Proyecto final Análisis Numérico

Grupo E

Mayo de 2024

1. Introducción

En este documento se solucionan las ecuaciones diferenciales que describen el comportamiento de difusión de un disipador de calor simulando diferentes estados y condiciones iniciales. Se muestran los resultados obtenidos de simulaciones realizadas en una aplicación gráfica de MATLAB.

2. Planteamiento del problema

Un disipador de calor es un componente electrónico que se utiliza para eliminar el exceso de calor generado por los circuitos electrónicos. Está diseñado para ayudar a prevenir el sobrecalentamiento de los componentes, reduciendo así el riesgo de daños y fallas en los equipos.

Para este proyecto lo que realizaremos será modelar el comportamiento de la temperatura sobre la superficie del disipador teniendo en cuenta diferentes condiciones iniciales.

- **Estado estacionario:** La ecuación para el estado estacionario es:

$$\frac{\partial^2 T}{\partial r^2} - \alpha^2 T = 0 \tag{1}$$

- **Estado transitorio:** La ecuación para el estado transitorio es:

$$\frac{\partial^2 T}{\partial r^2} - \alpha^2 T = \frac{\partial T}{\partial t} \tag{2}$$

  dependientes de:

  - Longitud de la superficie de difusión.

- Tiempo total de la simulación.
- Número de puntos en la superficie.
- Número de instantes de tiempo.
- Constante de material.
- Temperatura del contorno (inicial y final).
- Temperatura inicial.

Además existen estos tres casos de condiciones de contorno:

1. Temperatura fija del primer y el último nodo.
2. Temperatura fija del primer nodo y derivada igualada a cero del último nodo.
3. Derivada igualada a cero del primer nodo y temperatura fija del último nodo.

3. Estado estacionario

Para empezar, tengamos presente que la ecuación utilizada para modelar el estado estacionario del disipador es:

$$\frac{\partial^2 T}{\partial x^2} - \alpha^2 T = 0.$$

Para resolver numéricamente esta ecuación diferencial ordinaria de segundo orden, utilizamos el método de diferencias finitas. La ecuación presenta una única variable independiente, que es la superficie. Además, α es una constante del material cuyo valor estará definido, al igual que la temperatura del contorno y la longitud de la superficie de difusión.

Mediante el método de diferencias finitas, se formulan ecuaciones en cada nodo de la superficie donde no conozcamos su temperatura para así poder calcularla. Este método sugiere la siguiente aproximación:

$$\left(\frac{\partial^2 T}{\partial x^2}\right)_i \approx \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2},$$

donde i indica el número del nodo en la superficie.

De este modo, para cada nodo en la superficie se obtiene la siguiente ecuación:

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} - \alpha^2 T_i = 0.$$

Para simplificar su implementación, procedemos de la siguiente manera:

$$\begin{aligned} T_{i+1} - 2T_i + T_{i-1} - \Delta x^2 \alpha^2 T_i &= 0 \\ T_{i+1} + T_{i-1} - (2 + \Delta x^2 \alpha^2) T_i &= 0 \end{aligned}$$

Así, llegamos al siguiente sistema de ecuaciones para n nodos:

$$\begin{aligned} T_2 + T_0 - (2 + \Delta x^2 \alpha^2) T_1 &= 0 \\ T_3 + T_1 - (2 + \Delta x^2 \alpha^2) T_2 &= 0 \\ &\vdots \\ T_{n+1} + T_{n-1} - (2 + \Delta x^2 \alpha^2) T_n &= 0. \end{aligned}$$

Para este punto, nos encontramos con un sistema de n ecuaciones con $n + 2$ variables. Para obtener un sistema resoluble, debemos eliminar variables haciendo uso de las condiciones iniciales. En en nodo inicial y final hay dos opciones de condiciones de frontera:

1. **Temperatura fija de un nodo frontera.**
   En este caso, T_1 o T_n se vuelven valores conocidos y por lo tanto la ecuación para encontrar su valor no son necesarias. Además su valor se vuelve una constante para las ecuaciones que lo utlizan.

2. **Derivada igual a cero en un nodo frontera.**
   Si el valor de la derivada en alguno de estos nodos es igual a cero podemos asumir que nos encontramos frente a un máximo o mínimo local. Ya que la idea es que la distancia entre los nodos sea pequeña (δx pequeño), los valores del nodo anterior y posterior podemos asumir que son iguales ($T_{n-1} = T_{n+1}$). Para comprobar ello podemos ver que en la siguiente función al escoger puntos que tienen la misma distancia desde el máximo local, tienen valores similares. Claramente entre menor sea δx, más precisa será esta aproximación.

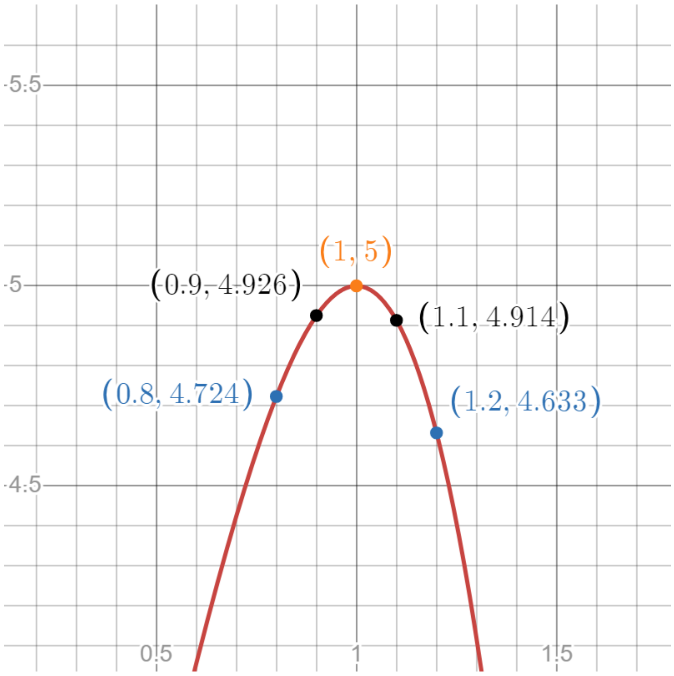


Figura 1: Función $y = -(x - 0{,}4142)^4 + 4(x - 0{,}4142)^2 + 1$,evaluada cerca de su máximo local (1,5).

Con estos conceptos claros, construiremos el sistema de ecuaciones que resolveremos en cada uno de los tres posibles casos de la modelación.

