

Métodos iterativos para calcular potenciales electrostáticos

ROBERTO ROMERO, JACINTO PINTO Y MARCO REYES

Universidad Nacional Autónoma de Honduras

Resumen

Al resolver en forma numérica las ecuaciones diferenciales en derivadas parciales surgen sistemas de ecuaciones lineales muy grandes, los métodos iterativos constituyen una alternativa a los métodos de eliminación. En este trabajo se revisan los métodos iterativos de Gauss Seidel e iteración con relajación para resolver los sistemas de ecuaciones lineales que surgen al trabajar numéricamente con las ecuaciones de Laplace, Poisson y de guía de Ondas, bajo el esquema de diferencias finitas.

Se proporcionan ejemplos concretos, para calcular potenciales electrostáticos y los programas en Matlab para automatizar los cálculos los cuales se vuelven inmanejables para calcularlos manualmente

Palabras clave: métodos iterativos, ecuación de Laplace, ecuación de Poisson, guía de onda, problema de Dirichlet

When solving numerically the equations in partial differential, arise systems of linear equations very large, the iterative methods are an alternative to disposal methods. In this document we review the iterative methods of Gauss Seidel and iteration with relaxation to solve the systems of linear equations that arise when numerically working with equations of Laplace, Poisson and Wave Guide, under the finite difference scheme.

In this document concrete examples are provided for calculate electrostatic potentials and the Matlab programs to automate calculations which become unmanageable if calculated manually

Keywords: Iterative methods, Laplace equation, Poisson equation, waveguide, Dirichlet problem

I. MÉTODOS ITERATIVOS

UN método iterativo muy utilizado para resolver sistemas de ecuaciones es el de Gauss-Seidel, suponga que se da un sistema de n ecuaciones:

$$[A] \{X\} = \{B\} \quad (1)$$

Si nos limitamos a un conjunto de ecuaciones de 3×3 y los elementos de la diagonal no todos son cero, la ecuación (1) se escribe

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

resolviendo para x_1 , x_2 y x_3 , se obtiene

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}} \quad (2)$$

$$x_2 = \frac{b_2 - a_{21}x_1 - a_{23}x_3}{a_{22}} \quad (3)$$

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}} \quad (4)$$

Ahora, se puede empezar el proceso de solución al escoger valores iniciales para las x . Una forma simple para obtener los valores iniciales es suponer que todos son cero [2],[4].

Estos ceros se sustituyen en las ecuación (2), la cual se utiliza para calcular un nuevo valor $x_1 = b_1/a_{11}$. Después, se sustituye este nuevo

valor de x_1 junto con el valor previo de cero de x_3 en la ecuación (3) y se calcula el nuevo valor de x_2 . Este proceso se repite con la ecuación (4) para calcular el nuevo valor de x_3 . Después se regresa a la primera ecuación y se repite todo el procedimiento hasta que la solución converja suficientemente cerca a los valores verdaderos. La convergencia se verifica usando el criterio

$$|\varepsilon_{a,i}| = \left| \frac{x_i^j - x_i^{j-1}}{x_i^j} \right| < \varepsilon_s \quad (5)$$

para todas la i , donde j y $j-1$ son las iteraciones actuales y previas, respectivamente [1], [4]

I. Mejoramiento de la convergencia usando relajación

La relajación representa una ligera modificación al método de Gauss-Seidel y está permite mejorar la convergencia. Después de que se calcula cada nuevo valor de x por medio de las ecuaciones (2), (3) y (4), ese valor se modifica mediante un promedio ponderado de los resultados de las iteraciones anterior y actual:

$$x_i^{\text{nuevo}} = \omega x_i^{\text{nuevo}} + (1 - \omega) x_i^{\text{anterior}} \quad (6)$$

donde ω es un factor ponderado que tiene un valor entre 0 y 2.

Si $\omega = 1$, $(1 - \omega)$ es igual a cero y el resultado no se modifica. Sin embargo, si a ω se le asigna un valor entre 0 y 1, el resultado es un promedio ponderado de los resultados actuales y anteriores. Este tipo de modificación se conoce como sobrerelajación. Se emplea comúnmente para hacer un sistema no convergente, converja o apresure la convergencia al amortiguar sus oscilaciones.

Para ω de 1 a 2, se le da una ponderación extra al valor actual, en este caso el nuevo valor se mueve en la dirección correcta hacia la solución verdadera, a este tipo de modificación se le llama sobrerrelajación [1], [3].

II. ECUACIÓN DE LAPLACE

El primer paso consiste en obtener una versión discretizada del operador de Laplace que

nos permita usarlo numéricamente. La fórmula para $f''(x)$ es

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2) \quad (7)$$

así que, al aplicar esta fórmula a la función Φ para aproximar Φ_{xx} y Φ_{yy} y sumar los resultados, obtenemos

$$\begin{aligned} \nabla^2 \Phi &= \frac{\Phi(x+h, y) + \Phi(x-h, y)}{h^2} \\ &+ \frac{\Phi(x, y+h) + \Phi(x, y-h) - 4\Phi(x, y)}{h^2} \\ &+ O(h^2) \end{aligned} \quad (8)$$

Ahora dividimos el rectángulo

$$R = \{(x, y) : 0 \leq x \leq a, 0 \leq y \leq b\}$$

en $(n-1) \times (m-1)$ cuadrados de lado h ($a = nh$ y $b = mh$), como se muestra en la figura 1.

