Universidad de Guanajuato
Método de Volumen Finito (Proyecto III)

García Sánchez Marco Antonio[1]
[1]División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato, Salamanca, Gto.

Docente: Damián Ascencio César Eduardo

November 2019

Resumen

Se presenta el método de Volumen Finito para analizar la evolución temporal de la distribución de temperatura. Se comparan diferentes esquemas de discretización: método explícito, método implícito y Crank–Nicolson (teórico), y se utilizan esquemas implícito y explícito para resolver numéricamente las ecuaciones obtenidas.

1. Introducción

El método de Volumen Finito permite discretizar y resolver numéricamente ecuaciones diferenciales. Es un método alternativo a los de diferencias finitas y elementos finitos. Se considera una malla de discretización del dominio. En torno a cada punto de esta malla se construye un volumen de control que no se traslapa con los volúmenes de los puntos vecinos. De esta forma, el volumen total del dominio resulta ser igual a la suma de los volúmenes de control considerados.
La ecuación diferencial a resolver se integra sobre cada volumen de control, lo cual entrega como resultado una versión discretizada de dicha ecuación. Para realizar la integración se requiere especificar perfiles de variación de la variable dependiente entre los puntos de la malla, para poder evaluar las integrales resultantes. La principal propiedad del sistema de ecuaciones discretizadas resultante es que la solución obtenida satisface en forma exacta las ecuaciones de conservación consideradas, independientemente del tamaño de la malla. Esto aplica tanto a problemas en estado estacionario como en estado transitorio.
El estado estacionario es aquel en el que no existen cambios de las variables de proceso. En el estado transitorio las variables de proceso cambian en el tiempo debido a la existencia de acumulación de energía o materia.

2. Ecuaciones gobernantes

Un esquema junto con los datos del problema que se atacará se muestran en la Figura 1.
La ecuación gobernante está dada por la siguiente expresión (ecuación unidimensional de difusión térmica):

$$\frac{\partial T}{\partial t} = D\frac{\partial^2 T}{\partial x^2}, \tag{1}$$

donde T es la temperatura, t el tiempo y D la difusividad térmica (constante en este trabajo).

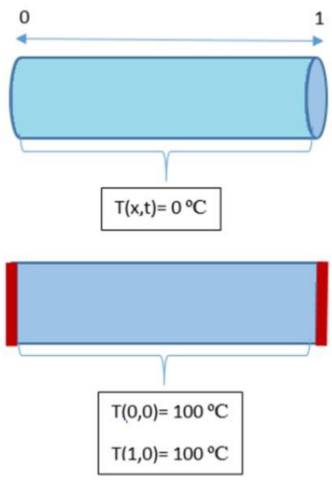


Figura 1. Esquema del problema.

3. Discretización

Discretizando la ecuación gobernante se obtiene, integrando en el volumen de control y en el intervalo temporal $[t^n, t^{n+1}]$:

$$\int_V \int_{t^n}^{t^{n+1}} \frac{\partial T}{\partial t}\, dt\, dV = D \int_{t^n}^{t^{n+1}} \left(\int_A \frac{\partial T}{\partial x}\, dA \right) dt.$$

Para poder unificar esquemas explícito, implícito y mixto se introduce el parámetro $\Theta \in [0, 1]$, de modo que:

$$\Theta = 0 \Rightarrow \text{esquema explícito}, \qquad \Theta = 1 \Rightarrow \text{esquema implícito}, \qquad \Theta = \tfrac{1}{2} \Rightarrow \text{Crank–Nicolson}.$$

Asumiendo una malla uniforme con paso espacial Δx y paso temporal Δt, y notando T_P^n la temperatura en el nodo P en el instante t^n, una forma compuesta (discretización Θ) para un nodo central es:

$$\begin{aligned} T_P^{n+1} = T_P^n + \frac{D\Delta t}{\Delta x^2} \Big[& \Theta \big(T_E^{n+1} - 2T_P^{n+1} + T_W^{n+1} \big) \\ & + (1 - \Theta) \big(T_E^n - 2T_P^n + T_W^n \big) \Big]. \end{aligned} \tag{2}$$