- **Temperatura fija del primer y el último nodo.**
  Como en esta ocasión ya conocemos la temperatura en ambos nodos, no tenemos que resolver esas temperaturas en el modelo, así que eliminamos la primera y la última ecuación del modelo propuesto anteriormente. De esta manera, nos quedamos con estas $n-2$ ecuaciones.

$$T_3 + T_1 - (2 + \Delta x^2 \alpha^2) T_2 = 0$$
$$\vdots$$
$$T_n + T_{n-2} - (2 + \Delta x^2 \alpha^2) T_{n-1} = 0.$$

  Aun así, seguimos teniendo n variables para las $n-2$ ecuaciones; sin embargo, ya que T_1 y T_n son constantes sólo buscaremos el valor de T_2 hasta T_{n-1}. Luego podemos reescribir el sistema de la siguiente manera:

$$T_3 - (2 + \Delta x^2 \alpha^2) T_2 = -T_1$$
$$\vdots$$
$$T_{n-2} - (2 + \Delta x^2 \alpha^2) T_{n-1} = -T_n.$$

En forma de matrices, este sistema quedaría representado de la siguiente manera:

$$\begin{pmatrix} -(2+\Delta x^2\alpha^2) & 1 & 0 & \cdots & 0 \\ 0 & -(2+\Delta x^2\alpha^2) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -(2+\Delta x^2\alpha^2) & 1 \end{pmatrix} \begin{pmatrix} T_2 \\ T_3 \\ \vdots \\ T_{n-1} \end{pmatrix} = \begin{pmatrix} -T_1 \\ 0 \\ \vdots \\ -T_n \end{pmatrix}$$

- **Temperatura fija del primer nodo y derivada igualada a cero del último nodo.**
  En este caso, al ser conocido el valor de la temperatura en el primer nodo omitimos su cálculo; sin embargo, sí debemos incluir la variable T_n en el sistema. Luego, estas serían las $n-1$ ecuaciones escogidas para el sistema:

$$\begin{aligned} T_3 + T_1 - (2+\Delta x^2\alpha^2)T_2 &= 0 \\ &\vdots \\ T_{n+1} + T_{n-1} - (2+\Delta x^2\alpha^2)T_n &= 0. \end{aligned}$$

  Como podemos notar en la última ecuación se necesita el valor de T_{n+1}, pero como la derivada es 0 para T_n podemos asumir que $T_{n+1} \approx T_{n-1}$, lo que nos da como resultado el siguiente sistema:

$$\begin{aligned} T_3 + T_1 - (2+\Delta x^2\alpha^2)T_2 &= 0 \\ &\vdots \\ 2T_{n-1} - (2+\Delta x^2\alpha^2)T_n &= 0. \end{aligned}$$

  El cual en forma matricial se vería así:

$$\begin{pmatrix} -(2+\Delta x^2\alpha^2) & 1 & 0 & \cdots & 0 & 0 \\ 0 & -(2+\Delta x^2\alpha^2) & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -(2+\Delta x^2\alpha^2) \end{pmatrix} \begin{pmatrix} T_2 \\ T_3 \\ \vdots \\ T_n \end{pmatrix} = \begin{pmatrix} -T_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- **Derivada igualada a cero del primer nodo y temperatura fija del último nodo** Estas condiciones son análogas al caso anterior: la temperatura T_1 sí la debemos calcular y para ello debemos sustituir T_0 por T_2 en la primera ecuación del sistema ya que podemos asumir que $T_0 \approx T_2$. Además, como el valor de T_n es conocido, su resta será utilizada para conformar el último valor del vector $\mathbf{b}$ usado en la resolución del sistema. De este modo así quedarían las sistemas y su expresión en matrices:

$$\begin{aligned} 2T_2 - (2+\Delta x^2\alpha^2)T_1 &= 0 \\ T_3 + T_1 - (2+\Delta x^2\alpha^2)T_2 &= 0 \\ &\vdots \\ 2T_{n-1} - (2+\Delta x^2\alpha^2)T_n &= 0. \end{aligned}$$

$$
\begin{pmatrix}
-(2+\Delta x^2\alpha^2) & 2 & 0 & \cdots & 0 & 0 \\
0 & -(2+\Delta x^2\alpha^2) & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & -(2+\Delta x^2\alpha^2)
\end{pmatrix}
\begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_{n-1} \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ \vdots \\ -T_n \end{pmatrix}
$$

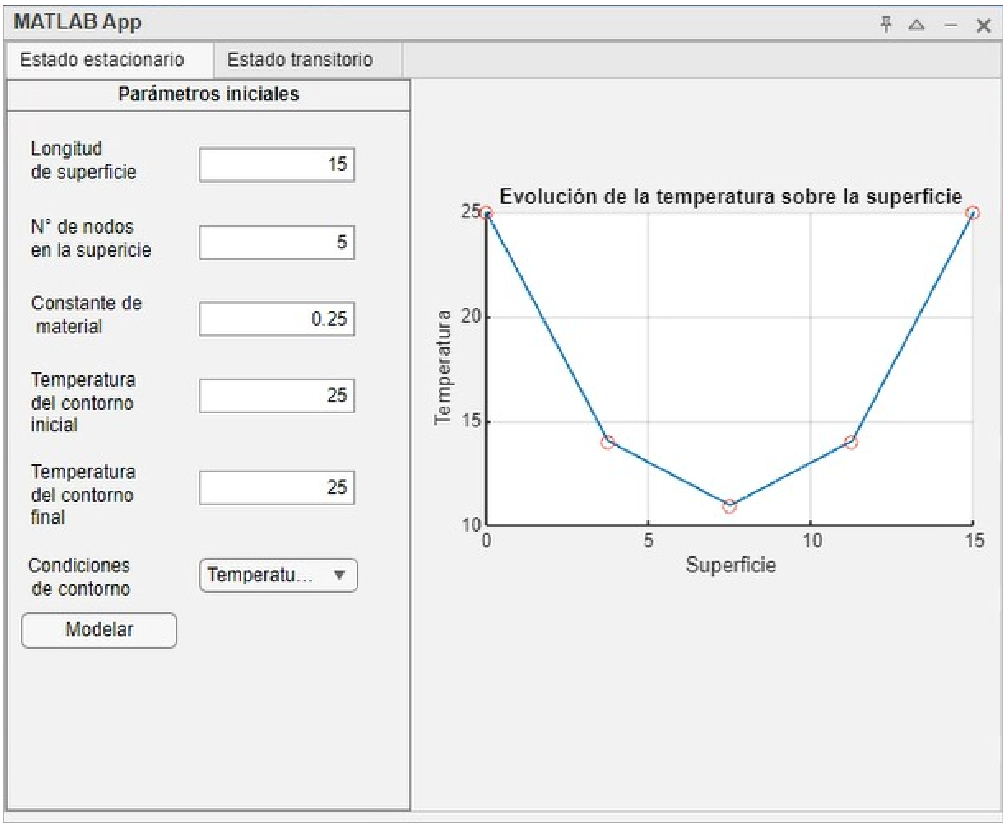
Así trasladamos esta idea a código en MATLAB:

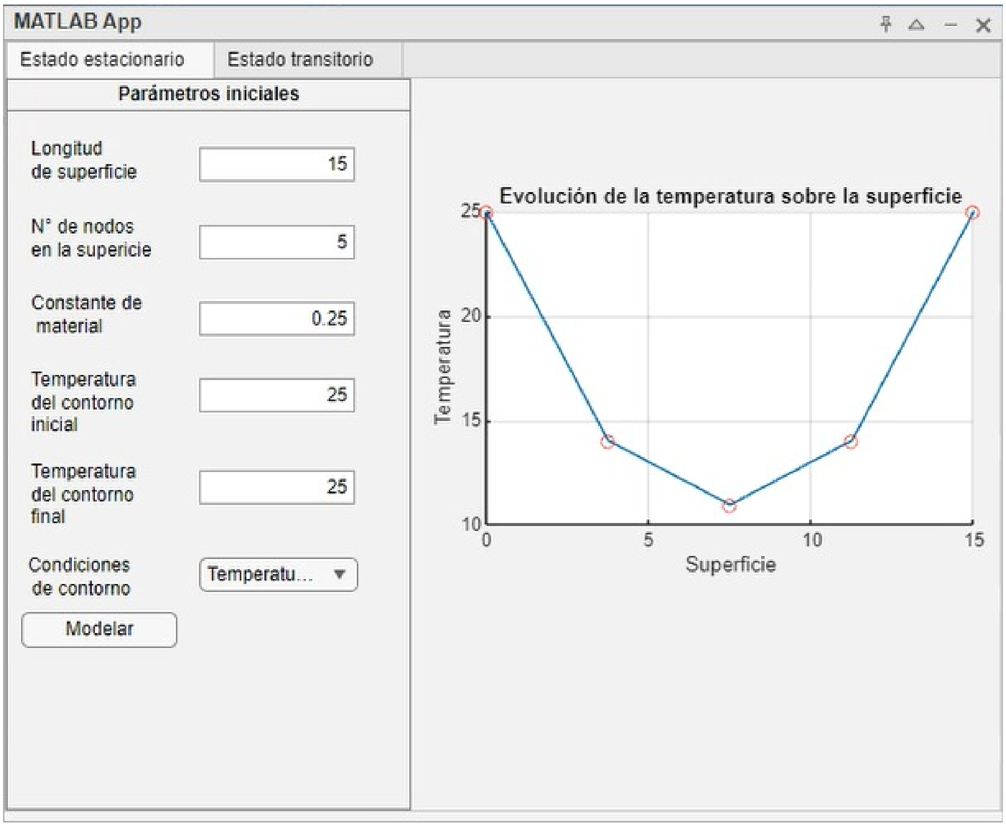
```
            if condicionContorno == "Temperatura fija del primer y el último nodo"
                nodosACalcular = n-2;
            elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo" || condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                nodosACalcular = n-1;
            end

            originalN = n
            gamma_cte = alpha^2*(L/n)^2;

            n = nodosACalcular;

            A = zeros(n,n);
            b = zeros(n,1);

            if condicionContorno == "Temperatura fija del primer y el último nodo"
                b(1) = -T;
                b(end) = -T2;
            elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                b(1) = -T;
            elseif condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                b(end) = -T2;
            end

            for i = 1:n
                A(i,i) = -(2+gamma_cte);
                if i == 1
                    if condicionContorno == "Temperatura fija del primer y el último nodo" || condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                        A(i,2) = 1;
                    elseif condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                        A(i,2) = 2;
                    end
                elseif i == n
                    if condicionContorno == "Temperatura fija del primer y el último nodo" || condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                        A(i,i-1) = 1;
                    elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                        A(i,i-1) = 2;
                    end
                else
                    A(i,i-1) = 1;
                    A(i,i+1) = 1;
                end
            end

            [solution,~] = partial_pivot(A,b);
           final_solution = [];
            if condicionContorno == "Temperatura fija del primer y el último nodo"
                final_solution = [T;solution;T2];
            elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                final_solution = [T;solution];
            elseif condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                final_solution = [solution;T2];
            end

            x_axis = linspace(0,L,originalN);
            plot(app.UIAxes,x_axis,final_solution,'-o', ...
                'MarkerEdgeColor','r')
        end
```

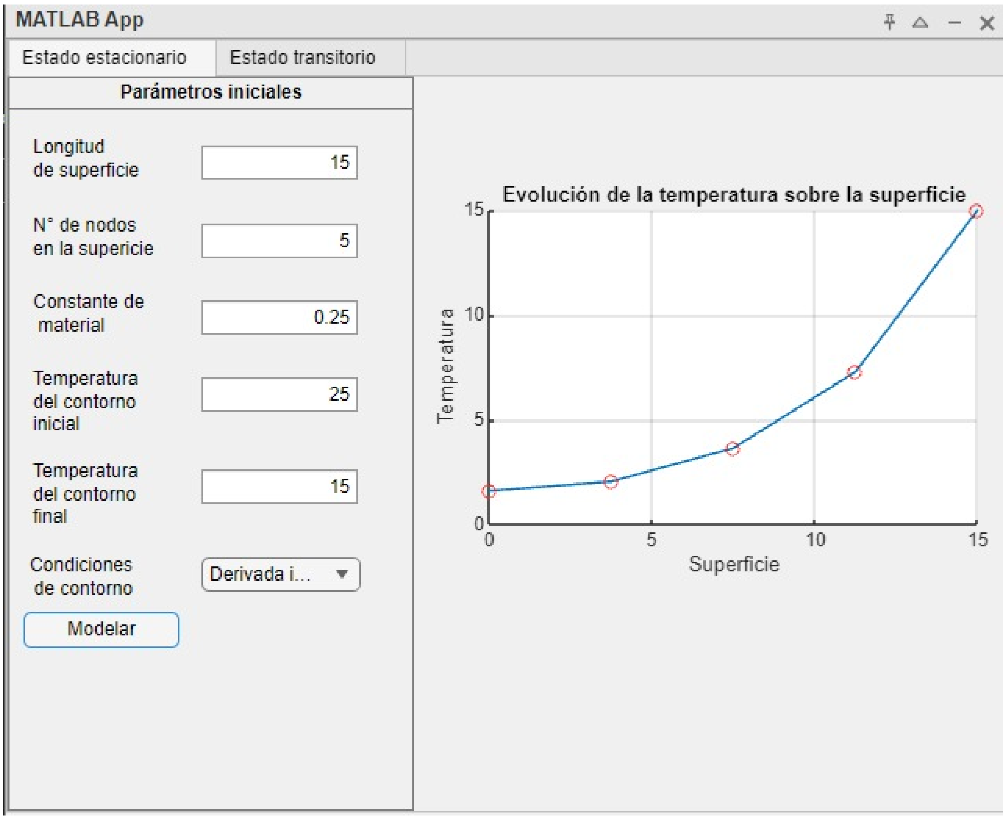
Figura 2: Código en MATLAB para simular el estado estacionario.