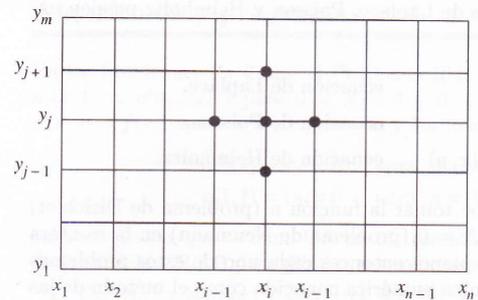


Figura 1: Malla usada en la ecuación en diferencias de Laplace [1]

Para resolver la ecuación de Laplace, imponemos la aproximación

$$\begin{aligned} &\frac{\Phi(x+h, y) + \Phi(x-h, y) + \Phi(x, y+h)}{h^2} \\ &+ \frac{\Phi(x, y-h) - 4\Phi(x, y)}{h^2} = 0 \end{aligned} \quad (9)$$

que tiene una precisión de orden $O(h^2)$ en los puntos interiores de la malla (x_i, y_i) para $i=2, \dots, n-1$ y $j=2, \dots, m-1$.

Como los puntos de la malla están espaciados uniformemente: $x_{i+1} = x_i + h$, $x_{i-1} = x_i - h$, $y_{i+1} = y_i + h$ e $y_{i-1} = y_i - h$; denotando por

$\Phi_{i,j}$ la aproximación al valor $\Phi(x_i, x_j)$, la ecuación (9) queda

$$\begin{aligned} \nabla^2 \Phi_{i,j} &= \frac{\Phi_{i+1,j} + \Phi_{i-1,j}}{h^2} \\ &+ \frac{\Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{i,j}}{h^2} \\ &= 0 \end{aligned} \quad (10)$$

expresión que se conoce como la fórmula de diferencias con cinco puntos para la laplaciana. Esta fórmula relaciona el valor de la función $u_{i,j}$ con sus cuatro valores adyacentes $u_{i+1,j}$, $u_{i-1,j}$, $u_{i,j+1}$ y $u_{i,j-1}$ como se muestra en la figura 2.

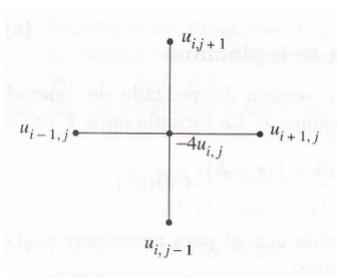


Figura 2: Esquema para la ecuación de Laplace [1].

Eliminando de la ecuación (10) h^2 obtenemos la fórmula de aproximación para la ecuación de Laplace [1, 2].

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0 \quad (11)$$

I. Construcción del sistema lineal

Supongamos que tenemos un problema de Dirichlet, es decir, que conocemos los valores de la función $u(x, y)$ en la frontera de la región R .

Vamos a determinar la solución aproximada de la ecuación de Laplace en el rectángulo

$$R = \{(x, y) : 0 \leq x \leq 4, 0 \leq y \leq 4\}$$

donde $u(x, y)$ denota el potencial en un punto (x, y) , los valores en la frontera son:

$$\begin{aligned} u(x, y) &= 20V \quad u(x, 4) = 180V \\ \text{para } 0 < x < 4 \end{aligned}$$

$$\begin{aligned} u(0, y) &= 80V \quad u(4, y) = 0V \\ \text{para } 0 < y < 4 \end{aligned}$$

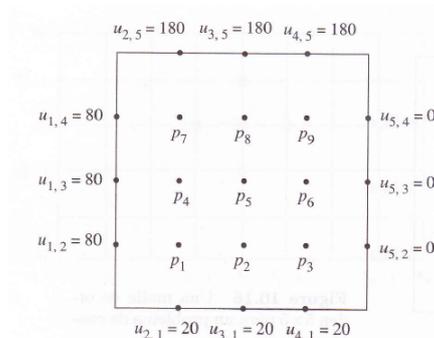


Figura 3: La malla de orden 5×5 [1]

Al aplicar la fórmula (11) en este caso el sistema $AP = B$ que se obtiene, ver figura 3.

Al resolver con Matlab obtenemos los potenciales en los puntos interiores de la malla, expresada en forma vectorial, son

$$\begin{aligned} P &= [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6 \ p_7 \ p_8 \ p_9]' \\ P_1 &= 55.70, P_2 = 43.21, P_3 = 27.14, \\ P_4 &= 79.64, P_5 = 70.00, P_6 = 45.35, \\ P_7 &= 112.5, P_8 = 111.78, P_9 = 84.28 \end{aligned}$$

En el método anterior aparece una matriz en banda, ahora vamos a mostrar un método iterativo para aproximar los potenciales, a continuación la primera iteración. Primero despejamos para $u_{i,j}$ de la ecuación (11)

$$u_{i,j} = \frac{1}{4} [u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}] \quad (12)$$

Evaluando en p_1 :

$$\begin{aligned} p_1 = u_{2,2} &= \frac{1}{4} [u_{3,2} + u_{1,2} + u_{2,3} + u_{2,1}] \\ &= \frac{1}{4} [0 + 80 + 0 + 20] = 25 \end{aligned}$$

evaluando en p_2

$$\begin{aligned} p_2 = u_{3,2} &= \frac{1}{4} [u_{4,2} + u_{2,2} + u_{3,3} + u_{3,1}] \\ &= \frac{1}{4} [0 + 25 + 0 + 20] = 11.25 \end{aligned}$$