Reordenando, la ecuación nodal para los nodos interiores queda:

$$a_P T_P^{n+1} - a_W T_W^{n+1} - a_E T_E^{n+1} = b, \tag{3}$$

con

$$a_W = a_E = -\Theta \frac{D\Delta t}{\Delta x^2}, \qquad a_P = 1 + 2\Theta \frac{D\Delta t}{\Delta x^2},$$

y

$$b = T_P^n + (1 - \Theta) \frac{D\Delta t}{\Delta x^2} \big(T_E^n - 2T_P^n + T_W^n \big).$$

De aquí se obtienen los esquemas:

- Explícito ($\Theta = 0$): se despeja T_P^{n+1} directamente.
- Implícito ($\Theta = 1$): se resuelve un sistema lineal para T^{n+1}.
- Crank–Nicolson ($\Theta = \frac{1}{2}$): combinación de ambos.

3.1. Nodos en fronteras (nodos fantasma)

Para implementar condiciones de Dirichlet en los extremos $x = 0$ y $x = L$ se usan nodos fantasma cuando es necesario mantener el mismo esquema de diferencias para todos los nodos (esto se detalla en las ecuaciones específicas más adelante).

3.2. Ecuaciones particulares (resumen)

Para mantener la notación coherente con el original, a continuación se exponen las expresiones equivalentes corregidas y con $\Delta x, \Delta t$ consistentes.

3.2.1. Nodos centrales (forma compacta)

$$\left(\frac{\Delta x}{\Delta t} + \Theta\frac{2D}{\Delta x}\right) T_P^{n+1} - \Theta\frac{D}{\Delta x}(T_E^{n+1} + T_W^{n+1}) = \left(\frac{\Delta x}{\Delta t} - (1-\Theta)\frac{2D}{\Delta x}\right) T_P^{n} + (1-\Theta)\frac{D}{\Delta x}(T_E^{n} + T_W^{n}). \tag{4}$$

(Equivalente a tu desarrollo original, expresado con notación consistente.)

3.2.2. Nodos en extremos

Nodo 1 (frontera izquierda, con nodo fantasma en W):

$$\begin{aligned} \left(\frac{\Delta x}{\Delta t} + \Theta\left(\frac{D_E}{\Delta x} + 2\frac{D_W}{\Delta x}\right)\right) T_1^{n+1} - \Theta\frac{D_E}{\Delta x} T_2^{n+1} = \\ \left(\frac{\Delta x}{\Delta t} + (1-\Theta)\left(-\frac{D_E}{\Delta x} + 2\frac{D_W}{\Delta x}\right)\right) T_1^{n} + (1-\Theta)\frac{D_E}{\Delta x} T_2^{n} + (2\Theta\frac{D_W}{\Delta x})T_a, \end{aligned} \tag{5}$$

donde T_a es la temperatura impuesta en la pared/frontera según el caso.

Nodo n (frontera derecha, con nodo fantasma en E):

$$\begin{aligned} \left(\frac{\Delta x}{\Delta t} + \Theta\left(2\frac{D_E}{\Delta x} + \frac{D_W}{\Delta x}\right)\right) T_n^{n+1} - \Theta\frac{D_W}{\Delta x} T_{n-1}^{n+1} = \\ \left(\frac{\Delta x}{\Delta t} + (1-\Theta)\left(-2\frac{D_E}{\Delta x} - \frac{D_W}{\Delta x}\right)\right) T_n^{n} + (1-\Theta)2\frac{D_E}{\Delta x} T_b^{n} + (1-\Theta)\frac{D_W}{\Delta x} T_{n-1}^{n}, \end{aligned} \tag{6}$$

donde T_b es la temperatura en la frontera derecha impuesta según el problema.

3.3. Esquemas explícito e implícito (formas particulares)

3.3.1. Explícito ($\Theta = 0$)

Nodos centrales

$$\frac{\Delta x}{\Delta t} T_P^{n+1} = \left(\frac{\Delta x}{\Delta t} - \frac{2D}{\Delta x}\right) T_P^{n} + \frac{D}{\Delta x}(T_E^{n} + T_W^{n}). \tag{7}$$

Nodo 1 y Nodo n (Se mantienen las mismas fórmulas del desarrollo general aplicando $\Theta = 0$; en la práctica se usan nodos fantasma para imponer las condiciones de frontera tal como en tu versión original.)

