

CC3501 – Modelación y Computación Gráfica para Ingenieros

# Tarea N°1

Alumna: María José Trujillo Berger

Profesora: María Cecilia Rivara

Auxiliares: Cristóbal Muñoz

Sergio Leiva

Fecha de Entrega: 2 de septiembre de 2017

## Problema 1:

Una placa de níquel con forma de L, cuyas dimensiones están en relación 55:89 [alto máximo: largo máximo], la cual posee un material aislante en dos de sus lados, es calentada con  $Q=200$  [kJ] de forma constante. El problema puede ser modelado con la ecuación de Poisson  $u_{xx} + u_{yy} = -\frac{Q}{k}$  Siendo  $k= 444 \left[ \frac{J}{kg \cdot K} \right]$ , el calor específico del material. La placa presenta las siguientes condiciones de borde:

### Tipo Dirichlet:

$$\begin{aligned}u(x,0)&=0 & 0 \leq x \leq 89 \\u(0,y)&=0 & 0 \leq y \leq 55 \\u(34,y)&=0 & 34 \leq y \leq 55 \\u(x,34)&=0 & 34 \leq x \leq 89\end{aligned}$$

### Tipo Neumann:

$$\begin{aligned}u_x(x, 55) &= 0 & y = 55 & 0 \leq x \leq 34 \\u_y(89, y) &= 0 & x = 89 & 0 \leq y \leq 34\end{aligned}$$

Dado que este problema está modelado por una ecuación diferencial parcial elíptica, se elige para su resolución usar el método iterativo de sobrerelajación sucesiva para diferencias finitas. Para ello, se intenta simular virtualmente la placa con una matriz rectangular de ceros, con dimensiones  $\frac{55}{h} \times \frac{89}{h}$  (filas x columnas) con  $h$  el salto utilizado para discretizar el problema.

Se identifican cinco dificultades principales al desarrollar el método: divisores en común de las proporciones, problema general, condiciones de borde tipo Neumann, iterar los puntos de la matriz que están fuera de la placa y luego quitar dichos puntos de la matriz final

### 1.- Divisores en común:

Es fácil notar que no hay divisores enteros en común entre 55 y 89 además del 1, por lo que no es posible utilizar pasos mayores que 1 con las dimensiones especificadas anteriormente. Para los pasos menores que 1, se pueden presentar “puntos perdidos” para algunos pasos en específico, por ejemplo  $89/0.5$  es 179, pero  $88/h$  es 176, por lo que si no es extremadamente cuidadoso al momento de iterar, pueden existir columnas que no sean iteradas. Para evitar estos inconvenientes y simplificar el problema de escoger “un paso adecuado” se agrega una función piso de manera que utilice el entero inferior al cociente entre las dimensiones de la placa y el paso escogido.

### Pseudocódigo:

```
x=fix(89/h); %piso para el largo de la placa
y=fix(55/h); %piso para el ancho de la placa
b=fix(34/h); %piso para definir la forma de L
z=fix(21/h); %piso para definir la forma de L
u= zeros((y),(x)); %Matriz con el piso incluido
```

### 2.-Problema general:

Luego de definir las variables tipo Neumann en el programa, se realizan las iteraciones sobre la matriz utilizando dos ciclos *for* anidados entre si, que recorren todos los puntos de la matriz comprendidos entre 2 y el largo de la fila o columna menos 1 (para quitar los bordes), realizando la iteración y redefiniendo los puntos de la matriz:

$$r = (u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) - 4 * u(i,j) - (h^2 * g))/4 \quad (1)$$

$$u(i,j) = u(i,j) + w * (u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) - 4 * u(i,j) - (h^2 * g))/4 \quad (2)$$

Con  $w = \frac{4}{2 + \sqrt{4 - (\cos(\frac{\pi}{55/h-1}) + \cos(\frac{\pi}{89/h-1}))^2}}$  factor de relajación.