$$\begin{aligned}
 -4p_1 + p_2 + 0p_3 + p_4 + 0p_5 + 0p_6 + 0p_7 + 0p_8 + 0p_9 &= -100 \\
 p_1 - 4p_1 + p_3 + 0p_4 + p_5 + 0p_6 + 0p_7 + 0p_8 + 0p_9 &= -20 \\
 0p_1 + p_2 - 4p_3 + 0p_4 + 0p_5 + p_6 + 0p_7 + 0p_8 + 0p_9 &= -20 \\
 p_1 + 0p_2 + 0p_3 - 4p_4 + p_5 + 0p_6 + p_7 + 0p_8 + 0p_9 &= -80 \\
 0p_1 + p_2 + 0p_3 + p_4 - 4p_5 + p_6 + 0p_7 + p_8 + 0p_9 &= 0 \\
 0p_1 + 0p_2 + p_3 + 0p_4 + p_5 - 4p_6 + 0p_7 + 0p_8 + p_9 &= 0 \\
 0p_1 + 0p_2 + 0p_3 + p_4 + 0p_5 + 0p_6 - 4p_7 + p_8 + 0p_9 &= -260 \\
 0p_1 + 0p_2 + 0p_3 + 0p_4 + p_5 + 0p_6 + p_7 - 4p_8 + p_9 &= -180 \\
 0p_1 + 0p_2 + 0p_3 + 0p_4 + 0p_5 + p_6 + 0p_7 + p_8 - 4p_9 &= -180
 \end{aligned}$$

Figura 4: Sistema resultante

evaluando p_3 :

$$\begin{aligned}
 p_3 = u_{4,2} &= \frac{1}{4}[u_{5,2} + u_{3,2} + u_{4,3} + u_{4,1}] \\
 &= \frac{1}{4}[0 + 11.25 + 0 + 20] = 7.81
 \end{aligned}$$

evaluando p_4 :

$$\begin{aligned}
 p_4 = u_{2,3} &= \frac{1}{4}[u_{3,3} + u_{1,3} + u_{2,4} + u_{2,2}] \\
 &= \frac{1}{4}[0 + 80 + 0 + 25] = 26.25
 \end{aligned}$$

evaluando p_5 :

$$\begin{aligned}
 p_5 = u_{3,3} &= \frac{1}{4}[u_{4,3} + u_{2,3} + u_{3,4} + u_{3,2}] \\
 &= \frac{1}{4}[0 + 26.25 + 0 + 11.25] = 9.375
 \end{aligned}$$

evaluando p_6 :

$$\begin{aligned}
 p_6 = u_{4,3} &= \frac{1}{4}[u_{5,3} + u_{3,3} + u_{4,4} + u_{4,2}] \\
 &= \frac{1}{4}[0 + 9.375 + 0 + 7.81] = 4.296
 \end{aligned}$$

evaluando p_7 :

$$\begin{aligned}
 p_7 = u_{2,4} &= \frac{1}{4}[u_{3,4} + u_{1,4} + u_{2,5} + u_{2,3}] \\
 &= \frac{1}{4}[0 + 80 + 180 + 26.25] = 71.56
 \end{aligned}$$

evaluando p_8 :

$$\begin{aligned}
 p_8 = u_{3,4} &= \frac{1}{4}[u_{4,4} + u_{2,4} + u_{3,5} + u_{3,3}] \\
 &= \frac{1}{4}[0 + 71.56 + 180 + 9.375] = 65.23
 \end{aligned}$$

evaluando p_9 :

$$\begin{aligned}
 p_9 = u_{4,4} &= \frac{1}{4}[u_{5,4} + u_{3,4} + u_{4,5} + u_{4,3}] \\
 &= \frac{1}{4}[0 + 65.23 + 180 + 4.296] = 62.38
 \end{aligned}$$

Tras calcular p_9 se inicia la segunda iteración en el primer nodo:

$$\begin{aligned}
 p_1 = u_{2,2} &= \frac{1}{4}[u_{3,2} + u_{1,2} + u_{2,3} + u_{2,1}] \\
 &= \frac{1}{4}[11.25 + 80 + 26.25 + 20] = 34.37
 \end{aligned}$$

evaluando en p_2 :

$$\begin{aligned}
 p_2 = u_{3,2} &= \frac{1}{4}[u_{4,2} + u_{2,2} + u_{3,3} + u_{3,1}] \\
 &= \frac{1}{4}[7.87 + 34.37 + 9.375 + 20] = 17.90
 \end{aligned}$$

y así sucesivamente

II. Aplicación de los métodos iterativos

Acabamos de ver cómo podemos resolver la ecuación en diferencias de Laplace construyendo un cierto sistema de ecuaciones lineales y resolviéndolo. El inconveniente que presenta este método es el almacenamiento. Puesto que para obtener resultados mejores hay que trabajar con una malla más fina, es posible que el número de ecuaciones sea muy elevado. Por ejemplo el cálculo numérico de la solución de un problema de Dirichlet requiere la solución de un sistema de $(n-2) \times (m-2)$ ecuaciones; si dividimos en un número modesto de cuadrados, digamos 10 por 10, entonces tenemos un sistema de 91 ecuaciones con 91 incógnitas. En consecuencia parece sensato trabajar con técnicas que reduzcan la cantidad de datos que se deben almacenar; así un método iterativo solo requeriría que se almacenaran las 100 aproximaciones numéricas $\Phi_{i,j}$ correspondientes a los puntos de malla. Empezando con la ecuación (11)

$$\begin{aligned} &\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1} \\ &+ \Phi_{i,j-1} - 4\Phi_{i,j} = 0 \end{aligned}$$

y supongamos que conocemos los valores de $\Phi(x, y)$ en el contorno:

$$\Phi(x_1, y_j) = \Phi_{1,j} \quad \text{para} \quad 2 \leq j \leq m-1$$

(a la izquierda)

$$\Phi(x_i, y_1) = \Phi_{i,1} \quad \text{para} \quad 2 \leq i \leq n-1$$

(abajo)

$$\Phi(x_n, y_j) = \Phi_{n,j} \quad \text{para} \quad 2 \leq j \leq m-1$$

(a la derecha)

$$\Phi(x_i, y_m) = \Phi_{i,m} \quad \text{para} \quad 2 \leq i \leq n-1$$

(arriba)