3.3.2. Implícito ($\Theta = 1$)

Nodos centrales

$$\left(\frac{\Delta x}{\Delta t} + \frac{2D}{\Delta x}\right) T_P^{n+1} - \frac{D}{\Delta x}(T_E^{n+1} + T_W^{n+1}) = \frac{\Delta x}{\Delta t} T_P^{n}. \tag{8}$$

Nodos en extremos (Se obtiene análogamente aplicando $\Theta = 1$ en las expresiones de frontera; se resuelven mediante la solución del sistema lineal resultante.)

4. Resultados e Independencia de Malla

Para la solución de las ecuaciones generadas en el método de Volumen Finito, es posible usar métodos numéricos para la solución iterativa de los sistemas de ecuaciones. Para el problema propuesto se implementó un código en Python (ver Anexos) que resuelve las ecuaciones discretizadas; el programa permite controlar el número de nodos espaciales y temporales así como las condiciones de frontera.

4.1. Esquema Implícito

Resolviendo numéricamente las ecuaciones discretizadas fue posible construir una solución temporal de la distribución de T. Las propiedades seleccionadas para la solución numérica se muestran en el Cuadro 1.

Cuadro 1: Propiedades del problema

Datos	**valor**	unidades
$T(x=0,t)$	100	°C
$T(x=1,t)$	100	°C
L	1	m
nodes_x	30	-
nodes_t	100	-
D	0.5	-
Θ	$0 \leq \Theta \leq 1$	-
$E_{propuesto}$	0.001	-

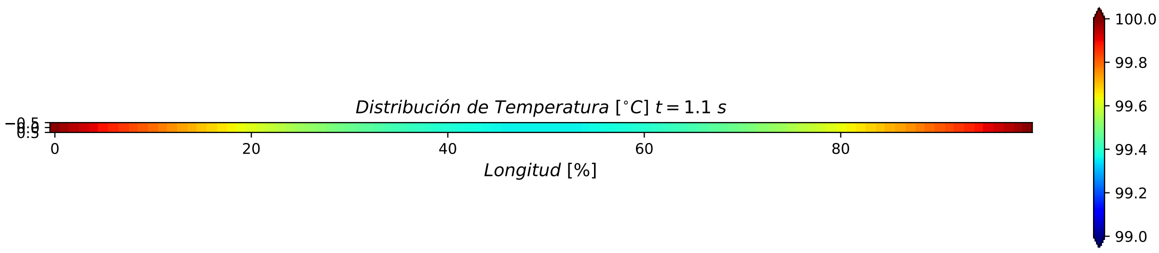
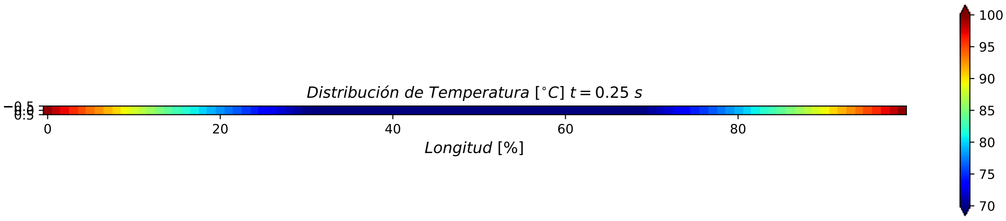
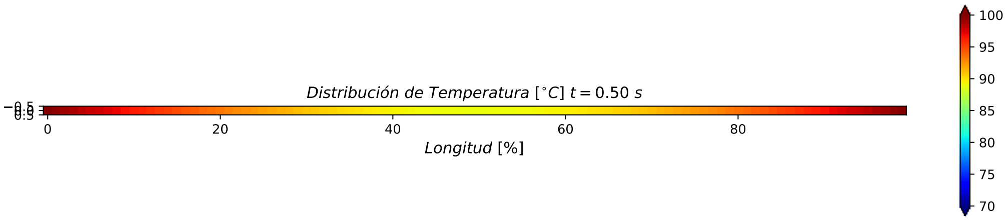


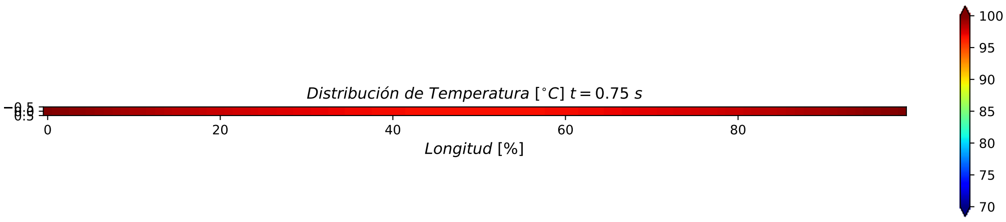
Figura 2. Método Implícito, $t = 1{,}1$ (instante aproximado de llegada a estado estable)