Esta iteración está anidada dentro de un ciclo *while* que existe mientras r sea mayor que un error máximo determinado como  $e = Q/k/100000 = 0.0045$ . Este error se determinó buscando el error más pequeño para el cual la función se ejecutaba en un tiempo razonable (menos de 30 segundos para paso 0.5). Además, se incluye un límite de iteraciones, de manera que el programa se detenga luego de realizadas 300 iteraciones, para evitar que en casos divergentes la función no sea capaz de detenerse por si sola con el uso del error.

### 3.- Condiciones tipo Neumann:

Es necesario notar que, además de necesitar una ecuación especial dada su condición de derivadas, las filas y columnas extremas de la matriz están fuera de la iteración por lo que nunca se llegará a ellas en la ecuación principal. Para solucionar este problema, en la misma estructura de la parte 2 se deben agregar las siguientes fórmulas para cada punto que cumpla con la condición tipo Neumann, especificando el índice del mismo:

$$u(1,j) = u(1,j) + w * ((u(1,j-1) + u(1,j+1) + (2 * u(2,j)) - 4 * u(1,j) - (h^2 * g))/4) \text{ para el lado BC de la placa. } (3)$$

$$u(i,89/h) = u(i,89/h) + w * ((u(i+1,89/h) + u(i-1,89/h) + (2 * u(i,(89/h) - 1)) - 4 * u(i,89/h) - (h^2 * g))/4) (4)$$

Para el lado EF de la placa.

Nota: el punto 89/h corresponde en realidad al punto  $x = \text{fix}(89/h)$ , no incluido por claridad.

#### 4.- Iterar puntos fuera de la placa:

Cómo existen puntos de la matriz que no existen en la placa, simplemente se utiliza el comando *continue* para que el programa no realice la iteración sobre dichos puntos.

##### Pseudocódigo para puntos 2, 3,4 :

```
e=1000; %Se elige un valor grande para poder entrar al while
emax= 0.0005; %Error muy pequeño para poder finalizar la iteración
iter=0; %Inicializa un contador
while e>emax && iter<=300
    e = 0; %resetear error
    for i=2:(totalFilas-1); %for anidados
        for j= 2:(TotalColumnas -1); recorren los puntos interiores de la matriz

            %Parte inexistente
            if i<(21/h)&& j>(34/h)
                continue

            %Condiciones tipo Dirichlet
        else
            Ecuación (1)
            Ecuación (2)
            %Condiciones tipo Neumann
            Ecuación (2)
            Ecuación (3)
```

Nota: Ecuaciones y “end” omitidos por claridad y estética. 21/h 34/h corresponden a los puntos ya corregidos con la función fix, no incluidos por claridad.

#### 5.- Quitar los puntos fuera de la placa:

Luego de que la iteración sale del ciclo *while* se deben redefinir los puntos que no existen en la placa para que no salgan representados en el gráfico, para ello a cada punto que esté fuera de la placa debe asignársele el valor *NaN*, lo cual se realiza con el siguiente *ciclo for*

##### Pseudocódigo para punto 4:

```
for i= 1: z-1
    for j= b-1 % Puntos fuera de la placa
        u(i,j)=NaN; %Asignar valor NaN
```

Se eligió un método iterativo sobre uno directo ya que, dado el gran tamaño de la matriz, el proceso de este último se vuelve muy engorroso y puede dar pie a errores. Al mismo tiempo, se presenta la ventaja de ser menos costoso computacionalmente en comparación a invertir una matriz de semejante tamaño como en el método directo.

Las desventajas, por otro lado, del método de sobrerelajación sucesiva en comparación al método de Newton, es que esta entrega como resultado una aproximación estrictamente dependiente de los parámetros admitidos, tales como  $r$  y  $w$  (mencionados anteriormente) y depende de la estabilidad del problema ya que la iteración podría divergir y nunca arrojar un resultado final. En cambio, el método directo entrega un resultado exacto.