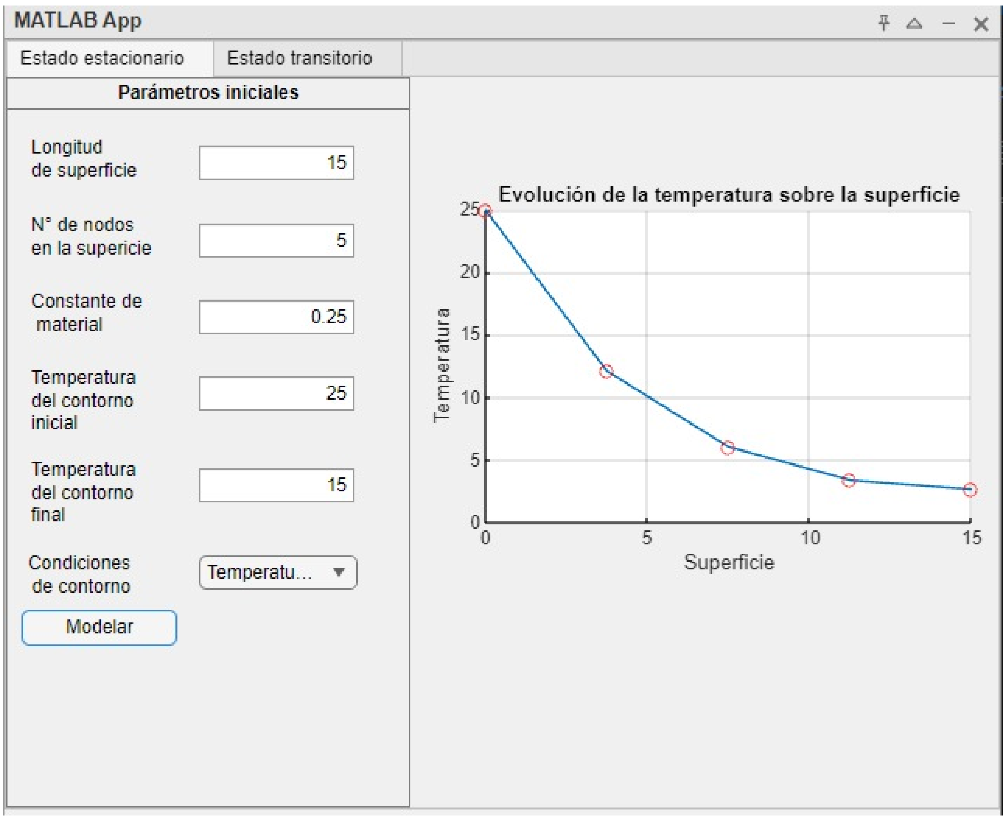
3.1. Resultados

A continuación se mostrarán imágenes de simulaciones para diferentes variables iniciales.


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
N° de nodos en la supericie
5
Constante de material
0.25
Temperatura del contorno inicial
25
Temperatura del contorno final
25
Condiciones de contorno
Temperatu...
Modelar
Evolución de la temperatura sobre la superficie
Temperatura
Superficie
25
20
15
10
0
5
10
15


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
N° de nodos en la supericie
5
Constante de material
0.25
Temperatura del contorno inicial
25
Temperatura del contorno final
25
Condiciones de contorno
Temperatu...
Modelar
Evolución de la temperatura sobre la superficie
Temperatura
Superficie
25
20
15
10
0
5
10
15


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
N° de nodos en la supericie
5
Constante de material
0.25
Temperatura del contorno inicial
25
Temperatura del contorno final
15
Condiciones de contorno
Derivada i...
Modelar
Evolución de la temperatura sobre la superficie
Temperatura
Superficie


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
N° de nodos en la supericie
5
Constante de material
0.25
Temperatura del contorno inicial
25
Temperatura del contorno final
15
Condiciones de contorno
Temperatu...
Modelar
Evolución de la temperatura sobre la superficie
Temperatura
Superficie

4. Estado transitorio

La ecuación para el estado transitorio es:

$$\frac{\partial^2 T}{\partial x^2} - \alpha^2 T = \frac{\partial T}{\partial t}.$$

Se trata de una ecuación diferencial parcial de segundo orden que describe la temperatura en función de la posición y el tiempo. Para resolverla numéricamente, recurrimos al método de diferencias finitas. Dado que hay 2 variables independientes, al discretizar obtendremos m sistemas de ecuaciones con $n-1$ ecuaciones cada uno, donde n es el número de nodos en la superficie y m es el número de nodos de tiempo.

Aproximamos el lado izquierdo de la ecuación igual que lo hicimos en el estado estacionario:

$$\left(\frac{\partial^2 T}{\partial x^2}\right)_{i,j} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2},$$

donde i indica los nodos en la superficie y j los nodos de tiempo. Además, para la derivada que se encuentra al lado derecho de la ecuación utilizamos el método de diferencias finitas para atrás tal que:

$$\left(\frac{\partial T}{\partial t}\right)_{i,j} \approx \frac{T_{i,j} - T_{i,j-1}}{\Delta t}.$$

Así, la ecuación en cada nodo se expresa como:

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} - \alpha^2 T_{i,j} = \frac{T_{i,j} - T_{i,j-1}}{\Delta t}.$$

El siguiente paso es manipular esta ecuación para su implementación computacional.