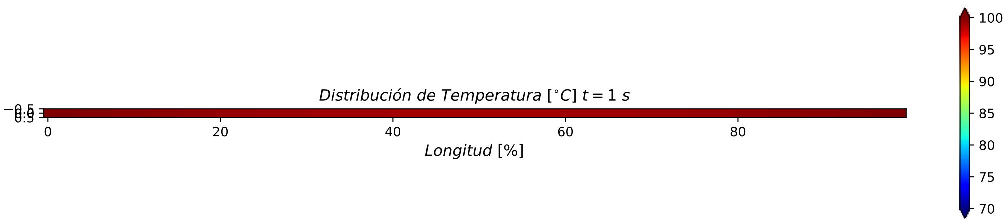
(a) Distribución de T en $t = 0{,}25$ s



(b) Distribución de T en $t = 0{,}5$ s



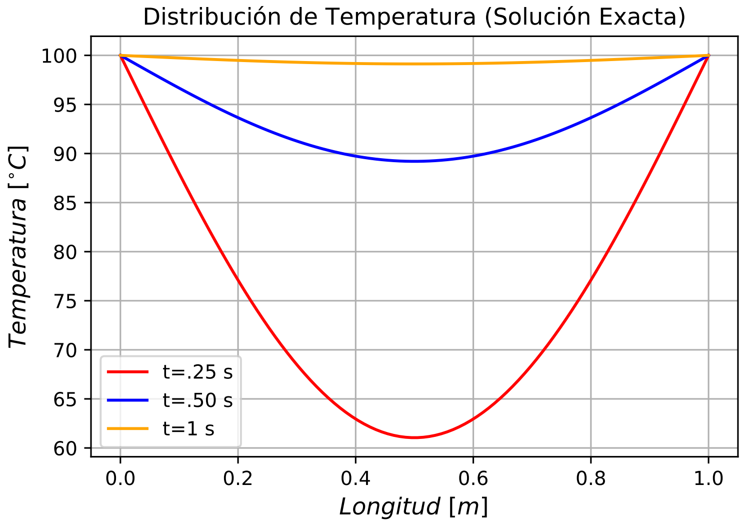
(c) Distribución de T en $t = 0{,}75$ s



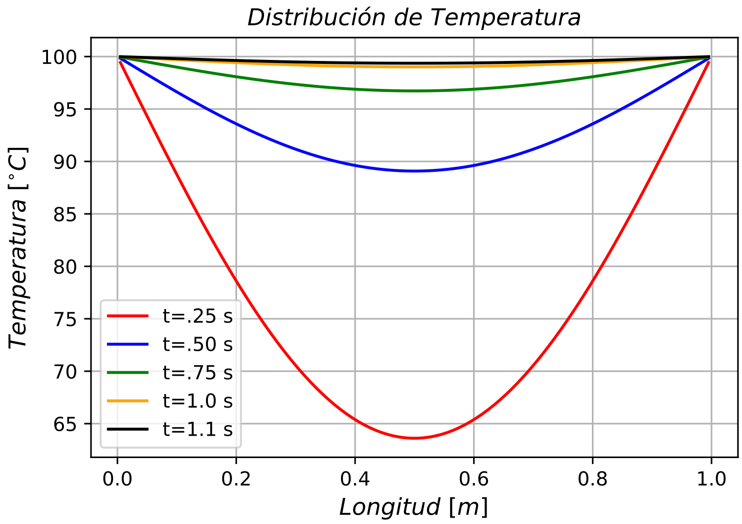
(d) Distribución de T en $t = 1$ s

Figura 1: Método Implícito: $0 \leq t \leq 1$.