### Resultados:

Luego de ejecutar el programa se obtienen los siguientes resultados, estos corresponden a las representaciones 2D, 3D y curvas de nivel de la placa calentada:

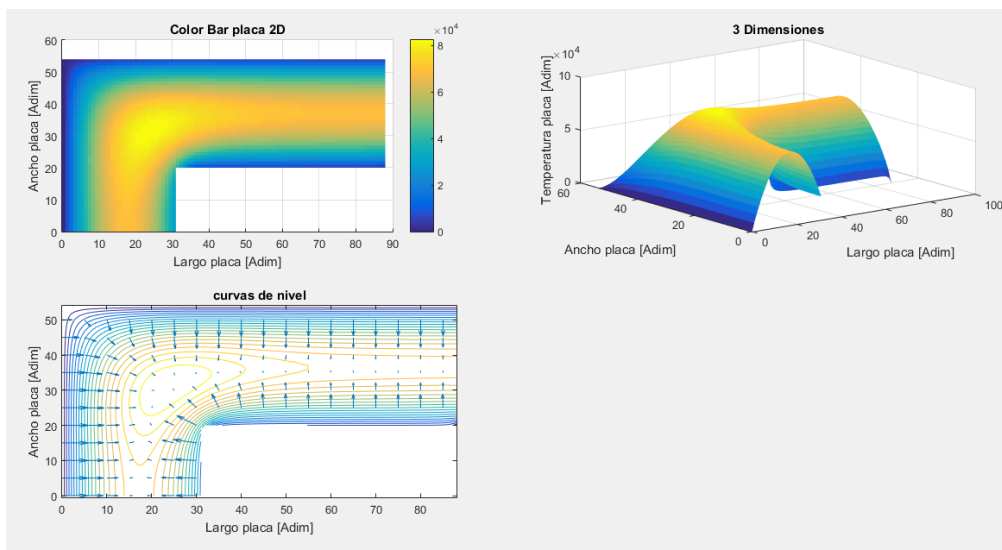


Gráfico 1: Barra de color, visualización 3D y curvas de nivel de la placa,  $h=1$

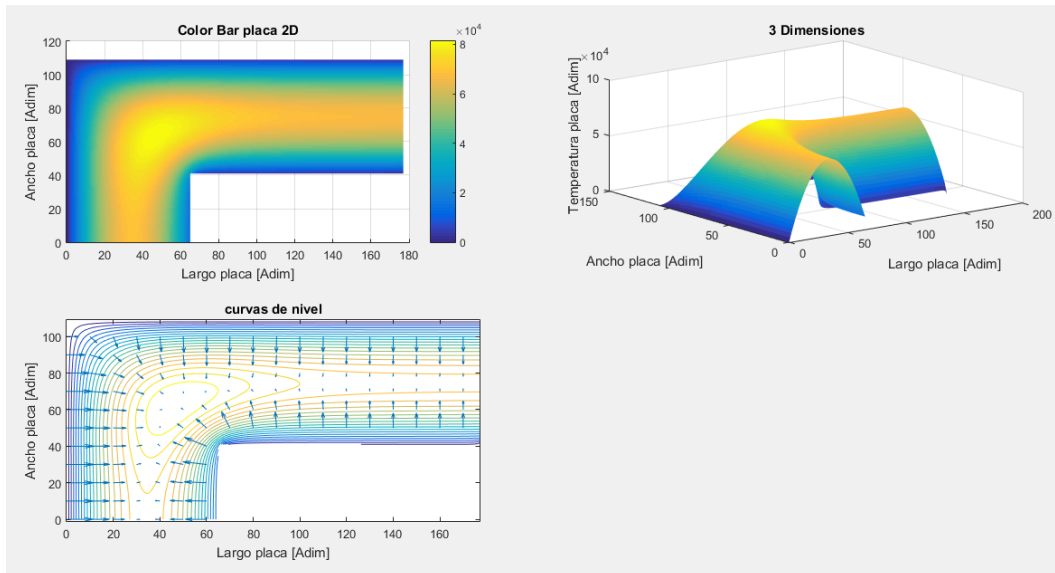


Gráfico 2: Barra de color, visualización 3D y curvas de nivel de la placa,  $h=0.5$

Se observa que en los sectores donde las condiciones de borde son cero, no hay flujo de calor (borde azul) a diferencia de aquellas donde se presenta el aislante, el cual dada su naturaleza, no permite que el calor se disipe (borde naranja). Es consistente que las zonas donde las condiciones de borde son distintas de cero se concentre una gran cantidad de calor debido a que el Níquel es un material de alta conductividad térmica.

