Proyecto final

Difusión de un disipador de calor

Juan Manuel Ramírez Osuna

Universidad del Rosario
Escuela de Ingeniería, Ciencia y Tecnología
Noviembre 29, 2023

1 Introducción

El proyecto de esta asignatura está basado en solucionar la ecuación diferencial que describe el comportamiento de difusión de un disipador de calor. Se dividió la solución en dos casos; el caso estacionario donde solo se varia la temperatura según la distancia en x y el caso transitorio donde además del cambio anterior se tiene un cambio en términos del tiempo transcurrido. Las ecuaciones empleadas fueron las siguientes

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

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

2 Interfaz y requisitos

Para la interfaz gráfica se utilizo matlab app designer. El resultado fue el siguiente

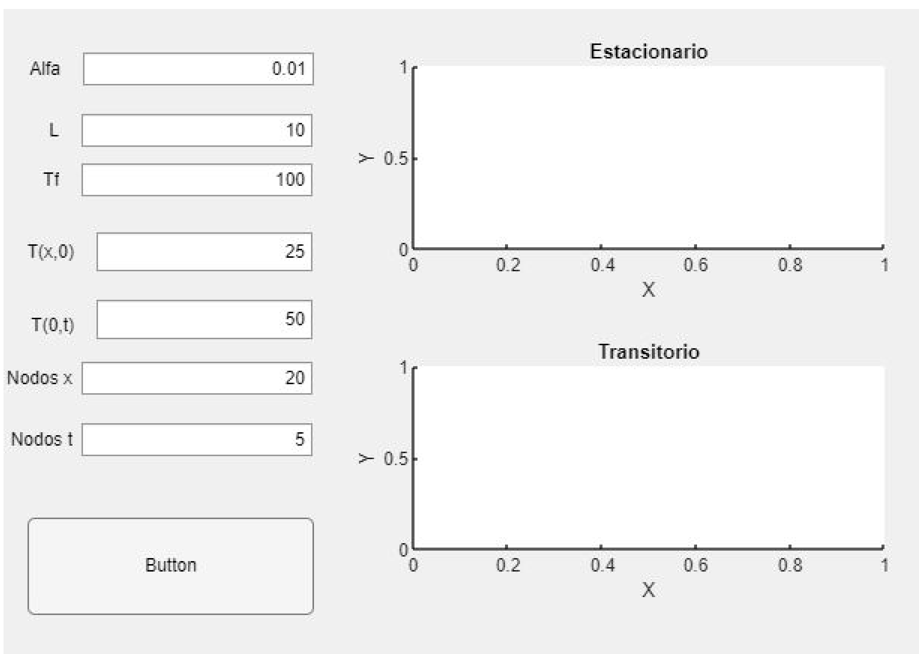

Figure 1: Interfaz grafica

Las variables requeridas para el funcionamiento son alfa, el tamaño de la barra, duración de las iteraciones, las condiciones iniciales de distancia y tiempo junto a la cantidad de nodos tanto para distancia como para tiempo. Es importante notar que todas las entradas son necesarias para el caso transitorio, pero el caso estacionario simplemente ignora las condiciones de tiempo.

3 Solución y código

3.1 Estacionario

Empezando por el caso estacionario, se tiene la ecuación:

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

Para la cual, usando el método de diferencias se obtuvo:

$$T_{j+1} - 2T_j + T_{j-1} - \alpha^2 \Delta x^2 T_j = 0$$

Es importante tener en cuenta que para la segunda posición o $x = \Delta x$ se tiene que T_{j-1} es la condición inicial de distancia, además para la ultima posición o $x = L$ se tiene que $T_{j+1} = T_{j-1}$ dado que se considera la condición inicial de $\frac{\delta T}{\delta x} = 0$ en este punto y por tanto se asume un comportamiento parabólico. En términos de código se inicia por la sección de almacenamiento de variables necesarias:

```
alfa = app.AlfaEditField.Value;
L = app.LEditField.Value;

dTxL = 0; % Se asume 0 dado que la temperatura no se puede seguir propagando
Tx0 = app.Tx0EditField.Value;

nx = app.NodosxEditField.Value;
dx = L/(nx-1);
```