En la Figura 2 se observa que la temperatura varía aproximadamente entre 99,5 °C y 100 °C en el instante $t = 1{,}1$ s, lo que indica que se alcanza un estado cercano al estacionario alrededor de 1.1 s con las condiciones impuestas.
La Figura 3 muestra la evolución temporal en diferentes instantes; se observa que para $t = 1$ s la solución ya se aproxima al estado estacionario, coherente con la Figura 2.
Dado que el esquema implícito no está restringido por un criterio de estabilidad condicional, se pueden usar Δt y Δx relativamente grandes comparados con el esquema explícito; en este caso se eligieron 100 nodos temporales para dividir el intervalo de tiempo hasta la estabilidad en pasos de 0,01 s y apreciar la evolución.



(a) Distribución de T (solución exacta)



(b) Distribución de T (numérica, implícita)

Figura 2: Comparación solución exacta vs numérica (implícita).

De acuerdo con la Figura 4 y calculando un error promedio por diferencia nodo a nodo (norma L_2 o promedio porcentual, según tu elección), se obtuvieron los valores reportados en el Cuadro 2.

Cuadro 2: Errores (implícito) — Error promedio

time	**Error Promedio**
$t = 0{,}25\ s$	1.45 %
$t = 0{,}50\ s$	1.034 %
$t = 1{,}0\ s$	1.23 %

4.2. Esquema Explícito

El esquema explícito tiene la particularidad de que su estabilidad depende del paso temporal y espacial. El criterio de estabilidad para el problema de difusión es:

$$\Delta t \leq \frac{\Delta x^2}{2D}. \tag{9}$$

El incumplimiento de esta condición provoca inestabilidad numérica en el esquema explícito.

Cuadro 3: Propiedades del problema (explícito)

Datos	**valor**	unidades
$T(x=0,t)$	100	°C
$T(x=1,t)$	100	°C
L	1	m
nodes_x	10	-
nodes_t	100	-
D	0.5	-
Θ	0	-
$E_{propuesto}$	0.001	-

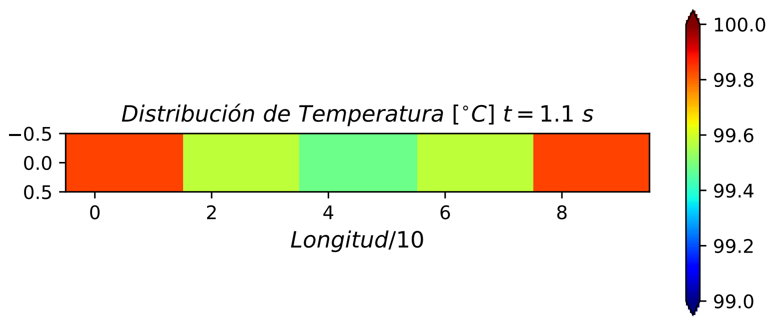
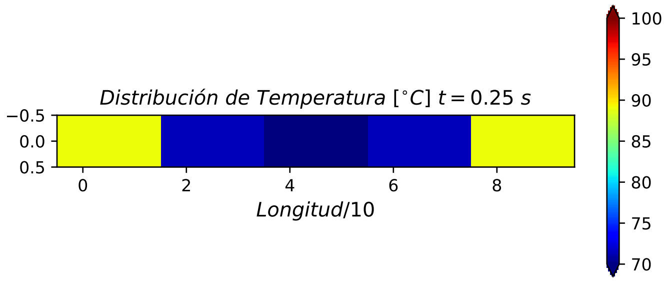
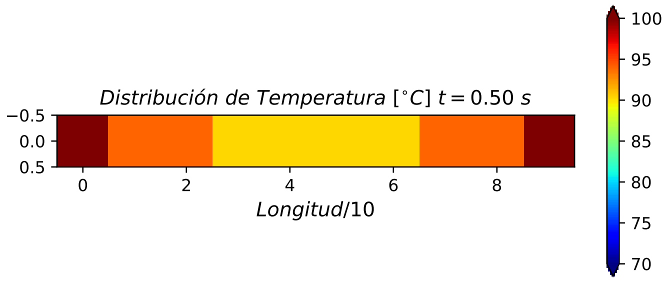


Figura 5. Método Explícito, $t = 1{,}1$ (tiempo aproximado de llegada a estado estable)

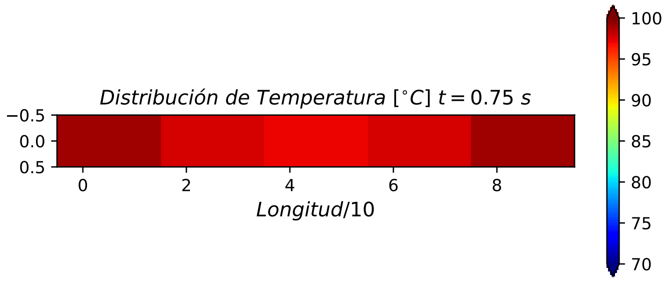
En la Figura 5 la resolución se ve afectada por la selección de nodos espaciales (aquí 10 nodos), lo que incrementa el error y hace menos suave la solución comparada con el esquema implícito.



