemos expresar las coordenadas x y y como

$$x_i = i \cdot dx \quad (2)$$

$$y_j = j \cdot dy \quad (3)$$

donde $i = 0, 1, \dots, n$ tal que $n \cdot dx = a$ y $j = 0, 1, \dots, m$ tal que $m \cdot dy = b$. Ahora podemos expandir V por medio de estas definiciones de las coordenadas y obtenemos tres diferencias para cada coordenada dadas por

$$\left. \frac{\partial V}{\partial x} \right|_{i,j} = \frac{V_{i+1,j} - V_{i,j}}{dx} + O(dx)^2 \quad (4)$$

$$\left. \frac{\partial V}{\partial x} \right|_{i,j} = \frac{V_{i,j} - V_{i-1,j}}{dx} + O(dx)^2 \quad (5)$$

$$\left. \frac{\partial V}{\partial x} \right|_{i,j} = \frac{V_{i+1,j} - V_{i-1,j}}{2dx} + O(dx)^2 \quad (6)$$

de tal forma que

$$\left. \frac{\partial^2 V}{\partial x^2} \right|_{i,j} = \frac{V_{i+1,j} + V_{i-1,j} - 2V_{i,j}}{(dx)^2} + O(dx)^2 \quad (7)$$

De forma completamente análoga podemos expresar la segunda derivada parcial con respecto a y por medio de una expansión a segundo orden como

$$\left. \frac{\partial^2 V}{\partial y^2} \right|_{i,j} = \frac{V_{i,j+1} + V_{i,j-1} - 2V_{i,j}}{(dy)^2} + O(dy)^2 \quad (8)$$

Ahora con estas expresiones podemos escribir el Laplaciano discretizado como

$$\frac{V_{i+1,j} + V_{i-1,j} - 2V_{i,j}}{(dx)^2} + \frac{V_{i,j+1} + V_{i,j-1} - 2V_{i,j}}{(dy)^2} = 0 \quad (9)$$

A partir de la expresión (9) consideramos el caso que vamos a solucionar computacionalmente, en donde $dx = dy$, en donde podemos simplificar esta expresión a

$$V_{i+1,j} + V_{i-1,j} - 2V_{i,j} + V_{j+1} + V_{j-1} - 2V_{i,j} = 0$$

$$V_{i,j} = \frac{V_{i+1,j} + V_{i-1,j} + V_{j+1} + V_{j-1}}{4} \quad (10)$$

y notamos que el valor en cada punto de la red es el promedio de sus 4 valores adyacentes si el Laplaciano es nulo.

2 Solución Analítica

Antes de discutir la solución computacional, nos proponemos solucionar de forma analítica un caso para usarlo como un test de la precisión del algoritmo. En particular, consideramos unas condiciones de frontera de las forma

$$V_0(x, y) = \begin{cases} 0, & \text{si } x = 0 \\ 0, & \text{si } x = a \\ -V_0, & \text{si } y = 0 \\ V_0, & \text{si } y = b \end{cases} \quad (11)$$

Como ya hemos estudiado soluciones de esta forma sabemos que la solución general por separación de variables con el tipo de condiciones de frontera dadas tiene la forma general

$$X(x) = A \sin(kx) + B \cos(kx) \quad (12)$$

$$Y(y) = C \sinh(ky) + D \cosh(ky) \quad (13)$$

por condiciones de frontera en $x = 0$ es claro que $B = 0$ y $k = \frac{n\pi}{a}$ de igual manera que hemos resuelto varias veces. Entonces usando la solución general (en forma de serie) y las condiciones de frontera no nulas

$$V(x, 0) = \sum_{n=0}^{\infty} C_n \sin\left(\frac{n\pi}{a}x\right) = -V_0 \quad (14)$$

$$V(x, a) = \sum_{n=0}^{\infty} \sin(kx) (A_n \sinh(kb) - C_n \cosh(kb)) \quad (15)$$

y obtenemos con el truco de Fourier y los procedimientos estudiados en clase (donde k' reemplaza el índice n por n')

$$-V_0 \int_0^a \sin(k'x) dx = \sum_{n=0}^{\infty} C_n \int_0^a \sin(kx) \sin(k'x) dx \quad (16)$$