Ahora escribimos la ecuación (11) de forma adecuada para iterar:

$$\Phi_{i,j} = \Phi_{i,j} + r_{i,j} \quad (13)$$

siendo

$$\begin{aligned} r_{i,j} = &\frac{\Phi_{i+1,j} + \Phi_{i-1,j} + \Phi_{i,j+1}}{4} \\ &+ \frac{\Phi_{i,j-1} - 4\Phi_{i,j}}{4} \end{aligned} \quad (14)$$

$$\text{para} \quad 2 \leq i \leq n-1 \quad \text{y} \quad 2 \leq j \leq m-1.$$

Es necesario disponer de valores iniciales en los puntos interiores de la malla; para ello puede valer la constante K , definida como la media de los $2n + 2m - 4$ valores en el contorno. Cada paso de la iteración consiste en hacer un barrido de todos los puntos interiores de la malla con la fórmula recursiva (13) hasta que el término residual $r_{i,j}$ que aparece en el miembro derecho de (13) se reduzca a cero, o sea hasta que tenga $|r_{i,j}| < \varepsilon$ para cada $2 \leq i \leq n-1$ y $2 \leq j \leq m-1$, siendo ε una tolerancia prefijada.

Podemos aumentar la velocidad de convergencia a cero de los términos residuales $\{r_{i,j}\}$, usando sobrerelajación sucesiva.

$$\Phi_{i,j} = \Phi_{i,j} + \omega r_{i,j}$$

$$\begin{aligned} \Phi_{i,j} = &\Phi_{i,j} + \omega \left(\frac{\Phi_{i+1,j}}{4} \right) \\ &+ \omega \left(\frac{\Phi_{i,j+1} + \Phi_{i,j-1} - 4\Phi_{i,j}}{4} \right) \end{aligned} \quad (15)$$

En el método de sobrerelajación sucesiva, cada paso de la iteración consiste en hacer un barrido de la malla con la fórmula recursiva (15) hasta que se tenga $|r_{i,j}| < \varepsilon$. Para elegir el valor óptimo del parámetro ω hay que estudiar los autovalores de la matriz que caracteriza el método iterativo que estamos usando para resolver el sistema lineal; en nuestro caso, dicho valor óptimo viene dado por la fórmula [1, 3]

$$\omega = \frac{4}{2 + \sqrt{4 - \left(\cos\left(\frac{\pi}{n+1}\right) + \cos\left(\frac{\pi}{m-1}\right) \right)^2}} \quad (16)$$

III. Solución con software

Ahora se proporciona la solución con el programa 1, desarrollado en Matlab para calcular los potenciales del problema estudiado en la sección 2.1 para cualquier mallado, este programa se adaptó de [1].

Al correr el programa los resultados son:

$$\begin{aligned} p_1 &= 55.7141 & p_2 &= 43.2142 \\ p_3 &= 27.1428 & p_4 &= 79.6428 \\ p_5 &= 69.9999 & p_6 &= 45.3571 \\ p_7 &= 112.8571 & p_8 &= 111.7857 \end{aligned}$$

Los resultados son iguales a los obtenidos con el método de la matriz en banda, la figura 5 muestra el potencial en forma gráfica.

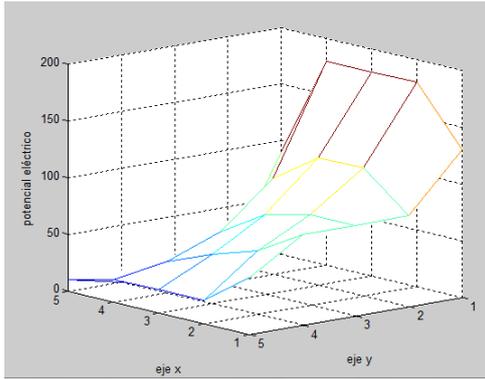


Figura 5: Potencial eléctrico versus x y y

III. ECUACIÓN DE POISSON

La definición del potencial electrostático, combinada con la forma diferencial de la ley de Gauss, provee una relación entre el potencial ϕ y la densidad de carga ρ :

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0}$$

Esta relación es una forma de la ecuación de Poisson.

I. Áreas de Aplicación de la Ecuación de Poisson

En matemática y física, la ecuación de Poisson es una ecuación en derivadas parciales con una amplia utilidad en electrostática, ingeniería mecánica y física teórica. Su nombre se lo debe al matemático, geómetra y físico francés Siméon-Denis Poisson.

II. Ecuación de Poisson en dos Dimensiones

En un sistema de coordenadas cartesianas tridimensional, toma la forma:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \varphi(x, y, z) = -\frac{\rho(x, y, z)}{\epsilon_0}$$

Para explicar el método de sobre relajación, aplicado a la ecuación de Poisson consideremos un ejercicio en el cual la densidad de carga varía de acuerdo a la siguiente expresión: $\rho = x(y-1) \text{ nC/m}^3$ para $0 \leq x, y \leq 1$

Encontraremos el potencial para los puntos mostrados en la figura:

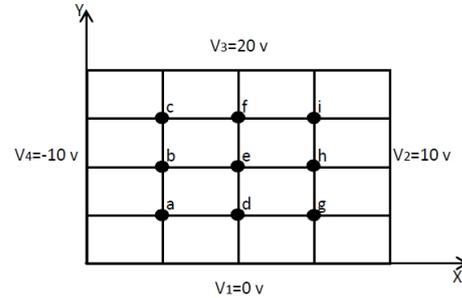


Figura 6: Malla con densidad de carga variable

Este ejercicio tiene solución analítica, pero resolverla utilizando métodos numéricos resulta mucho más fácil. La solución completa se plantea utilizando el principio de superposición:

$$\phi = \phi_1 + \phi_2$$

Donde ϕ_1 es la solución de la ecuación de Laplace. $\nabla^2 \phi_1 = 0$. Y resolvemos con las condiciones de frontera no homogéneas mostradas en la figura 6, y ϕ_2 es la solución de la ecuación de Poisson $\nabla^2 \phi_2 = -\frac{\rho}{\epsilon_0}$ sujeta a las condiciones frontera.

