

Software Graficador de Potencial en *App Designer* mediante la aplicación de la ecuación de Laplace y su solución numérica

Bermeo Pablo¹, Calle David¹, Novillo Tyrone¹, Ochoa Stalyn¹, Plaza Lenin¹

Facultad de Ingeniería, Universidad de Cuenca, Cuenca, Azuay, Ecuador
<https://www.ucuenca.edu.ec/ingenieria>

Abstract. This report presents the developing of a grapher software which is able to show the distribution of Electric potential in a two-dimensional surface, by solving the Laplace equation with a discrete method. This project has been developed in *MatLab* and it's tool, *App Designer*. The final result allows to show the distribution of the electric potential produced by an specific geometric configuration: a square shell with a triangular equipotential surface in it's center. Also, the distribution and it's visualization is up on the resolution inserted in the software: 16, 32, 64, Finally, this program can be considered as a *multi-solver* of the Laplace equation because of it's capacity of being extrapolated to other areas.

Keywords: Electric Field · Field Lines · Equipotential surfaces · App Designer · Matlab

1. Introducción

Para calcular la disposición del potencial al momento que se trata de un problema bidimensional (varia en dos coordenada a la vez), se tienen dos métodos para su cálculo: una solución aplicando la ecuación de Laplace y aplicarla de forma analítica y una solución a la ecuación mediante la aplicación de métodos numéricos que es el tema a tratar.

La aplicación de la ecuación de Laplace de forma analítica obviamente se obtendrá una respuesta exacta o verdadera mente próxima a la solución real con un error muy bajo, pero lleva consigo varias dificultades para su aplicación puesto que si la región a trabajar presenta formas irregulares se complica en gran medida su cálculo. Siendo así la aplicación de métodos numéricos una alternativa viable.

La conveniencia de la aplicación de la solución numérica de este método interactivo es que no cuenta con mayor complejidad en el cálculo y que no se complica el cálculo dependiendo de la forma de la región a trabajar. Contrastando con esta idea se tiene como un posible "desventaja" el tiempo computacional en realizar las iteraciones previstas.

Comúnmente al momento de resolver este tipo de problemas donde se trabaja con distribuciones de potenciales se obvia la forma y la distribución que va a tener

este potencial en el espacio puesto que no se cuentan con herramientas didácticas que apoyen con la visualización de como está distribuido el potencial.

A consideración de la necesidad de visualización de como está distribuido el potencial se desarrolló un software graficador que se basa en un métodos numéricos que da una solución numérica a la ecuación de Laplace con el fin de facilitar el análisis de como está distribuido el potencial en el espacio.

2. Marco Teórico

2.1. Solución analítica de la ecuación de Laplace

Al tratarse de un problema bidimensional se hace uso de (1), al ser este un método analítico se tiene que trabajar algebraicamente sobre (1) y resolver esa ecuación parcial de segundo orden.

$$\nabla^2 V = 0$$

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \quad (1)$$

Luego de trabajar sobre la ecuación inicial queda (2).

$$V = XY = (A \cosh \alpha x + B \sinh \alpha x)(C \cos \alpha y + D \sin \alpha y) \quad (2)$$

Dependiendo de la configuración con la que se vaya a trabajar se calcularan las constantes A , B , C y D , mediante condiciones de frontera.

2.2. Solución a la ecuación de Laplace por métodos numéricos (solución numérica)

Al momento de aplicar la solución numérica de la ecuación de Laplace se la hace por un método iterativo y se parte de (3).