Entonces, como la sumatoria se reduce a un delta de Kronecker por ortogonalidad de \sin , obtenemos

$$C_n = -\frac{2}{a} V_0 \int_0^a \sin\left(\frac{n\pi'x}{a}\right) dx \quad (17)$$

$$C_n = \begin{cases} 0, & n = 2k \\ -\frac{4V_0}{n\pi}, & n = 2k + 1 \end{cases}, \text{ para } k \in \mathbb{N} \quad (18)$$

Ahora, por la otra condición, de forma analoga tenemos

$$V_0 \int_0^a \sin(k'x) dx = (A_n \sinh(kb) + C_n \cosh(kb)) \int_0^a \sin(k'x) \sin(kx) dx \quad (19)$$

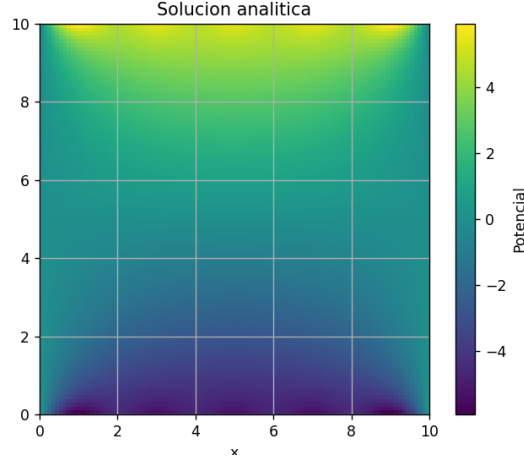


Figure 1: Solución analítica del problema planteado en la sección 2

Dado que ya conocemos la solución de este tipo de integrales solo necesitamos despejar A_n , en donde obtenemos

$$A_n = \begin{cases} 0, n = 2k \\ \frac{4V_0}{n\pi}(\sinh^{-1}(kb) + \tanh^{-1}(kb)), n = 2k + 1 \end{cases}, \text{ para } k \in \mathbb{N} \quad (20)$$

De esta forma la solución final iterando solo sobre los valores n impares es

$$V(x, y) = \frac{4V_0}{\pi} \sum_{n=0}^{\infty} \frac{1}{n} \sin\left(\frac{n\pi}{a}x\right) \left(\sinh^{-1}\left(\frac{n\pi}{a}b\right) + \tanh^{-1}\left(\frac{n\pi}{a}b\right) \right) \sinh\left(\frac{n\pi}{a}y\right) - \cosh\left(\frac{n\pi}{a}y\right) \quad (21)$$

3 Implementación computacional

El método explicado en la sección del marco teórico nos indica que podemos aproximar el valor de los puntos interiores de la red como el promedio de los 4 valores adyacentes. De esta forma por medio de python se implemento un algoritmo en donde se inicializa una red discreta de tamaño $N \times M$ con separación entre puntos $dx = dy$ y se inician todos los puntos de la red con el valor promedio de las 4 condiciones de frontera y la frontera del dominio se reemplaza por su valor correspondiente en cada punto. Una vez se tiene la red inicial definida, se itera el método de relajación en donde cada elemento interno (no de la frontera) de la red se cambia por el promedio de sus 4 valores adyacentes, este proceso se repite un máximo de iteraciones definidas en el programa o se finaliza con anterioridad si el cambio total de todos los sitios interiores de la red es menor a un valor de convergencia definido en el programa. La implementación detallada se puede encontrar con sus correspondientes explicaciones en el repositorio: <https://github.com/AlejandroBorda/TAREA-3-EM-I>.

3.1 Resultados

En la figura 2 se puede encontrar la gráfica resultante de la implementación del algoritmo descrito en la sección anterior para los valores $N = M = 100$, $dx = dy = 0.1$ y con las mismas condiciones de frontera de la solución analítica estudiada en la sección 2, para el valor $|V_0| = 5$. Además se puede encontrar ilustrada de igual manera la solución analítica en la figura 1.