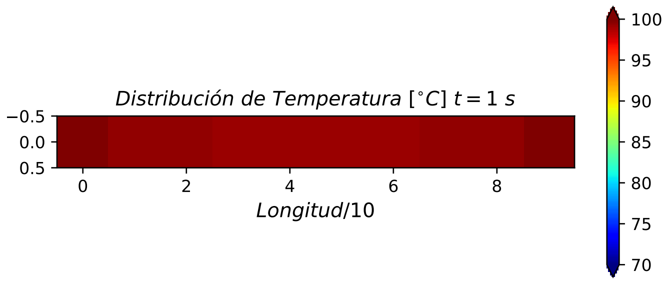
(a) Distribución de T en $t = 0{,}25$ s



(b) Distribución de T en $t = 0{,}5$ s

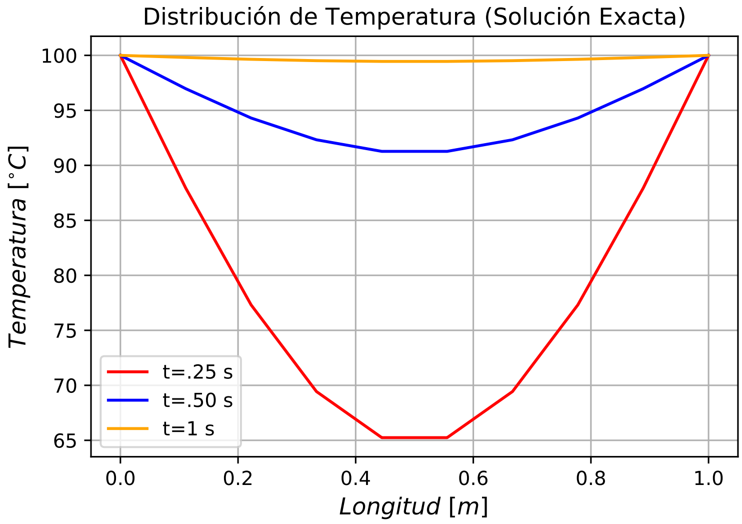


(c) Distribución de T en $t = 0{,}75$ s

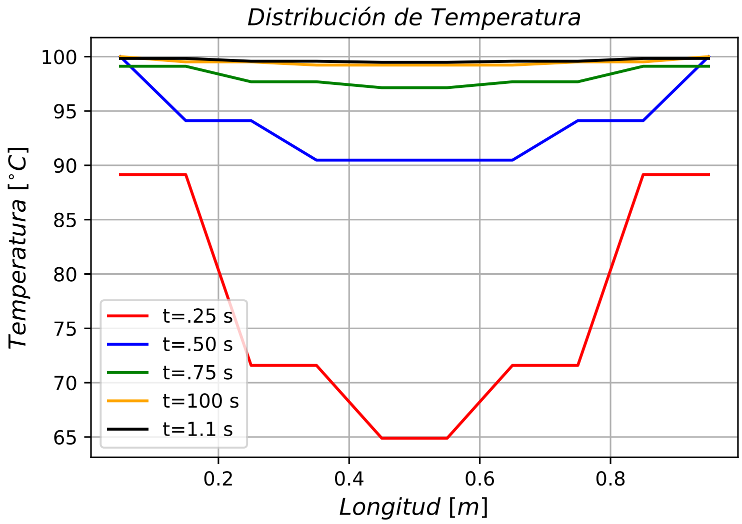


(d) Distribución de T en $t = 1$ s

Figura 3: Método Explícito: $0 \leq t \leq 1$.



(a) Distribución de T (solución exacta)



(b) Distribución de T (numérica, explícita)

Figura 4: Comparación solución exacta vs numérica (explícito).