Si suponemos ahora que la placa está hecha de agua, se obtienen los siguientes resultados:

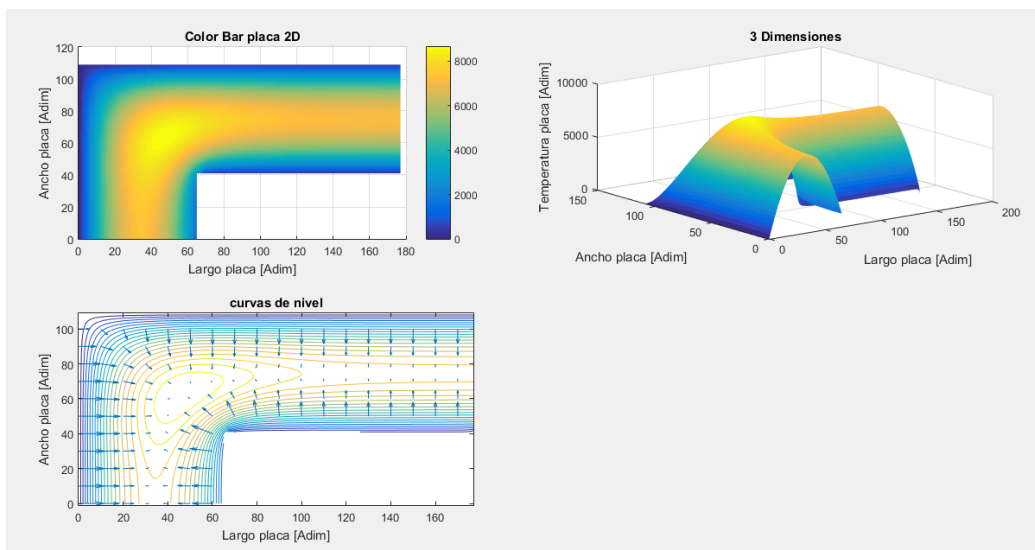


Gráfico 3: Barra de color, visualización 3D y curvas de nivel de la placa de agua  $h=0,5$

Se observa que esta vez, la temperatura por la placa es mucho menor a cuando la placa es de níquel, esto debido a que el calor específico del agua  $k=4181.4 \left[ \frac{\text{J}}{\text{kg} \cdot \text{K}} \right]$

es significativamente mayor. Esto implica que requiere recibir más energía para elevar en un grado su temperatura que la placa de níquel.

Al contrario, al probar con un metal cuyo valor específico es inferior al del níquel, como lo es el oro con  $k=129 \left[ \frac{\text{J}}{\text{kg} \cdot \text{K}} \right]$  se observa que la temperatura aumenta exponencialmente. (En particular para el oro, sube de un orden  $10^4$  (Níquel) a  $10^5$ )