Al utilizar métodos numéricos, a la región rectangular, se debe encontrar el valor óptimo para el factor de sobre relajación, utilizando la siguiente expresión cuadrática:

$$t^2 w^2 - 16w + 16 = 0$$

donde $t = \cos(\pi/N_x) + \cos(\pi/N_y)$ y N_x y N_y son el número de intervalos a lo largo de los ejes x y y , respectivamente.

Trataremos tres casos en los cuales $N_x = N_y = 4, 12$ y 20 con los cuales $\Delta x = \Delta y = h = 1/4, 1/12$ y $1/20$ respectivamente. Al despejar para w queda:

$$w = \frac{8 - \sqrt{64 - 16t^2}}{t^2}$$

Para tener un patrón de comparación de este ejercicio, utilizaremos la solución analítica de este problema, al aplicar las técnicas de solución de la ecuación de Poisson, el potencial se puede expresar como:

$$\phi = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right)$$

donde

$$\begin{aligned}
 A_{mn} &= \int_0^a \int_0^b g(x, y) \operatorname{sen}\left(\frac{m\pi x}{a}\right) \operatorname{sen}\left(\frac{n\pi y}{b}\right) dx dy \\
 &= \frac{1 - \frac{1}{b}[1 - (-1)^n]}{(m\pi/a)^2 + (n\pi/b)^2} \cdot \frac{(-1)^{m+n} 144ab}{mn\pi}
 \end{aligned} \quad (17)$$

Para este caso en particular $a = b = 1$, al correr el script en matlab y sustituir los valores para y y x en la solución exacta de la ecuación diferencial; se obtiene el siguiente cuadro comparativo y el gráfico cuya intensidad representa el potencial en los puntos de la malla.

```

>> poisson
SOR Factor Omega = 1.7295
LA SOLUCION CONVERGE EN 43iteraciones
h = 0.05

```

numerico	exacto
-3.4239	-3.4293
-2.0121	-2.0285
4.2769	4.2773
-0.10872	-0.11822
2.9206	2.9135
9.5784	9.5926
2.9092	2.9019
6.0692	6.0652
11.126	11.133

Figura 7: Comparativo método exacto vs. numérico

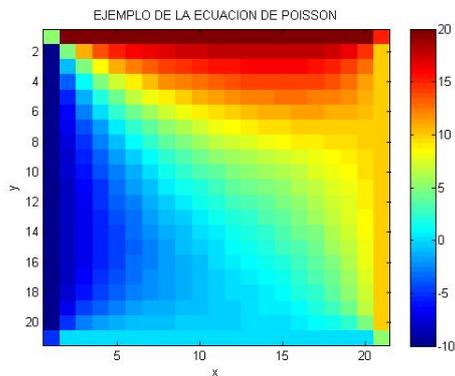


Figura 8: Intensidad de potencial

Ver el programa 2, en la sección de programas.

IV. GUÍA DE ONDAS

En electromagnetismo una guía de onda es cualquier estructura física que guía ondas electromagnéticas.

La solución de los problemas de guía de ondas es muy adecuado para esquemas de diferencias finita debido a que tiene una solución está cerrada. Esto equivale a la solución de la ecuación de onda o Helmholtz:

$$\nabla^2 \phi + K^2 \phi = 0$$

Donde: $\phi = E_Z$ para el modo TM o $\phi = H_Z$ para el modo TE. Si tenemos que k representa el número de onda tenemos:

$$k^2 = \omega^2 \mu \epsilon - \beta^2$$

(TM) Modos transversal Magnético: no depende del campo magnético en la dirección de propagación. Estos a veces se llaman modos E, porque sólo depende del campo eléctrico a lo largo de la dirección de propagación.

(TE) Modos transversal eléctrico: no depende del campo eléctrico en la dirección de propagación. Estos a veces se llaman modos H, porque sólo depende del campo magnético a lo largo de la dirección de propagación.

La permitividad ϵ del medio dieléctrico puede ser real para un medio sin pérdidas o complejo para un medio con pérdidas. Consideramos que todos los campos varían con el tiempo y la distancia axial de la forma $e^{j(\omega t - \beta z)}$ para cada valor propio de la ecuación anterior, tanto k como ϕ están por determinarse. La longitud de onda de corte es $\lambda_c = 2\pi/k_c$. Para cada valor del número de onda de corte k_c , es una solución para la función propia ϕ_i , que representa la configuración de campo de un modo de propagación [3].

Para aplicar el método de diferencia finita, que describe la sección transversal de la guía de ondas por una malla cuadrada. Aplicando la ecuación:

$$\begin{aligned}
 \frac{\partial \phi^2}{\partial x^2} &= \frac{\phi(i+1, j) - 2\phi(i, j) + \phi(i-1, j)}{(\Delta x)^2} \\
 &+ O(\Delta x)^2
 \end{aligned}$$

En la ecuación:

$$\nabla^2 \phi + K^2 \phi = 0$$

Obtenemos la siguiente ecuación:

$$\begin{aligned} \phi(i+1, j) + \phi(i-1, j) + \phi(i, j+1) \\ + \phi(i, j-1) - (4 - h_2^2 k^2) \phi(i, j) = 0 \end{aligned}$$

Donde $\Delta x = \Delta y = h$ representa el tamaño de la malla.