Al ver las dos soluciones es claro que estas se parecen considerablemente, en primera instancia es completamente razonable concluir que el método de relajación es una buena aproximación de la solución real y que además fue implementado correctamente Sin embargo, al examinarlo con mayor cuidado es

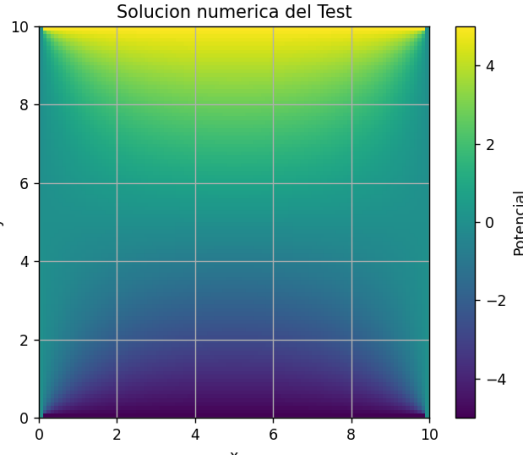


Figure 2: Solución computacional por medio del método de relajación del problema planteado en la sección 2 con $V_0 = 5$ y las constantes $N = M = 100$ y $dx = dy = 0.1$.

claro que cerca de las esquinas superiores del dominio la solución computacional presenta valores mas altos (o bajos para el caso inferior) y pareciera que la propagación del potencial es menos marcada en la solución real. Esto se le puede atribuir a la elección de valores en las esquinas de la frontera pues para estos casos hay claramente dos posibilidades, entonces de modificar la elección en las esquinas en este caso de V_0 a 0 se esperaría una mayor similitud con el valor analítico. Además otra forma mas orgánica de mitigar esta falla es aumentando el numero de puntos en la red lo cual tiene como consecuencia reducir el peso de la elección del valor de la frontera en las esquinas, sin embargo esta solución tiene como problema un incremento en el tiempo de procesamiento del programa.

Además en la figura 3 se puede encontrar el resultado de ejecutar el programa con las condiciones de frontera

$$V_0(x, y) = \begin{cases} 5, & \text{si } x = 0 \\ -5, & \text{si } x = a \\ 10, & \text{si } y = 0 \\ 2, & \text{si } y = b \end{cases} \quad (22)$$

este resultado es prudente para mostrar la generalidad del programa desarrollado y "suavidad" inherente del método incluso en las regiones en donde fronteras cercanas difieren fuertemente en valor.

3.2 Conclusiones

Es razonable concluir que el programa desarrollado implementa satisfactoriamente el método de relajación para solucionar la ecuación de Laplace en un dominio rectangular con condiciones de frontera de Dirichlet, además en base a los resultados obtenidos se puede ver claramente como se ve reflejada la teoría desarrollada en la primera sección con respecto a la suavidad de la solución y la forma en la que se propaga el promedio del potencial. Además, se analizaron las ligeras diferencias entre el modelo teórico y computacional y como se podrían mejorar a cambio de un mayor costo computacional. La implementación particular con su respectiva documentacion se puede encontrar en GitHub: <https://github.com/AlejandroBorda/TAREA-3-EM-I>.

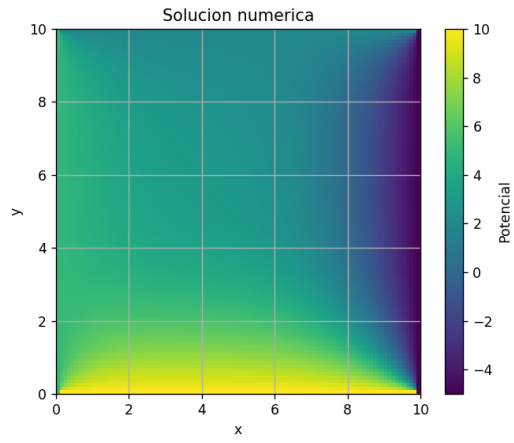


Figure 3: Solución numérica para condiciones de frontera homogéneas no nulas descritas en la sección 3.1 y constantes N , M , $dx = dy$ iguales a previas iteraciones.