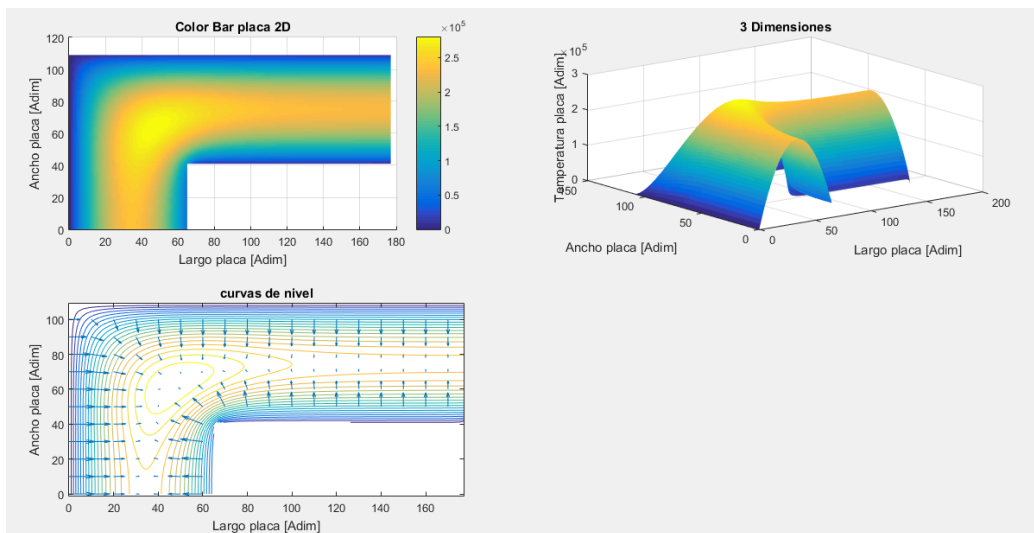


Gráfico 4: Barra de color, visualización 3D y curvas de nivel de la placa de oro  $h=0,5$

Nota: los gráficos han sido adjuntados en la carpeta de la tarea, en caso de necesitar una mejor resolución.

## Problema 2:

Se tiene una cuerda de largo  $3L$  (pudiendo ser  $L$  definida como cualquier valor, de modo que se obtenga una solución general de la ecuación) que oscila en el tiempo, sujeta por ambos extremos. Se busca resolver la ecuación de onda

$$\frac{\partial^2 \psi(x,t)}{\partial t^2} = \frac{c^2 \partial^2 \psi(x,t)}{\partial x^2}.$$

Con condiciones iniciales:

$$\psi(x, 0) = 3 * n * \sin \pi x$$

$$\psi_t(x, 0) = 0$$

Donde  $n$  corresponde al número de lista del alumno, en este caso correspondiente a 57

Dado que está sujeta por ambos extremos se imponen las siguientes condiciones de borde:

$$\psi(0, t) = 0$$

$$\psi(3L, t) = 0$$

Esta ecuación de onda corresponde a una ecuación diferencial parcial hiperbólica, y para solucionarla se realizará, nuevamente, un método iterativo. Este se aplicará sobre una matriz que busca simular el movimiento de la cuerda en el tiempo, siendo de la forma tiempo (filas) x largo de la cuerda (columna), parámetros que se deben indicar al inicio del programa.

Antes de comenzar a iterar se deben definir los saltos para cada variable de la ecuación. Dado que  $c = 2$  se elige  $h = 0.1$  y  $k = 0.05$ .  $K$  se elige en función de  $c$  y  $h$  buscando que  $r = \frac{c*k}{h} = 1$  para mantener la estabilidad del método.

Podemos observar dos dificultades principales en el desarrollo del método iterativo: Valores de la segunda fila y el problema general:

### 1.- Valores segunda fila

Se necesitan como mínimo los valores de dos filas consecutivas para poder realizar el método iterativo con la fórmula que este tiene, si bien existen varias formas de aproximar los valores de la segunda fila de la matriz, dado que  $r=1$ , y  $\psi_t(x, 0) = 0$ , una aproximación razonable es copiar los valores de la condición de borde que están en la primera fila, en la segunda como muestra el siguiente Pseudocódigo:



```

for j= 1:largo cuerda
    Cin=3*n*sin(pi*(0.1*j));%Condición de borde
    %Se multiplica j por 0.1 para que no quede sin(0)
    A(1,j)=Cin;
    A(2,j)= Cin;

```

2.- Problema general: Debido a que no hay más singularidades, se puede ejecutar el método en un solo bloque de código compuesto por dos ciclos *for* anidados que recorren todos los puntos de la matriz realizando la iteración:

$$A(i + 1, j) = A(i, j + 1) + A(i, j - 1) - A(i - 1, j)$$

Pseudocódigo:

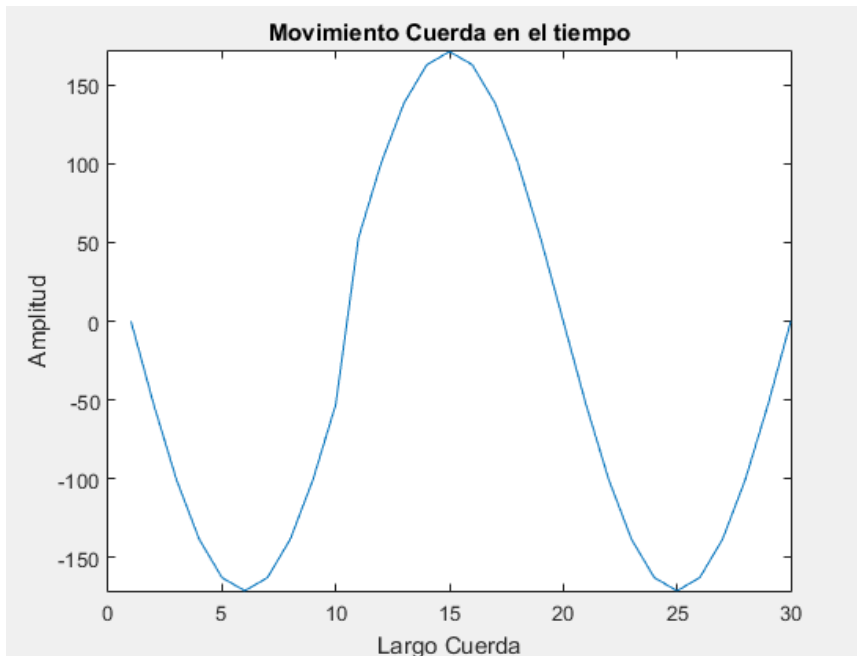
```

for i= 2:((tiempo/pasotiempo))-1 %Recorrer las filas sin sus extremos
    for j= 2:(3*L/pasolargo)-1    %Recorrer las columnas sin sus extremos
        A(i+1,j)=A(i,j+1)+A(i,j-1)-A(i-1,j); %Aplicar el método
    end
end

```

Resultados:

