INFORME LAB 06 – FC

Link Codigo:
https://colab.research.google.com/drive/19avHU65PRlZSqCaPMQ1HxC1BMOsfzUSk?usp=sharing

Introducción

En este informe se aborda la solución numérica de las ecuaciones de calor y de onda utilizando el método de diferencias finitas. Estas ecuaciones diferenciales parciales son fundamentales en la modelación de fenómenos físicos de difusión de calor y propagación de ondas, con aplicaciones en diversos campos de la ingeniería y la física. Mediante el uso de Python y la implementación de los esquemas de diferencias finitas, buscamos aproximar las soluciones de estas ecuaciones en un dominio espacial y temporal dado, bajo condiciones específicas de frontera e iniciales. Los resultados se visualizan mediante gráficas 2D y 3D para interpretar el comportamiento de las soluciones a lo largo del tiempo.

Marco Teórico

Ecuación de Calor

La ecuación de calor en una dimensión describe la distribución de temperatura u(x,t)en un medio a lo largo del tiempo t en función de la posición x. Su forma general es:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

donde α es el coeficiente de difusión térmica, una constante que depende de las propiedades del material. La ecuación de calor es un modelo de difusión que describe cómo se transfiere el calor en un medio a través de la conducción.

Para resolver esta ecuación numéricamente, aplicamos el método de diferencias finitas, que discretiza las derivadas en el espacio y el tiempo, resultando en la siguiente ecuación en diferencias:

$$u_{i,j+1} = (1 - 2r)u_{i,j} + r(u_{i+1,j} + u_{i-1,j})$$

donde $r = \alpha \frac{\Delta t}{\Delta x^2}$ es un parámetro de estabilidad que depende del tamaño de los pasos temporal (Δt) y espacial (Δx). Para garantizar la estabilidad del método, es necesario que r≤0.5

Ecuación de Onda

La ecuación de onda en una dimensión describe la propagación de ondas en un medio y se expresa como:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

donde c es la velocidad de la onda en el medio. Esta ecuación es crucial en la modelación de fenómenos donde una perturbación se propaga a través de un medio, como en el caso de ondas acústicas, sísmicas o electromagnéticas.

La fórmula de diferencias finitas utilizada para aproximar la solución es:

$$u_{i,j+1} = 2(1 - r^2)u_{i,j} + r^2(u_{i+1,j} + u_{i-1,j}) - u_{i,j-1}$$

donde $r = \frac{c\Delta t}{\Delta x}$. La estabilidad de este método depende de que r≤1, de lo contrario, se generan inestabilidades numéricas que afectan la precisión de la solución.

Método de Diferencias Finitas

El método de diferencias finitas es una técnica numérica que reemplaza las derivadas continuas por diferencias finitas, permitiendo resolver ecuaciones diferenciales parciales de forma discreta. Este método implica descomponer el dominio en una malla de puntos y calcular las aproximaciones de las derivadas en cada punto de la malla.

Desarrollo

A continuación, se presentan los fragmentos más importantes de cada código y una explicación sobre su funcionamiento.

Código 1: Ecuación de Calor

1. **Definición de Parámetros**:

```
# Parámetros del problema
L = 1.0    # Longitud del dominio
T = 0.1    # Tiempo total
nx = 50    # Número de divisiones espaciales
nt = 500   # Número de pasos de tiempo
alpha = 1.0   # Coeficiente de difusión térmica

dx = L / (nx - 1)  # Paso espacial
dt = T / nt         # Paso temporal
r = alpha * dt / dx**2  # Parámetro r
```

En este fragmento, se definen los parámetros del problema: longitud del dominio, tiempo total de simulación, número de divisiones espaciales y temporales, y el coeficiente de difusión térmica α. También se calcula el parámetro de estabilidad r, que debe ser menor o igual a 0.5 para garantizar la estabilidad.

2. **Condición Inicial**:

```
u[:, 0] = np.sin(np.pi * np.linspace(0, L, nx))  # Condición inicial
```

La condición inicial establece una distribución de temperatura en forma de onda senoidal en el dominio.

3. **Bucle de Diferencias Finitas**:

```python
# Bucle de tiempo para resolver la ecuación de calor
for j in range(0, nt - 1):
    for i in range(1, nx - 1):
        u[i, j + 1] = (1 - 2 * r) * u[i, j] + r * (u[i + 1, j] + u[i - 1, j])
```

Este bucle aplica la fórmula de diferencias finitas a cada punto de la malla en cada paso de tiempo, actualizando la temperatura en el dominio según la ecuación de calor.