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \quad (3)$$

Estas derivadas parciales se aproximan de la siguiente manera:

En (4) se aproxima para la coordenada x :

$$\left[\frac{\partial^2 V}{\partial x^2} \right]_0 = \frac{\left[\frac{\partial V}{\partial x} \right]_a - \left[\frac{\partial V}{\partial x} \right]_c}{h} = \frac{V_1 - 2V_o + V_3}{h^2} \quad (4)$$

En (5) se aproxima para la coordenada y :

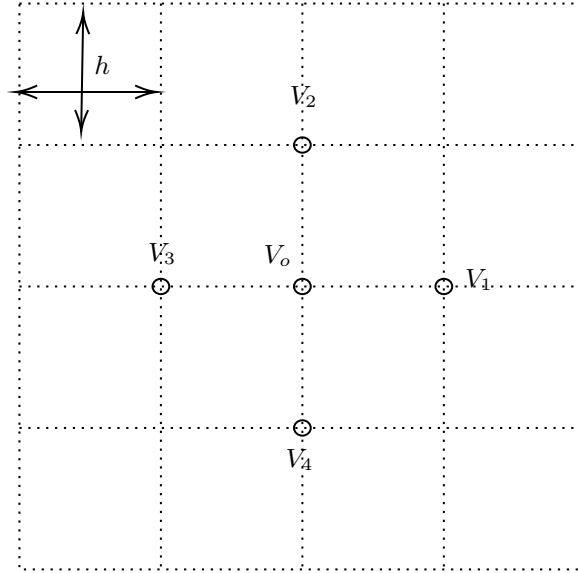
$$\left[\frac{\partial^2 V}{\partial y^2} \right]_0 = \frac{\left[\frac{\partial V}{\partial y} \right]_b - \left[\frac{\partial V}{\partial y} \right]_d}{h} = \frac{V_2 - 2V_o + V_4}{h^2} \quad (5)$$

Al igualar estas dos expresiones se tiene (6):

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = \frac{V_1 + V_2 + V_3 + V_4 - 4V_0}{h^2} = 0$$

$$V_o = \frac{1}{4}(V_1 + V_2 + V_3 + V_4) \quad (6)$$

Siendo V_o el potencial que se desea calcular y los diferentes V_i las tensiones hacia arriba, abajo, izquierda y derecha de V_o . Se explica de una mejor manera en el siguiente diagrama:



3. Desarrollo

La geometría propuesta para este trabajo está compuesta por una zona triangular de voltaje constante (25 V), dentro de un marco cuadrado. El marco se caracteriza por aislantes (0 V) que se disponen en los lados derecho, izquierdo e inferior, en tanto que el lado superior está sometido a 100 V. La disposición de esta geometría se encuentra en la Figura 1

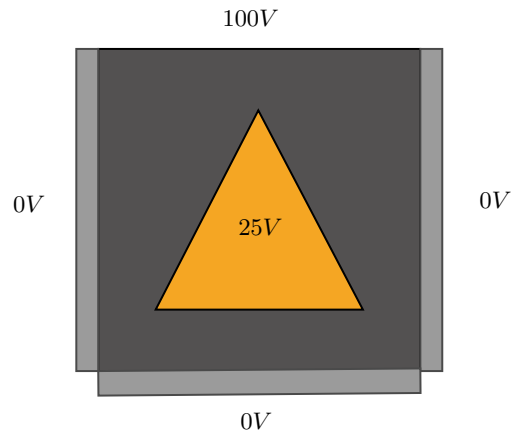


Figura 1. Geometría del problema propuesto

La resolución del problema está compuesta por dos partes: generación de la matriz de valores iniciales con una zona triangular interna con valores constantes y la aplicación del método numérico para la solución de la ecuación de Laplace.

3.1. Generación de la matriz de valores iniciales

Los valores iniciales en la matriz se han propuesto todos en $V_o = 0V$. Esto, a partir de la generación de una matriz de ceros de longitud n . Además, se ha determina que los valores de n están restringidos a múltiplos de 4. Por último, se deben añadir los valores iniciales en la primera fila de 100 V.

```
%Inserción del parámetro n
n = 1024;

%Generación de la matriz de ceros de dimensión nxn
MAT = zeros(n);

%Generar valores constantes de 100 V en la primera fila
MAT(1:1,1:n) = 100*ones(1,n);
```

Luego, se procede a generar la zona triangular con valores constantes de voltaje. Para la generación del mismo, se ha considerado un algoritmo iterativo que forma la zona triangular solo en el cuadrado central de la matriz. Mismo que tiene mediciones $n/2$.