La ecuación Anterior se aplica en el interior de todos los nodos libres en donde en los puntos de frontera, se aplican siguientes condiciones: Condición de Dirichlet para los modos TM:

$$\phi = 0$$

Condición de Neumann para los modos TE:

$$\frac{\partial \phi}{\partial n} = 0$$

Analizando la figura (9) en el punto A:

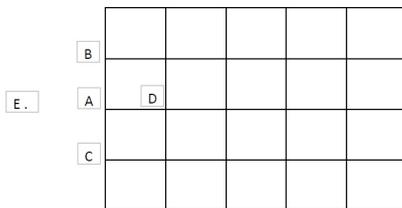


Figura 9: Diferencia finita de malla para una guía de ondas

Luego de aplicar las condiciones de frontera obtenemos:

Condición de Dirichlet para modos TM:

$$\phi_A = 0$$

Condición de Neumann para modos TE:

$$\frac{\partial \phi}{\partial n} = 0$$

Entonces tenemos que:

$$\phi_D = \phi_E$$

Sustituyendo esos términos en la siguiente ecuación:

$$\begin{aligned} \phi(i+1, j) + \phi(i-1, j) + \phi(i, j+1) \\ + \phi(i, j-1) - (4 - h_2^2 k^2) \phi(i, j) = 0 \end{aligned}$$

Logramos reducir términos como sigue:

$$\phi_B + \phi_C + 2\phi_D - (4 - h^2 k^2) \phi_A = 0$$

Mediante la aplicación de las ecuaciones anteriores a todos los puntos de malla en la sección transversal de guía de ondas, obtenemos m ecuaciones simultáneas con m incógnitas $(\phi_1, \phi_2, \phi_3, \dots, \phi_n)$. Entonces podemos de forma conveniente escribir un sistema matricial de ecuaciones de la forma:

$$(A - \lambda I) \phi = 0$$

Entonces encontramos que:

$$A \phi = \lambda \phi$$

Donde sabemos que A es una matriz de $m \times m$ e I representa una matriz identidad.

Deducimos entonces que los vectores propios serán:

$$\phi = (\phi_1, \phi_2, \dots, \phi_n)$$

El valor propio será:

$$\lambda = (kh)^2 = \left(\frac{2\pi h}{\lambda_c} B i g \right)^2$$

Hay varias formas de determinar λ y la correspondiente ϕ nosotros consideraremos dos métodos.

I. Método directo

Para satisfacer la ecuación:

$$(A - \lambda I) \phi = 0$$

Debemos considerar:

$$|A - \lambda I| = 0$$

Esto da como resultado un polinomio en λ que puede ser resuelto para los diversos valores propios de λ . Para cada λ se obtiene la correspondiente ϕ de la ecuación:

$1, 2, \dots, nx - 1, j = 1, 2, \dots, ny - 1$ esto se desarrolla fácilmente relacionando los dos sistemas de numeración. Para determinar el valor de elemento de A_{mn} ... buscamos $NL(i, j)$ para encontrar (i_m, j_m) y (i_n, j_n) , que son los valores de (i, j) correspondientes a los nodos m y n .

III. Ejemplo

Calcular la impedancia Z_0 para la micro sección de la línea de transmisión de la figura 11.

Datos:

$$a = b = 2.5\text{cm}, d = 0.5\text{cm}, w = 1\text{cm},$$

$$t = 0.001\text{cm}, \epsilon_1 = \epsilon_0, \epsilon_2 = 2.35\epsilon_0$$

Desarrollaremos el programa 3 en Matlab en un computador, se muestra el desarrolló en base al procedimiento de cinco pasos descrito anteriormente. Al especificar el tamaño de paso h y el número de iteraciones, el programa

establece primero el potencial en todos los nodos iguales a cero. El potencial en el conductor exterior se fija igual a cero, mientras que en el conductor interior se establece en 100 voltios de modo que $vd = 100$. El programa encuentra C_0 cuando se retira el bloque dieléctrico y C cuando la losa está en su lugar y finalmente determinamos Z_0 . Para seleccionar h , el número de iteraciones debe ser lo suficientemente grande y mayor que el número de divisiones a lo largo de x o dirección y . La figura siguiente muestra algunos resultados típicos:

Impedancias Características		
h	Numero de Iteraciones	Zo
0.25	700	49.05
0.1	500	58.07
0.05	500	65.82
0.05	700	63.1
0.05	100	61.53

Figura 11: Impedancias Características

V. PROGRAMAS

Los siguientes programas se adaptaron de los textos de Mathews-Kurtis [1] y Sadiku [3].

Programa 1

```

function U=laplace1(funcion1,funcion2,funcion3,funcion4,a,b,h,tol,max1)
% Datos
% funcion1,funcion2,funcion3,funcion4 son las funciones en el contorno,
% almacenadas como cadenas de caracteres
% a y b son los extremos superiores de los intervalos [0,a] y [0,b]
% h es el incremento
% tol es la tolerancia
% U es la matriz, aqui se almacena la solución numérica
% Inicialización de los parámetros y de U
n=fix(a/b)+1;
m=fix(b/h)+1;
ave=(a*(feval('funcion1',0)+feval('funcion2',0))...
+b*(feval('funcion3',0)+feval('funcion4',0)))/(2*a+2*b);
U=ave*ones(n,m)
% Condiciones de contorno
U(1,1:m)=feval('funcion3',0:h:(m-1)*h)';
U(n,1:m)=feval('funcion4',0:h:(m-1)*h)';
U(1:n,1)=feval('funcion1',0:h:(n-1)*h)';
U(1:n,m)=feval('funcion2',0:h:(n-1)*h)';
U(1,1)=(U(1,2)+U(2,1))/2;
U(1,m)=(U(1,m-1)+U(2,m))/2;
U(n,1)=(U(n-1,1)+U(n,2))/2;
U(n,m)=(U(n-1,m)+U(n,m-1))/2;
    
```

```

% Parámetro de sobrerrelajación
w=4/(2+sqrt(4-(cos(pi/(n-1))+cos(pi/(m-1)))^2));
% Mejora de las aproximaciones
err=1;
cnt=0;
while((err>tol)&&(cnt<=max1))
    err=0;
    for j=2:m-1
        for i=2:n-1
            relx=w*(U(i,j+1)+U(i,j-1)+U(i+1,j)+U(i-1,j)...
                -4*U(i,j))/4;
            U(i,j)=U(i,j)+relx;
            if (err<=abs(relx))
                err=abs(relx);
            end
        end
    end
    cnt=cnt+1;
end
U=flipud(U');
mesh(U);
xlabel('eje x');
ylabel('eje y');
zlabel('Potencial eléctrico');