$$\begin{aligned}
T_{i+1,j} - 2T_{i,j} + T_{i-1,j} - \Delta x^2 \alpha^2 T_{i,j} &= \frac{T_{i,j} - T_{i,j-1}}{\Delta t}\Delta x^2 \\
T_{i+1,j} + T_{i-1,j} - (2 + \Delta x^2 \alpha^2)T_{i,j} - \frac{T_{i,j}}{\Delta t}\Delta x^2 &= -\frac{T_{i,j-1}}{\Delta t}\Delta x^2 \\
T_{i+1,j} + T_{i-1,j} - \left(2 + \Delta x^2 \alpha^2 - \frac{\Delta x^2}{\Delta t}\right) T_{i,j} &= -\frac{T_{i,j-1}}{\Delta t}\Delta x^2
\end{aligned}$$

Por lo tanto para calcular la temperatura en cada instante de tiempo j y en cada nodo de superficie j, tenemos estas ecuaciones:

$$
\begin{aligned}
T_{2,j} + T_{0,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{1,j} &= -\frac{T_{1,j-1}}{\triangle t}\triangle x^2 \\
T_{3,j} + T_{1,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{2,j} &= -\frac{T_{2,j-1}}{\triangle t}\triangle x^2 \\
&\vdots \\
T_{n+1,j} + T_{n-1,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{n,j} &= -\frac{T_{n,j-1}}{\triangle t}\triangle x^2.
\end{aligned}
$$

Cabe resaltar que para este modelo requerimos de un nuevo dato inicial que es la temperatura inicial de la simulación. Así que en instante de tiempo inicial que para nosotros es $j = 1$, la temperatura tomará ese valor inicial de la simulación. Además, aquí volvemos a tener el mismo problema que en el caso estacionario ya que existen n ecuaciones para cada nodo de superficie y $n-2$ incógnitas de temperatura en cada instante de tiempo. Por lo cual debemos hacer uso, de nuevo, de las condiciones de frontera iniciales del contorno. Haciendo uso del raciocinio expuesto en el modelo estacionario, estos son los sistemas de ecuaciones a resolver para cada uno de los tres casos de condiciones de frontera posibles en nuestra simulación.

- **Temperatura fija del primer y el último nodo.**
  Haciendo uso del valor de la temperatura ambos nodos de fronteras podemos decir que $T_1, j = T_{nodoInicial}$ y que $T_n, j = T_{nodoFinal}$. De esa manera obtenemos el siguiente sistema de $n-2$ ecuaciones, que tiene que ser solucionado m veces.

$$
\begin{aligned}
T_{3,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{2,j} &= -\frac{T_{2,j-1}}{\triangle t}\triangle x^2 - T_{1,j} \\
T_{4,j} + T_{2,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{3,j} &= -\frac{T_{3,j-1}}{\triangle t}\triangle x^2 \\
&\vdots \\
T_{n-2,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{n-1,j} &= -\frac{T_{n-1,j-1}}{\triangle t}\triangle x^2 - T_{n,j}.
\end{aligned}
$$

  Luego, para un determinado instante de tiempo j, el sistema a resolver en forma matricial es:

$$
\begin{pmatrix}
-(2+\Delta x^2\alpha^2) & 1 & 0 & \cdots & 0 \\
0 & -(2+\Delta x^2\alpha^2) & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 1 & -(2+\Delta x^2\alpha^2)
\end{pmatrix}
\begin{pmatrix}
T_{2_j} \\ T_{3_j} \\ \vdots \\ T_{n-1,j}
\end{pmatrix}
=
$$

$$\begin{pmatrix} -\frac{T_{2,j-1}}{\triangle t}\triangle x^2 - T_{nodoInicial} \\ -\frac{T_{3,j-1}}{\triangle t}\triangle x^2 \\ \vdots \\ -\frac{T_{n-1,j-1}}{\triangle t}\triangle x^2 - T_{nodoFinal} \end{pmatrix}.$$

- **Temperatura fija del primer nodo y derivada igualada a cero del último nodo.**
  Para este caso, utilizaremos el valor de la temperatura del nodo inicial y ya que la derivada es cero en nodo final, podemos asumir que $T_{n+1,j} \approx T_{n-1,j}$. Por lo cual así se ve el sistema de $n-1$ ecuaciones para cada instante de tiempo j:

$$\begin{aligned} T_{3,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{2,j} &= -\frac{T_{2,j-1}}{\triangle t}\triangle x^2 - T_{1,j} \\ T_{4,j} + T_{2,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{3,j} &= -\frac{T_{3,j-1}}{\triangle t}\triangle x^2 \\ &\vdots \\ 2T_{n-1,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{n,j} &= -\frac{T_{n,j-1}}{\triangle t}\triangle x^2. \end{aligned}$$

  Así se ve el sistema representado en forma matricial:

$$\begin{pmatrix} -(2+\Delta x^2\alpha^2) & 1 & 0 & \cdots & 0 \\ 0 & -(2+\Delta x^2\alpha^2) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 2 & -(2+\Delta x^2\alpha^2) \end{pmatrix} \begin{pmatrix} T_{2_j} \\ T_{3_j} \\ \vdots \\ T_{n,j} \end{pmatrix} =$$

$$\begin{pmatrix} -\frac{T_{2,j-1}}{\triangle t}\triangle x^2 - T_{nodoInicial} \\ -\frac{T_{3,j-1}}{\triangle t}\triangle x^2 \\ \vdots \\ -\frac{T_{n,j-1}}{\triangle t}\triangle x^2 \end{pmatrix}.$$

- **Derivada igualada a cero del primer nodo y temperatura fija del último nodo** En esta ocasión se presenta una analogía con el suceso anterior. Usaremos el valor de la temperatura del nodo final y al ser cero la derivada evaluada en el nodo inicial, asumiremos que $T_0 \approx T_2$. Por tanto, el sistema de de $n-1$ ecuaciones resultante para resolver en cada instante de tiempo

j será:

$$
\begin{aligned}
2T_{2,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{1,j} &= -\frac{T_{1,j-1}}{\triangle t}\triangle x^2 \\
T_{3,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{2,j} &= -\frac{T_{2,j-1}}{\triangle t}\triangle x^2 \\
&\vdots \\
T_{n-2,j} - \left(2 + \triangle x^2\alpha^2 - \frac{\triangle x^2}{\triangle t}\right) T_{n-1,j} &= -\frac{T_{n-1,j-1}}{\triangle t}\triangle x^2 - T_{n,j}.
\end{aligned}
$$