Luego de la iteración se obtienen los siguientes resultados

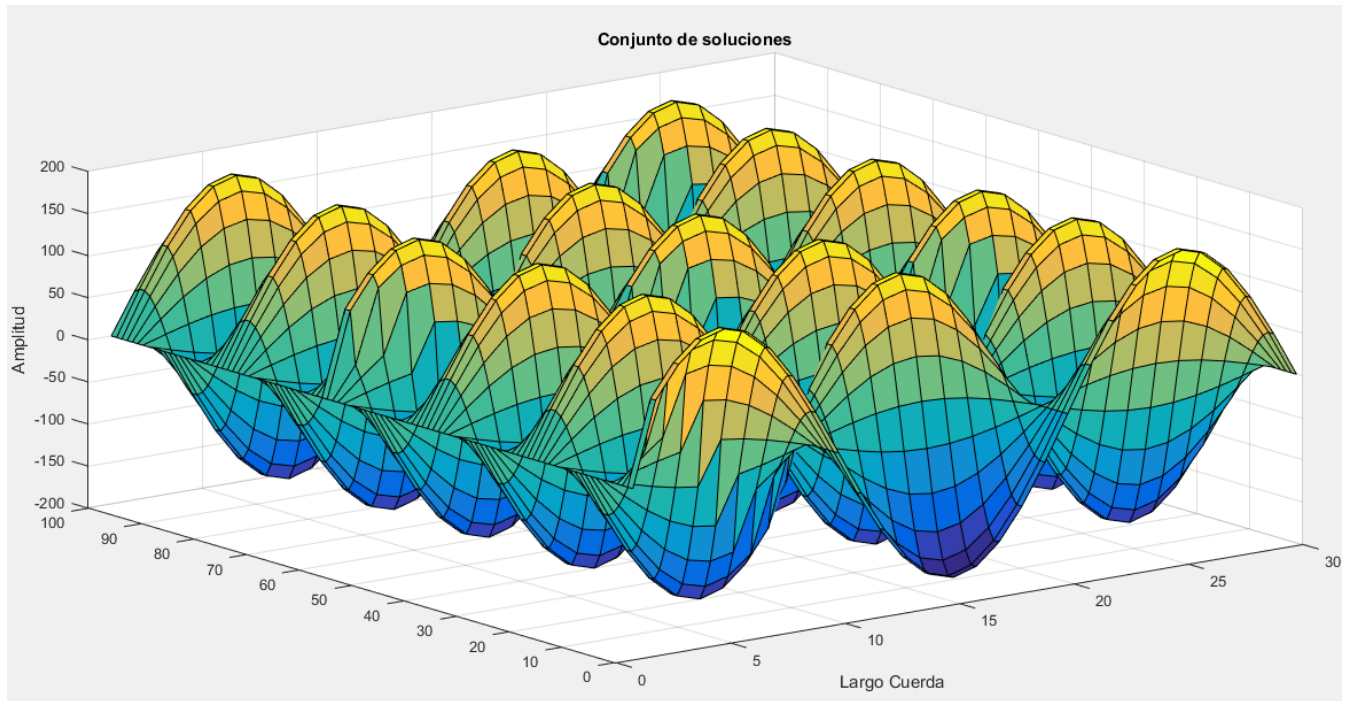


*Gráfico 2: Movimiento de la cuerda en el tiempo.*

Se observan sectores “quebrados” en el movimiento de la cuerda y esto puede deberse netamente a un error en la aproximación en la segunda fila, ya que es más riguroso utilizar una expansión de Taylor (cuyo valor también contiene un determinado error) a utilizar los valores de la primera fila como aproximación razonable, sin embargo, no es de una

importancia drástica para el resultado general. Estos sectores también pueden ser producto de la discretización del problema, causando sectores “anómalos” en los puntos donde su error es mayor.

Se puede graficar también, el conjunto solución de la ecuación, arrojando el siguiente resultado:



*Conjunto solución del movimiento de la cuerda*

Se elige un método implícito sobre uno explícito ya que, al igual que en la pregunta anterior, se busca un método que no requiera cálculos engorrosos debido al gran tamaño que puede llegar a tener la matriz trabajada. Bajo esta perspectiva, al usar este método se encuentra la desventaja de la posible inestabilidad que pueda tener, sin embargo, eso se soluciona imponiendo  $r \leq 1$ . Si bien el método no es tan estable como un método implícito, es una aproximación lo suficientemente razonable como para que sustituya el gasto computacional que implica resolver todas las ecuaciones de este último.

## Conclusiones Generales

A pesar de las diferencias entre ambos problemas, uno correspondiente a una EDP Elíptica y el otro a una EDP hiperbólica, se concluye que un método iterativo es más conveniente que uno directo para darles solución.

El principal motivo para no elegir un método directo en estos problemas está ligado a la cantidad de operaciones que deben ser definidas que, en estos casos, daban pie a errores al momento de programar. Por ello, es preferible calcular una aproximación del problema, teniendo las consideraciones necesarias para que el error asociado no afecte los resultados de manera relevante.

Se observa, en el caso de la placa, consistencia entre la teoría y los resultados obtenidos luego de la iteración, ya que el calor se disipa en los bordes donde la placa tiene condiciones tipo Dirichlet y se acumula donde se encuentra el aislante térmico en las condiciones tipo Neumann (no puede salir).

En el caso de la cuerda, se observa un movimiento oscilatorio en el tiempo, el cual – dejando de lado algunas singularidades del gráfico- es similar al movimiento de una cuerda bajo un impulso.

Por lo tanto, ambos métodos resuelven eficientemente los problemas.