Cuadro 4: Errores (explícito) — Error promedio

time	**Error Promedio**
$t = 0{,}25\ s$	5.12 %
$t = 0{,}50\ s$	3.67 %
$t = 1{,}0\ s$	1.87 %

5. Conclusiones

En este trabajo se presentó una comparativa entre el método implícito y el método explícito para la solución de un problema transitorio tratado con Volumen Finito. Observando los resultados, se concluye que el método

más conveniente es el método implícito, ya que es incondicionalmente estable frente al esquema explícito. Sin embargo, el implícito implica resolver un sistema de ecuaciones en cada paso temporal, lo que demanda más operaciones.
El porcentaje de error de cada método frente a la solución exacta depende en gran medida del número de nodos espaciales seleccionados: a mayor número de elementos es más fácil captar variaciones y reducir el error.
Los resultados muestran que el sistema alcanza el estado estacionario en aproximadamente 1,1 s; las comparaciones en $t = 0{,}25, 0{,}50, 0{,}75$ y 1,0 s permiten verificar la evolución hacia la relajación térmica. El valor final reportado (variación aproximada de 0,003 °C en $t = 1{,}2$ s) concuerda con la estabilidad observada en las simulaciones.

Referencias

[1] H. Versteeg and W. Malalasekera, “An introduction to computational fluid dynamics,” *Finite Volume Method, Essex, Longman Scientific & Technical*, 1995.