Se procede a construir las matriz necesaria de variables desconocidas, usando la solución del método de diferencias, las cuales solo cambian para la segunda posición en el caso estacionario. La matriz se construye asumiendo todas las variables que podrían considerarse y finalmente se ajusta la ultima linea y se reduce la dimensionalidad para tener una matriz cuadrada.

```
a = zeros(nx-1,nx+1);
a(1,2) = -2 - alfa^2*dx;
a(1,3) = 1;
for i = 2:nx-1
    a(i,i) = 1;
    a(i,i+1) = -2 - alfa^2*dx;
    a(i,i+2) = 1;
end
a(nx-1, end) = 0;
a(nx-1, nx-1) = 2;
a = a(:,2:nx);
```

La creación del vector b es similar, solo se cambia la primera posición por la condición inicial en x y el resto se mantiene en 0.

```
b = zeros(nx-1,1);
b(1) = -Tx0;
```

La solución de la temperatura se realiza con el método de pivoteo parcial visto en clase. Se utiliza 0 como un entero cualquiera para el parámetro de redondeo y falso para evitar el conteo de operaciones. Se agrega la temperatura de la posición inicial al vector para la gráfica y se aplica a la primera gráfica.

```
T = pivoteo(a,b,'parcial',0,false);
T = [Tx0; T];
plot(app.UIAxes,0:dx:L, T)
```

3.2 Transitorio

Para el caso transitorio, se tiene la ecuación:

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

Para la cual, usando el método de diferencias, aplicando diferencias hacia atras en tiempo, se obtuvo:

$$\frac{1}{\Delta x^2}T_{j+1,i} + (-\frac{2}{\Delta x^2} - \alpha^2 - \frac{1}{\Delta t})T_{j,i} + \frac{1}{\Delta x^2}T_{j-1,i} + \frac{1}{\Delta t}T_{j,i-1} = 0$$

Las aclaraciones del caso anterior se mantienen. Al agregar el tiempo también se tiene que tener en cuenta que para el segundo momento o $t = \Delta t$ se tiene que $T_{j,i-1}$ es la condición inicial de tiempo. En términos de código también se inicia por la sección de almacenamiento de variables necesarias:

```
alfa = app.AlfaEditField.Value;
L = app.LEditField.Value;
tf = app.TfEditField.Value;

Tt0 = app.Tx0EditField.Value;
dTxL = 0; % Se asume 0 dado que la temperatura no se puede seguir propagando
Tx0 = app.T0tEditField.Value;

nt = app.NodostEditField.Value;
nx = app.NodosxEditField.Value;

dx = L / (nx-1);
dt = tf / (nt-1);
```

Desde este punto se inicia un ciclo sobre los nodos de tiempo. Para la solución se incluyo la condición del tiempo 0 como un caso aparte en la construcción de la matriz a y el vector b. Para la construcción de a se creo una diagonal de unos ya que las temperaturas iniciales son conocidas al ser fijadas por el usuario. Desde el segundo momento se vuelve al modo de definir la primera linea según la condición inicial de distancia. Las demás lineas se hacen con la solución de diferencias hallada previamente. Al final se vuelve a ajustar la ultima linea según la condición de la derivada al final de la barra.

```
for i = 1:nt
    if i == 1
        a = diag(ones(1,nx-1));
```

```
        else
            a = zeros(nx-1,nx+1);
            a(1,2) = -2/dx^2 - alfa^2-1/dt;
            a(1,3) = 1/dx^2;
            for j = 2:nx-1
                a(j,j) = 1/dx^2;
                a(j,j+1) = -2/dx^2 - alfa^2-1/dt;
                a(j,j+2) = 1/dx^2;
            end
            a(nx-1, nx-1) = 2*a(nx-1, nx-1);
            a(nx-1, end) = 0;
            a = a(:,2:nx);
        end
```