El pseudocódigo con el que se entiende el algoritmo es el siguiente:

1. Posicionarse en las coordenadas $(n/4, \text{limite izquierdo})$ y $(n/4, \text{limite derecho})$
2. Luego, rellenar del valor constante V_c toda la fila junto con la inferior

3. ¿Se ha llegado a la última fila posible ($3(n/4)$)? Si, parar; no, repetir el proceso.
4. En caso de repetir el proceso. El límite izquierdo tiene un desplazamiento hacia la izquierda $li - 1$, y el límite derecho, a la derecha $ld + 1$

```

li = n/2; % Límite izquierdo de cada fila a rellenar
ld = n/2 + 1; % Límite derecho de cada fila a rellenar
i = 0;

while 1

    %Actualización de la fila
    fila = n/4 + ( 1 + 2*i );

    %Verificar que la fila no haya alcanzado su límite
    if ( fila == 3*(n/4) - 1 )
        break;
    end

    %Reemplazo de valores de voltaje constante
    MAT( fila:fila+1,li:ld ) = V_const * ones( 2 , abs(ld - li) + 1 );

    %Actualización de los límites laterales
    i = i + 1;
    li = li - 1;
    ld = ld + 1;
end

```

De esta forma se genera el triángulo para dimensiones de 16,32,64,128... Un ejemplo se muestra en la Figura 2.

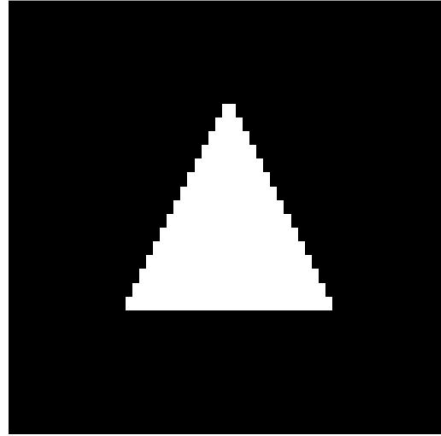


Figura 2. Generación del triángulo con $n = 64$

3.2. Solución de la ecuación de Laplace y mapa de calor

El proceso para la solución de la ecuación de Laplace requiere de una cuantificación del error, la solución aplicada a este problema, reside en la evaluación de la diferencia entre los valores máximos, provenientes de cada matriz generada en las iteraciones. La codificación de este proceso puede ser visto en las siguientes líneas.

```
error=max(max(abs(V_now-V_prev)));
```

Luego, el pseudocódigo que representa la aplicación sucesiva de la solución numérica de ecuación de Laplace es el siguiente:

1. aplicación del método para los valores que necesitan ser encontrados. Es decir, se ignoran los voltajes constantes.
2. Esto genera una matriz nueva de valores que será comparada con la anterior con el fin de cuantificar el error.
3. Si el error no ha sido alcanzado, entonces se vuelve al primer apartado.

La codificación de este proceso ha de ser visto en las siguientes líneas

```
while(error>0.01)%Ejecutar hasta alcanzar el umbral
```

```
    iter=iter+1; %Incrementar la iteración
```

```
    %Aplicación de las diferencias entre los puntos vecinos
```

```
    for i=2:1:xdim
```

```
        for j=2:1:ydim
```

```
            V_now(i,j)=(V_now(i-1,j)+V_now(i+1,j)
```

```

        +V_now(i,j-1)+V_now(i,j+1))/4;
    end
end
error=max(max(abs(V_now-V_prev))); %Calcula el error
V_prev=V_now; % Actualiza las matrices

%Cada iteración muestra la evolución del mapa de calor
imagesc(V_now);colorbar;
title(['Voltage distribution on a ',int2str(xdim),' x ',
int2str(ydim),' grid at iteration no ',int2str(iter)],'Color','k');
getframe;
end

```

4. Análisis de resultados

A continuación se presenta la distribución de potencial con $n = 16$. En esta configuración, las fronteras entre los diferentes valores de voltaje son bruscas y en forma de mosaico. Esto, claramente, se debe a la cantidad mínima de valores que conforman la matriz.