Lo podemos representar de forma matricial como:

$$
\begin{pmatrix}
-(2+\Delta x^2\alpha^2) & 2 & 0 & \cdots & 0 \\
0 & -(2+\Delta x^2\alpha^2) & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 1 & -(2+\Delta x^2\alpha^2)
\end{pmatrix}
\begin{pmatrix}
T_{1_j} \\ T_{2_j} \\ \vdots \\ T_{n-1,j}
\end{pmatrix}
=
$$

$$
\begin{pmatrix}
-\frac{T_{1,j-1}}{\triangle t}\triangle x^2 \\
-\frac{T_{2,j-1}}{\triangle t}\triangle x^2 \\
\vdots \\
-\frac{T_{n-1,j-1}}{\triangle t}\triangle x^2 - T_{nodoFinal}
\end{pmatrix}.
$$

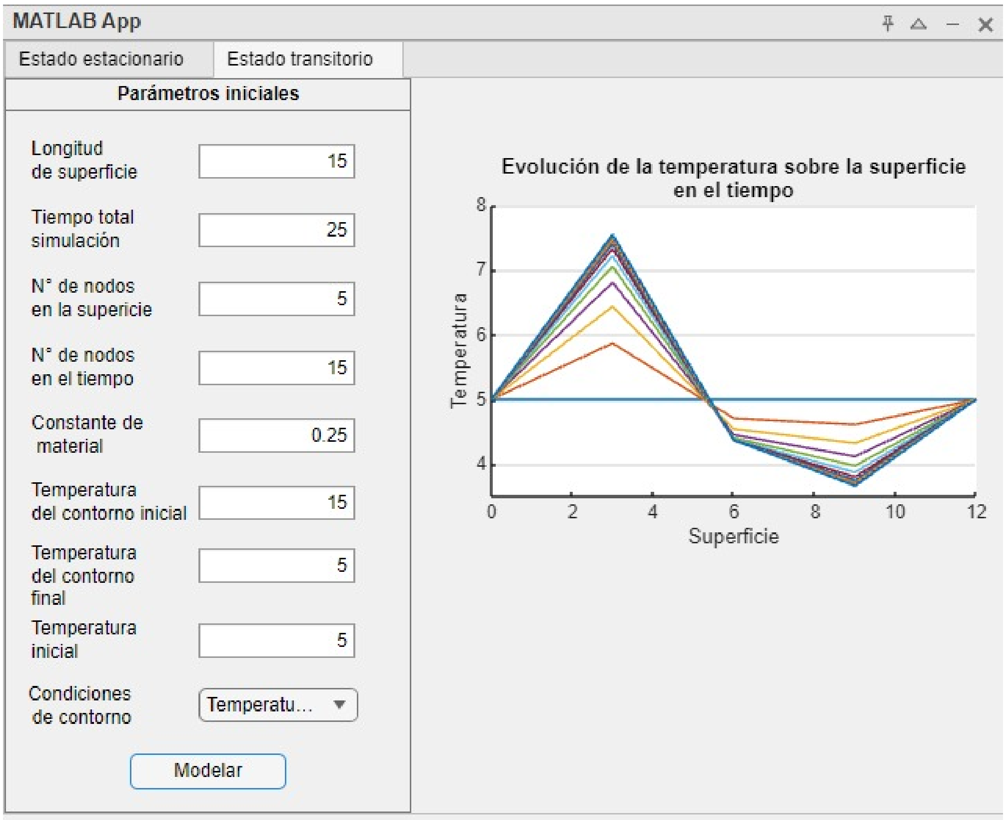
Llevando la teoría a código, esto fue lo que realizamos en MATLAB:

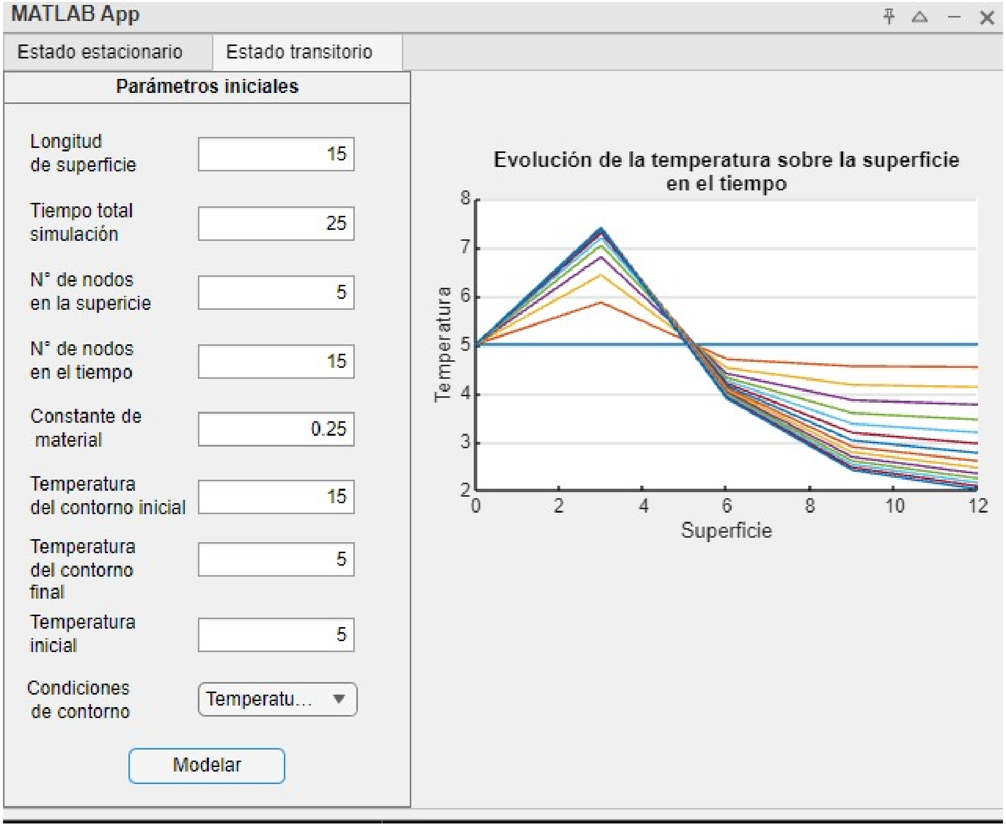
```matlab
difusionCalorTransitorio(L,t,n_L,n_t,alpha,Tc,Ti,Tf,condicionContorno)
                originalN_l = n_L;

                if condicionContorno == "Temperatura fija del primer y el último nodo"
                    n_L = n_L-2;
                elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo" || condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                    n_L = n_L-1;
                end

                A = zeros(n_L,n_L,n_t-1);
                b = zeros(n_L,n_t-1);

                dx = L/originalN_l;
                dt = t/n_t;

                gamma_cte = (alpha^2*dx^2) + dx^2/dt;
                T = zeros(n_L,n_t);
                T(:,1) = Ti+zeros(n_L,1);
                for j = 1:n_t-1
                    for i = 1:n_L
                        A(i,i,j) = -(2+gamma_cte);
                        if i == 1
                            if condicionContorno == "Temperatura fija del primer y el último nodo" || condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                                A(i,2,j) = 1;
                                b(i,j) = -Tc - (T(i,j)/(t/n_t))*dx^2;
                            elseif condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                                A(i,2, j) = 2;
                                b(i,j) = - T(i,j)/(t/n_t)*dx^2;
                            end

                        elseif i == n_L
                            if condicionContorno == "Temperatura fija del primer y el último nodo" || condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                                A(i,i-1,j) = 1;
                                b(i,j) = -Tf - (T(i,j)/(t/n_t))*dx^2;
                            elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                                A(i,i-1,j) = 2;
                                b(i,j) = - T(i,j)/(t/n_t)*dx^2;
                            end
                        else
                            A(i,i-1,j) = 1;
                            A(i,i+1,j) = 1;
                            b(i,j) = - T(i,j)/(t/n_t)*dx^2;
                        end
                    end
                    j,A(:,:,j),b(:,j)


                    [solution,~] = partial_pivot(A(:,:,j),b(:,j));
                    T(:,j+1) = solution;
                end
                inicial=[0];
                [a,b]=size(T);
                for i=1:originalN_l-1
                    inicial(i+1)=inicial(i)+dx;
                end
                inicial=transpose(inicial)
                for k=1:b
                    A=T(:,k);

                    if condicionContorno == "Temperatura fija del primer y el último nodo"
                        A = [Ti;A;Tf];
                    elseif condicionContorno == "Temperatura fija del primer nodo y derivada igualada a cero del ultimo nodo"
                        A = [Ti;A];
                    elseif condicionContorno == "Derivada igualada a cero del primer nodo y temperatura fija del último nodo"
                        A = [A;Tf];
                    end

                    plot(app.UIAxes_2,inicial,A)
                    hold(app.UIAxes_2,'on')
                end
                hold (app.UIAxes_2,'off')
            end
            function [x, iter] = partial_pivot(A, b)
                n = length(b);
                x = zeros(n, 1);
                iter = 0;

                for k = 1:n-1
                    [~, i_max] = max(abs(A(k:n, k)));
                    i_max = i_max + k - 1;

                    A([k, i_max], :) = A([i_max, k], :);
                    b([k, i_max]) = b([i_max, k]);
                    iter = iter + 1;

                    for i = k+1:n
                        factor = A(i, k) / A(k, k);
                        A(i, k:n) = A(i, k:n) - factor * A(k, k:n);
                        b(i) = b(i) - factor * b(k);
                        iter = iter + 1;
                    end
                end

                x(n) = b(n) / A(n, n);
                for i = n-1:-1:1
                    x(i) = (b(i) - A(i, i+1:n) * x(i+1:n)) / A(i, i);
                    iter = iter + 1;
                end
            end
        end
    end
```