```

Programa 2

```

% SOLUCION DE LA ECUACION DE POISSON POR METODO DE SOBRERELAJACION
% Vxx+Vyy=G
% USO DEL METODO DE SOBRERELAJACION
% Nx : Numero de intervalos a lo largo del eje x
% Ny : Numero de intervalos a lo largo del eje y
% AXB : DIMENSION DE LA REGION SOLUCION
% V(I,J) : POTENCIAL EN LOS PUNTOS DE RED (X,Y)=H*(I,J)
% DONDE I= 0,1,...,Nx, J=0,1,...,Ny
% H : Tamano de la malla

% ESPECIFICAR LAS CONDICIONES FRONTERA Y PARAMETROS NECESARIO

A=1;B=1;
V1=0;V2=10;V3=20;V4=-10;
NX=20; % 4 12 20
NY=NX;
H=A/NX;

% Configuración inicial de los valores fijos

for I=1:NX-1
    for J=1 : NY-1
        V(I+1,J+1)=(V1 + V2 + V3 + V4)/4.0;
    end
end

```

```

    end
end

% Establecer los potenciales en los nodos fijos

for I = 1:NX-1
    V(I+1,1)=V1;
    V(I+1,NY+1)=V3;
end

for J=1:NY-1
    V(1,J+1)=V4;
    V(NX+1,J+1)=V2;
end

V(1,1)=(V1+V4)/2.0;
V(NX+1,1)=(V1+V2)/2.0;
V(1,NY+1)=(V3+V4)/2.0;
V(NX+1,NY+1)=(V2+V3)/2.0;

% ENCONTRAR EL FACTOR OPTIMO DE SOBRE RELAJACION

T = cos(pi/NX) + cos(pi/NY);
W = (8 - sqrt(64-16*T^2))/(T^2);
disp(['SOR Factor Omega = ',num2str(W)])
W4=W/4;

% COMIENZA LA ITERACION
NCOUNT = 0;

loop=1;
while loop == 1;
    RMIN = 0;
    for I =1:NX-1
        X = H*I;
        for J = 1:NY-1
            Y = H*J;
            G = -36.0*pi*X*(Y-1.0);
            R = W4*(V(I+2,J+1) + V(I,J+1) + V(I+1,J+2) + V(I+1,J)
                - 4.0*V(I+1,J+1) - G*H*H);
            RMIN = RMIN + abs(R);
            V(I+1,J+1) = V(I+1,J+1) + R;
        end
    end

    RMIN = RMIN/(NX*NY);
    if (RMIN>=0.0001)
        NCOUNT = NCOUNT + 1;
        if (NCOUNT>100)
            loop = 0;
            disp ('LA SOLUCIÓN NO CONVERGE DESPUÉS DE 100 INTERACIONES')
        end
    end
end

```

```

        end
    else
        % CUANDO RMIN SEA MENOR QUE 0.0001 Y LA SOLUCION ESTE CONVERGIENDO
        loop = 0;
        disp(['LA SOLUCION CONVERGE EN ',num2str(NCOUNT), 'iteraciones'])
        disp(['h = ',num2str(H)])
    end
end
end

Vnum = V;

% GRABACION DE LOS PUNTOS ORIGINALES A TRAVES DE i

    abc = zeros(1,9);
    a_tic = 1;
    vec = [0:H:1];
    for ii = .25:.25:.75
        for jj = .25:.25:.75
            xind = find(vec==ii);
            yind = find(vec==jj);
            % disp([xind,yind])
            abc(a_tic) = Vnum(xind,yind);
            a_tic = a_tic + 1;
        end
    end
end

% SALIDA DE RESULTADOS APROXIMADOS POR DIFERENCIAS FINITAS

%-----

% CALCULO DE LA SOLUCION EXACTA
%
% SOLUCION MEDIANTE EXPANSION DE SERIES DE LA ECUACION DE POISSON CON
% CONDICIONES DE FRONTERA HOMOGENEAS
%
for I = 1:NX-1
    X = H*I;
    for J = 1:NY-1
        Y = H*J;
        SUM = 0;
        for M = 1:10 % TOMANDO SOLAMENTE 10 TERMINOS DE LA SERIE
            FM = M;
            for N = 1:10
                FN = N;
                FACTOR1 = (FM*pi/A)^2 + (FN*pi/B)^2;
                FACTOR2 = ( (-1)^(M+N))*144*A*B/(pi*FM*FN);
                FACTOR3 = 1-(1-(-1)^N)/B;
                FACTOR = FACTOR2*FACTOR3/FACTOR1;
                SUM = SUM + FACTOR*sin(FM*pi*X/A)*sin(FN*pi*Y/B);
            end
        end
    end
end
end