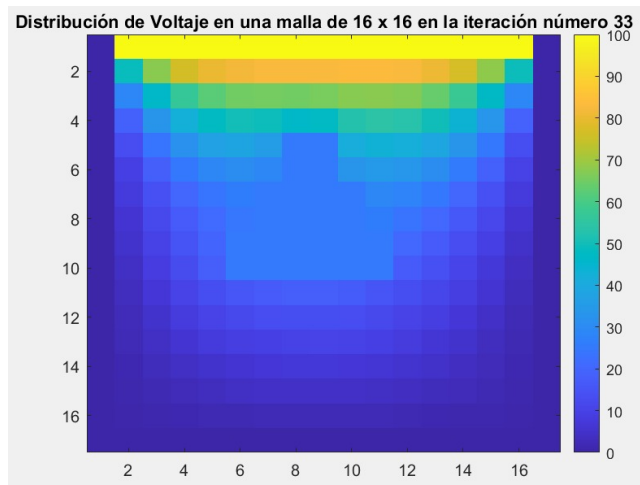


Figura 3. Generación del mapa de calor con $n = 16$

En la ejecución de una distribución con $n = 64$, se aprecia una cierta tendencia a la suavidad en la distribución del potencial. Es decir, las fronteras entre los distintos valores se difuminan y generan una imagen mas nítida. Sin embargo, los bordes aún presentan una diferencia observable.

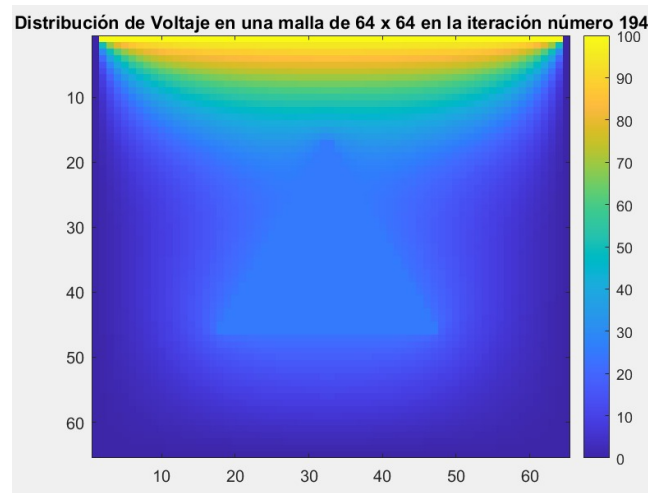


Figura 4. Generación del triángulo con $n = 64$

Para un valor de 128, las fronteras entre los voltajes son suaves y aún mas definidas. Además la figura geométrica central pierde opacidad. Luego, los valores de voltaje presentan una mayor distinción y la gama de colores empleada es aún más esclarecedora.