Appendix I: Solution Codes

```
print('\033[H\033[J')
import numpy as np
import matplotlib.pyplot as plt
from gauss_seidel import gauss_seidel
n=100 #orden del sistema (numero de nodos)
n_time=110 #orden del tiempo (numero de nodos temporales)
time=1.1 #segundos
error=.001 #error propuesto

A=np.zeros(((n,n)),float)
X=np.zeros(((n)),float)
B=np.zeros(((n)),float)
Xc=np.zeros(((n)),float)
E=np.zeros(((n)),float)
Ln=np.zeros(((n)),float)
B0=np.zeros(((n)),float)

#vectores para gracicar cada 25% de avance del tiempo de solucion
X25=np.zeros(((n)),float)
X50=np.zeros(((n)),float)
X75=np.zeros(((n)),float)
X100=np.zeros(((n)),float)
X110=np.zeros(((n)),float)


theta=1
T_a=100
T_b=100
L=1
dx=L/n
dt=time/n_time
De=Dw=.5
a=dx/dt
ae=De/dx
aw=Dw/dx

for j in range(0,n_time):
   for i in range(0,n):
       aux1=i+1
       aux2=i-1
       Ln[i]=aux1*dx-(dx/2)
       if i==0:
          A[i,i]=a+(theta)*(ae+2*aw)
          A[i,aux1]=-theta*ae
          B[i]=(a+(1-theta)*(-ae-2*aw))*B0[i]+(ae*(1-theta))*B0[aux1]+(2*aw*(1-theta))*T_a
    +(2*aw*theta)*T_a
       if i==(n-1):
          A[i,i]=a+(theta)*(2*ae+aw)
          A[i,aux2]=-theta*aw
          B[i]=(a+(1-theta)*(-2*ae-aw))*B0[i]+(ae*(1-theta))*B0[aux2]+(2*ae*(1-theta))*T_b
    +(2*ae*theta)*T_b
```

```
        if i!=0 and i!=n-1:
            A[i,i]=a+theta*(ae+aw)
            A[i,aux1]=-theta*ae
            A[i,aux2]=-theta*aw
            B[i]=(a+(1-theta)*(-ae-aw))*B0[i]+(ae*(1-theta))*B0[aux1]+(aw*(1-theta))*B0[aux2]
    AA=np.linalg.inv(A)
    sol=np.dot(AA,B)
    #X=gauss_seidel(A,B,X,Xc,E,n,error,np)
    B0=sol
    if j==25:
        X25=sol
    if j==50:
        X50=sol
    if j==75:
        X75=sol
    if j==100:
        X100=sol
    if j==109:
        X110=sol

plt.figure()
plt.plot(Ln,X25,label = "t=.25 s", color = 'red')
plt.hold(True)
plt.plot(Ln,X50,label = "t=.50 s", color = 'blue')
plt.plot(Ln,X75, label = "t=.75 s", color = 'green')
plt.plot(Ln,X100, label = "t=1 s", color = 'orange')
plt.plot(Ln,X110, label = "t=1.1 s", color = 'black')
plt.grid(True)
plt.legend(loc="lower left")
plt.xlabel(r"$Longitud\ [m]$", fontsize = 12, color ='black')
plt.ylabel(r"$Temperatura\ [^{\circ} C]$", fontsize = 12, color = 'black')
plt.title(r"$Distribucion\ de\ Temperatura$",fontsize = 12, color = 'black', verticalalignment = 'baseline', horizontalalignment = 'center')
plt.savefig('Grafica1.pdf')
plt.show()

espesor=int(n/10)
X25_1=np.zeros(((espesor*n)),float)
X50_1=np.zeros(((espesor*n)),float)
X75_1=np.zeros(((espesor*n)),float)
X100_1=np.zeros(((espesor*n)),float)
X110_1=np.zeros(((espesor*n)),float)
for k in range(1,espesor+1):
    for l in range(0,(espesor+1)*n):
        if l<k*n and l>=(k-1)*n:
            X25_1[l]=X25[l-(k-1)*n]
            X50_1[l]=X50[l-(k-1)*n]
            X75_1[l]=X75[l-(k-1)*n]
            X100_1[l]=X100[l-(k-1)*n]
            X110_1[l]=X100[l-(k-1)*n]

plt.figure(1)
arr = X25.reshape((1,n))
fig = plt.figure(figsize=(15, 3))
im1 = plt.imshow(arr, cmap='jet')
plt.colorbar(extend='both')
plt.clim(70, 100);
plt.xlabel(r"$Longitud\ [\%]$", fontsize = 12, color = 'black')
plt.title(r"$Distribucion\ de\ Temperatura\ [^{\circ} C]\ t=0.25\ s$",fontsize = 12, color = 'black', verticalalignment = 'baseline', horizontalalignment = 'center')
```

```
plt.savefig('Grafica2_a.pdf')

plt.figure(2)
arr2 = X50.reshape((1,n))
fig = plt.figure(figsize=(15, 3))
im2 = plt.imshow(arr2, cmap='jet')
plt.colorbar(extend='both')
plt.clim(70, 100);
plt.xlabel(r"$Longitud\ [\%]$", fontsize = 12, color = 'black')
plt.title(r"$Distribucion\ de\ Temperatura\ [^{\circ} C]\ t=0.50\ s$",fontsize = 12, color = 'black', verticalalignment = 'baseline', horizontalalignment = 'center')
plt.savefig('Grafica2_b.pdf')

plt.figure(3)
arr3 = X75.reshape((1,n))
fig = plt.figure(figsize=(15, 3))
im3 = plt.imshow(arr3, cmap='jet')
plt.colorbar(extend='both')
plt.clim(70, 100);
plt.xlabel(r"$Longitud\ [\%]$", fontsize = 12, color = 'black')
plt.title(r"$Distribucion\ de\ Temperatura\ [^{\circ} C]\ t=0.75\ s$",fontsize = 12, color = 'black', verticalalignment = 'baseline', horizontalalignment = 'center')
plt.savefig('Grafica2_c.pdf')

plt.figure(4)
arr4 = X100.reshape((1,n))
fig = plt.figure(figsize=(15, 3))
im4 = plt.imshow(arr4, cmap='jet')
plt.colorbar(extend='both')
plt.clim(70, 100);
plt.xlabel(r"$Longitud\ [\%]$", fontsize = 12, color = 'black')
plt.title(r"$Distribucion\ de\ Temperatura\ [^{\circ} C]\ t=1\ s$",fontsize = 12, color = 'black', verticalalignment = 'baseline', horizontalalignment = 'center')
plt.savefig('Grafica2_d.pdf')

plt.figure(4)
arr5 = X110.reshape((1,n))
fig = plt.figure(figsize=(15, 3))
im4 = plt.imshow(arr5, cmap='jet')
plt.colorbar(extend='both')
plt.clim(99, 100);
plt.xlabel(r"$Longitud\ [\%]$", fontsize = 12, color = 'black')
plt.title(r"$Distribucion\ de\ Temperatura\ [^{\circ} C]\ t=1.1\ s$",fontsize = 12, color = 'black', verticalalignment = 'baseline', horizontalalignment = 'center')
plt.savefig('Grafica2_e.pdf')
```