Similarmente a la construcción de a se toman los dos casos. En el primer caso se genera un vector con las temperaturas iniciales en t dado que la solución se realiza solo desde el segundo momento. En el segundo caso se borra la temperatura inicial que sea agrega en cada iteración y se multiplican las demás temperaturas por el cambio en el tiempo. La primera entrada del vector se ajusta para incluir la condición inicial de la distancia la cual es invariable en el tiempo.

```
        if i == 1
            b = ones(nx-1,1)*Tt0;
        else
            b = - T(2:end)/dt;
            b(1) = -Tx0/dx^2-Tt0/dt;
        end
```

Se llega al final donde se realiza la solución con la misma función que en el caso estacionario agregando la temperatura de la posición inicial para la gráfica. Al graficar se agregan parámetros para nombrar las curvas. Al finalizar las iteraciones sobre el tiempo se imprime la leyenda de la gráfica para identificar las curvas según su tiempo.

```
        T = pivoteo(a,b,'parcial',0,false);
        T = [Tx0; T];

        plot(app.UIAxes_2, 0:dx:L, T,'DisplayName',['T = ',num2str((i-1)*dt)])
        hold(app.UIAxes_2, 'on')
    end
    hold(app.UIAxes_2, 'off')
    legend(app.UIAxes_2)
```

4 Ejemplo

Para el ejemplo se utilizaron los valores que se pusieron como predeterminados en la aplicación con los cuales se llega a observar como el calor a lo largo de la barra se va igualando a la condición inicial de distancia. Estos valores fueron $\alpha = 0.01$, L = 10, Tf = 100, condicion inicial de tiempo de 25, condicion inicial de distancia de 50, 20 nodos de distancia y 5 de tiempo. El resultado y los valores ingresados se pueden ver en la siguiente figura:

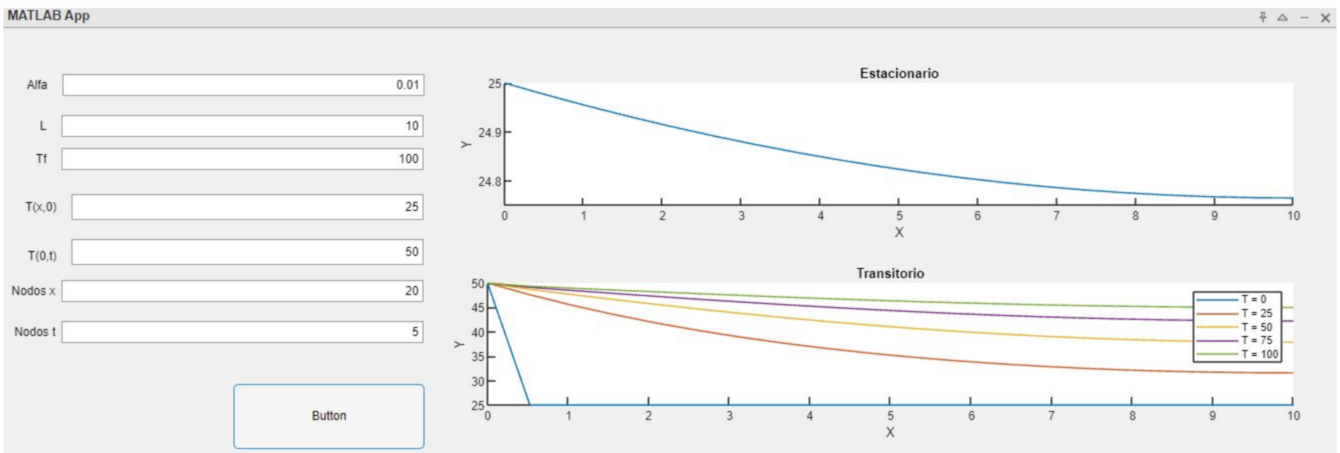

Figure 2: Ejemplo de funcionamiento