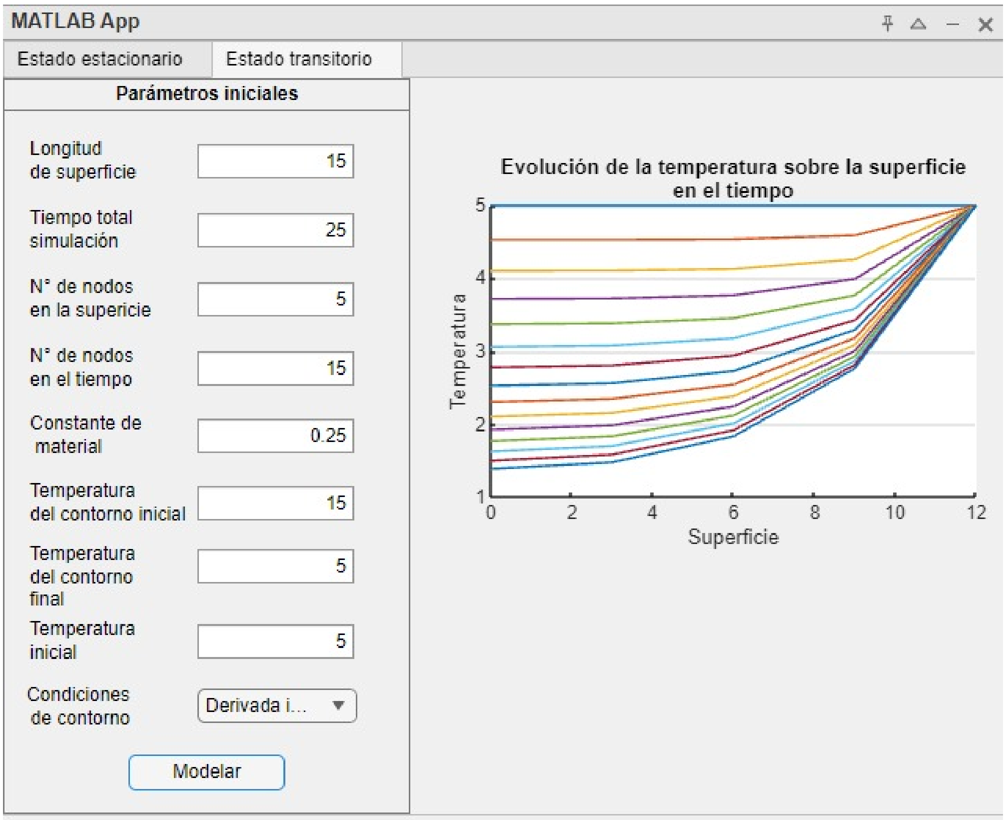
Figura 3: Código en MATLAB para simular el estado transitorio.