```

```

VH = SUM;
C1=4*V1/pi;
C2=4*V2/pi;
C3=4*V3/pi;
C4=4*V4/pi;
SUM=0;
for K = 1:10 % TOMANDO LOS PROMEROS 10 TERMINOS DE LA SERIE
    N=2*K-1;
    AN=N;
    A1=sin(AN*pi*X/B);
    A2=sin(AN*pi*(A-Y)/B);
    A3=sinh(AN*pi*A/B);
    TERM1=C1*A1*A2/A3;
    B1=sinh(AN*pi*X/A);
    B2=sin(AN*pi*Y/A);
    B3=AN*sinh(AN*pi*B/A);
    TERM2=C2*B1*B2/B3;
    D1=sin(AN*pi*X/B);
    D2=sinh(AN*pi*Y/B);
    D3=AN*sinh(AN*pi*A/B);
    TERM3=C3*D1*D2/D3;
    E1=sinh(AN*pi*(B-X)/A);
    E2=sin(AN*pi*Y/A);
    E3=AN*sinh(AN*pi*B/A);
    TERM4=C4*E1*E2/E3;
    TERM = TERM1+TERM2+TERM3+TERM4;
    SUM=SUM + TERM;
end
VI = SUM;
Vexact(I+1,J+1) = VH + VI;
end
end

%Grabando los puntos originales a traves de i

abc2 = zeros(1,9);
a_tic = 1;
vec = [0:H:1];
for ii = .25:.25:.75
    for jj = .25:.25:.75
        xind = find(vec==ii);
        yind = find(vec==jj);
        % disp([xind,yind])
        abc2(a_tic) = Vexact(xind,yind);
        a_tic = a_tic + 1;
    end
end
end

figure(1),
imagesc(flipud(Vnum')),
colorbar

```

```

ylabel('y'), xlabel('x')
title('EJEMPLO DE LA ECUACION DE POISSON')

format short g
disp(' numerico exacto')
disp([abc' abc2'])

```

Programa 3

```

%*****
% Usando el método de diferencias finitas
% Este programa calcula la impedancia de características
% de una línea de transmisión
%*****

clear all; format compact;
% Salidas:
%
% H NT Z0
% -----
% 0.25 700 49.05
% 0.1 500 58.074
% 0.05 500 65.817
% 0.05 700 63.103
% 0.05 1000 61.53

H = 0.1;
NT = 500;

A = 2.5; B = 2.5; D = 0.5; W = 1.0;

ER = 2.35;
EO = 8.81E-12;
U = 3.0E+8;

NX = A/H;
NY = B/H;
ND = D/H;
NW = W/H;
VD = 100.0;

% Calcular la carga con y sin dieléctrico
ERR = 1.0;
for L= 1:2
    E1=EO;
    E2=EO*ERR;

% Inicialización
V = zeros(NX+2,NY+2);

```

```

% Establecer potencial en el conductor interno (nodos fijos) igual a vd
V(2:NW+2,ND+2) = VD;

% Calcular potencial en los nodos libres
P1= E1/(2*(E1+E2));
P2= E2/(2*(E1+E2));
for K=1:NT
    for I=0: NX-1
        for J=0:NY-1
            if ((J==ND)&&(I<=NW))
% no hacer nada
                elseif(J==ND)

% Imponer condiciones de contorno en la interfaz
                V(I+2,J+2) = 0.25*(V(I+3,J+2) + V(I+1,J+2)) +...
                    P1*V(I+2,J+3) + P2*V(I+2,J+1);

                elseif(I==0)
% Imponer condiciones de simetría a lo largo del eje y
                    V(I+2,J+2) = (2*V(I+3,J+2) + V(I+2,J+3) + V(I+2,J+1))/4.0;
                    elseif(J==0)
% Imponer condiciones de simetría a lo largo del eje X
                    V(I+2,J+2) = (V(I+3,J+2) + V(I+2,J+2) + 2*V(I+2,J+3))/4.0;
                    else
                    V(I+2,J+2) = (V(I+3,J+2) + V(I+1,J+2) + V(I+2,J+3) + V(I+2,J+1))/4.0;
                    end
                    end
                    end
                    end
% Animation of calculation
% figure(1), imagesc(v), colorbar,title([num2str(k),'/',num2str(NT)])
% drawnow

end

% Ahora calcula la carga total encerrada en A
% Trazado rectangular que rodea el conductor interno
    IOUT = round((NX+NW)/2);
    JOUT = round((NY+ND)/2);
% Potencial de suma en los bucles internos y externos
    for k= 1:2
        SUM= E1*sum(V(3:IOUT+1,JOUT+2))...
            + E1*V(2,JOUT+2)/2 + E2*V(IOUT+2,2)/2;
        for J=1:JOUT-1
            if( J<ND)
                SUM=SUM + E2*V(IOUT+2,J+2);
            elseif(J==ND)
                SUM=SUM +(E1+E2)*V(IOUT+2,J+2)/2;
            else
                SUM= SUM + E1*V(IOUT+2,J+2);
            end
        end
    end
end

```

```

if K==1
    SV(1) = SUM;
end
IOUT= IOUT-1;
JOUT= JOUT-1;
end
SUM=SUM + 2.0*E1*V(IOUT+2,JOUT+2);
SV(2)=SUM;
Q(L)= abs(SV(1)-SV(2));
ERR= ER;
end
% Finalmente calcular Z0
C0= 4.0*Q(1)/VD;
C1=4.0*Q(2)/VD;
Z0=1.0/(U*sqrt(C0*C1));

disp([H,NT,Z0])

```

REFERENCIAS

- [1] Kurtis D. Fink Jhon H. Mathews. *Metodos Numéricos usando Matlab*. Prentice Hall, 2008.
- [2] G. Lindfield and J. Penny. *Numerical Methods Using Matlab*. Prentice Hall, 2009.
- [3] Matthew N. O. Sadiku. *Numerical techniques in electromagnetics with Matlab*. CRC Press, 2009.
- [4] Raymond P. Canale Steven C. Chapra. *Métodos numéricos para ingenieros*. McGraw-Hill, 2009.