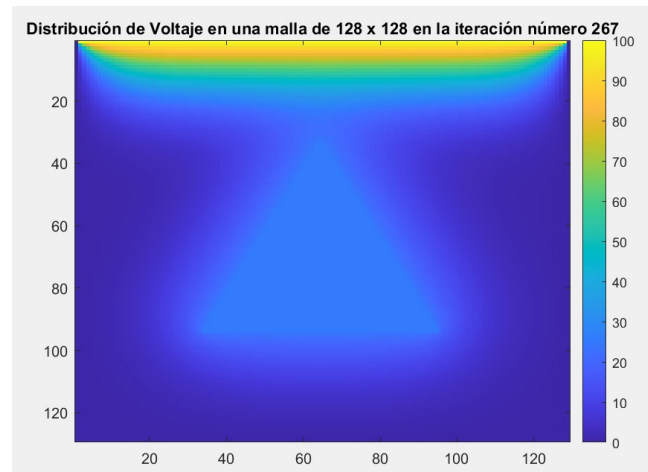


Figura 5. Generación del triángulo con $n = 128$

4.1. Desarrollo de la GUI

El desarrollo de la *Graphical User Interface* se realizó en la herramienta de *MatLab, App Designer*. Esta aplicación permitió la generación de ventanas y la matriz de valores de voltaje, matriz que se actualizaba constantemente. Esta es la interfaz que permite el uso del software:

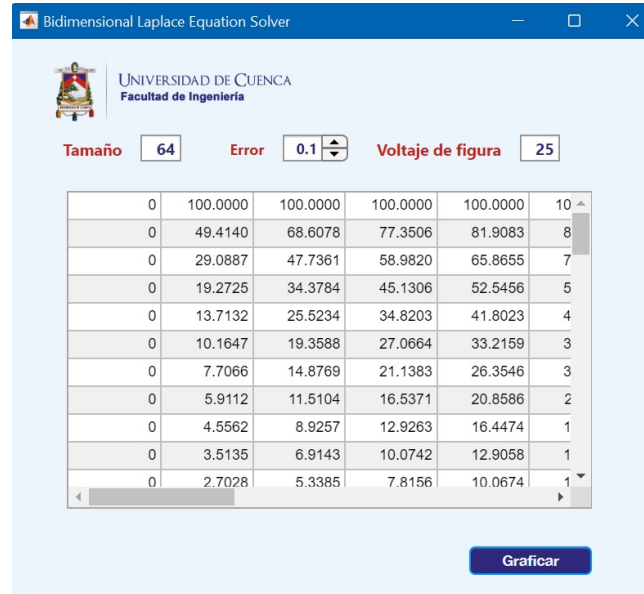


Figura 6. Interfaz para la interacción con el usuario.

5. Conclusiones

Las disposiciones geométricas formadas dentro de la matriz han de ser alcanzadas a través de algoritmos iterativos que consideran las superficies de las mismas figuras. En este caso, el triángulo requirió de algoritmo dinámicos en que la principal evolución estuvo centrada en los valores li y ld . El proceso para otras figuras geométricas retomaría distintas distribuciones para el algoritmo iterativo. Por ejemplo, un cuadrado centrado sería un proceso trivial al tener que seleccionar límites estáticos en función de las dimensiones del cascarón cuadrado.

Las gráficas en el mapa de calor muestran que un error suficiente para alcanzar un resultado aceptable es de 0.01. Este valor de error genera una cantidad de iteraciones mayor a 100, lo cual asegura una visualización significativa de los valores de la matriz en el mapa (*id*, *est.*, la gama de colores permite reconocer las fronteras entre voltajes distintos). En cuanto a la dimensión adecuada, se asegura que una resolución (o valor de n) mayor o igual a 128 muestra unas fronteras significativas entre los valores de voltaje.

En cuanto a la creación de la GUI (*Graphical User Interface*), se recomienda el uso de la herramienta, propia de *MatLab*, *App Designer*. Esta herramienta provee de comodidad a la hora de establecer una ventana en la que se visualiza algún resultado proveniente del código, en este caso, el mapa de calor y la tabla de valores para la matriz.

El algoritmo generado en este proyecto tiene la capacidad de ser empleado en las distintas aplicaciones de la Ecuación de Laplace. Las condiciones para su uso residen, únicamente, en la naturaleza de las condiciones iniciales.

Por último, la solución de Ecuaciones diferenciales parciales (PDEs), se facilita a través del método de diferencias finitas. Este último método genera la fórmula iterativa usada en el algoritmo para calcular los voltajes en distintos puntos. Cabe recalcar que este método proviene de una discretización de la ecuación de Laplace.