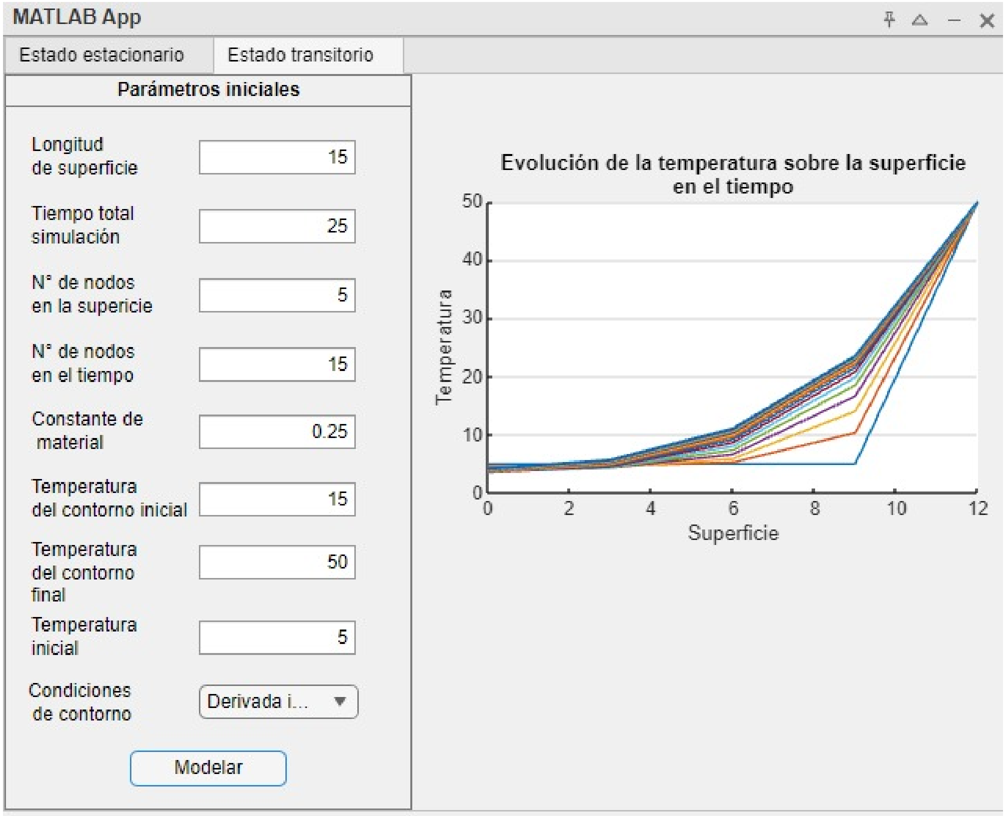
4.1. Resultados

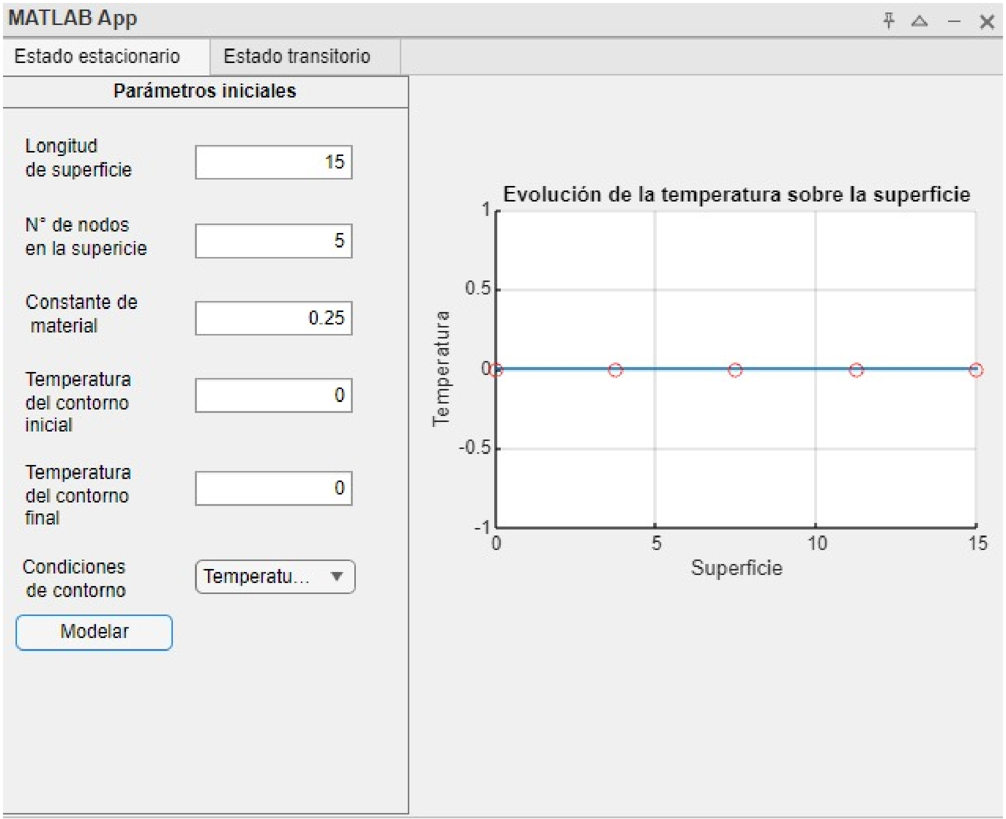
Para finalizar, así se ven diferentes modelaciones realizadas con nuestro código en el estado transitorio.




MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
Tiempo total simulación
25
N° de nodos en la supercie
5
N° de nodos en el tiempo
15
Constante de material
0.25
Temperatura del contorno inicial
15
Temperatura del contorno final
5
Temperatura inicial
5
Condiciones de contorno
Temperatu...
Modelar
Evolución de la temperatura sobre la superficie en el tiempo
Temperatura
Superficie


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
Tiempo total simulación
25
N° de nodos en la supercie
5
N° de nodos en el tiempo
15
Constante de material
0.25
Temperatura del contorno inicial
15
Temperatura del contorno final
5
Temperatura inicial
5
Condiciones de contorno
Derivada i...
Modelar
Evolución de la temperatura sobre la superficie en el tiempo
Temperatura
Superficie


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
Tiempo total simulación
25
N° de nodos en la supercie
5
N° de nodos en el tiempo
15
Constante de material
0.25
Temperatura del contorno inicial
15
Temperatura del contorno final
50
Temperatura inicial
5
Condiciones de contorno
Derivada i...
Modelar
Evolución de la temperatura sobre la superficie en el tiempo
Temperatura
Superficie


MATLAB App
Estado estacionario
Estado transitorio
Parámetros iniciales
Longitud de superficie
15
N° de nodos en la supercie
5
Constante de material
0.25
Temperatura del contorno inicial
0
Temperatura del contorno final
0
Condiciones de contorno
Temperatu...
Modelar
Evolución de la temperatura sobre la superficie
Temperatura
Superficie

5. Conclusiones

De acuerdo a las simulaciones realizadas con los diferentes parámetros iniciales posibles, podemos notar como el disipador de calor sí funciona de manera efectiva para que la temperatura vaya disminuyendo conforme se atraviesa su superficie.