4. **Visualización en 2D**:

```python
plt.plot(np.linspace(0, L, nx), u[:, j], label=f't = {j * dt:.2f}s')
```

Este fragmento genera una gráfica que muestra la distribución de temperatura en distintos momentos para analizar cómo se difunde el calor en el tiempo.

Código 2: Ecuación de Onda

1. **Definición de Parámetros y Condición de Estabilidad**:

```python
# Parámetros
L = 1.0            # Longitud del dominio espacial
T = 1.0            # Tiempo total de simulación
nx = 100           # Número de puntos espaciales
nt = 200           # Número de puntos de tiempo
dx = L / (nx - 1)  # Paso espacial
dt = T / (nt - 1)  # Paso de tiempo
c = 1.0            # Velocidad de la onda
r = (c * dt / dx)  # Parámetro de estabilidad (debe ser <= 1 para estabilidad)
```

Aquí, se define la longitud del dominio, el tiempo total de simulación y el número de divisiones espaciales y temporales. El parámetro rrr se calcula para satisfacer la condición de estabilidad de la ecuación de onda.

2. **Condiciones Iniciales y de Frontera**:

```python
# Condición inicial en t=0 (ejemplo: onda centrada en el dominio)
u[:, 0] = np.exp(-100 * (x - L/2)**2)
u[:, 1] = u[:, 0]          # Inicialmente asumimos que la onda no cambia
```

La condición inicial establece una onda centrada en el dominio con amplitud máxima en el centro. Los bordes están fijos a cero, simulando una frontera rígida.

3. **Bucle de Diferencias Finitas**:

```python
for j in range(1, nt - 1):
    for i in range(1, nx - 1):
        u[i, j+1] = 2 * (1 - r**2) * u[i, j] + r**2 * (u[i+1, j] + u[i-1, j]) - u[i, j-1]
```

Este bucle calcula el desplazamiento de la onda en cada paso de tiempo usando la fórmula de diferencias finitas de la ecuación de onda.

4. **Visualización en 2D**:

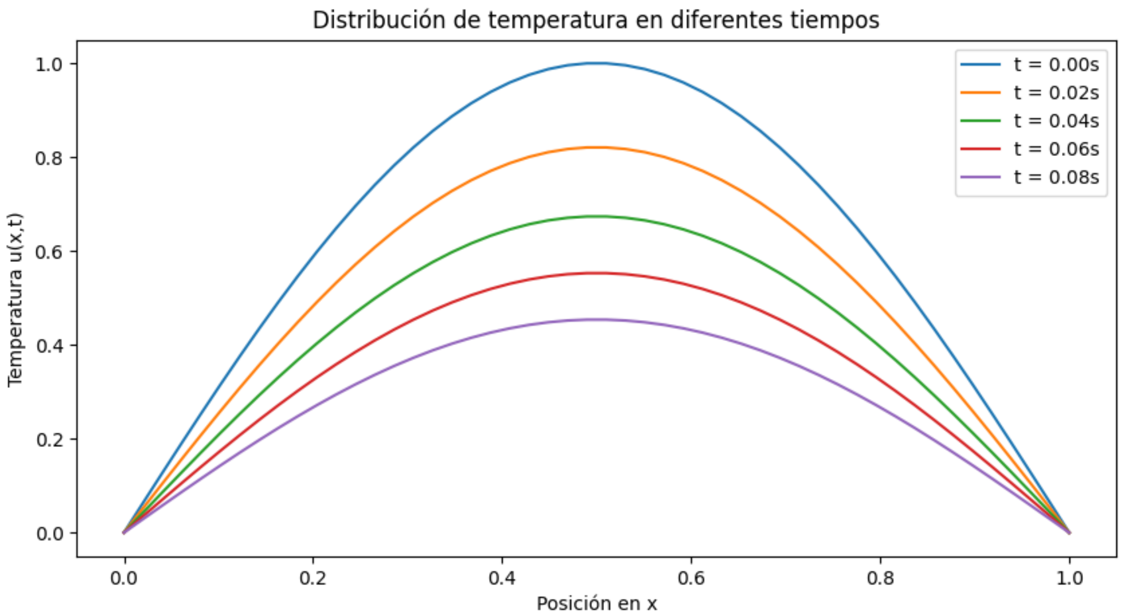
```python
plt.plot(x, u[:, j], label=f"t = {j * dt:.2f}")
```

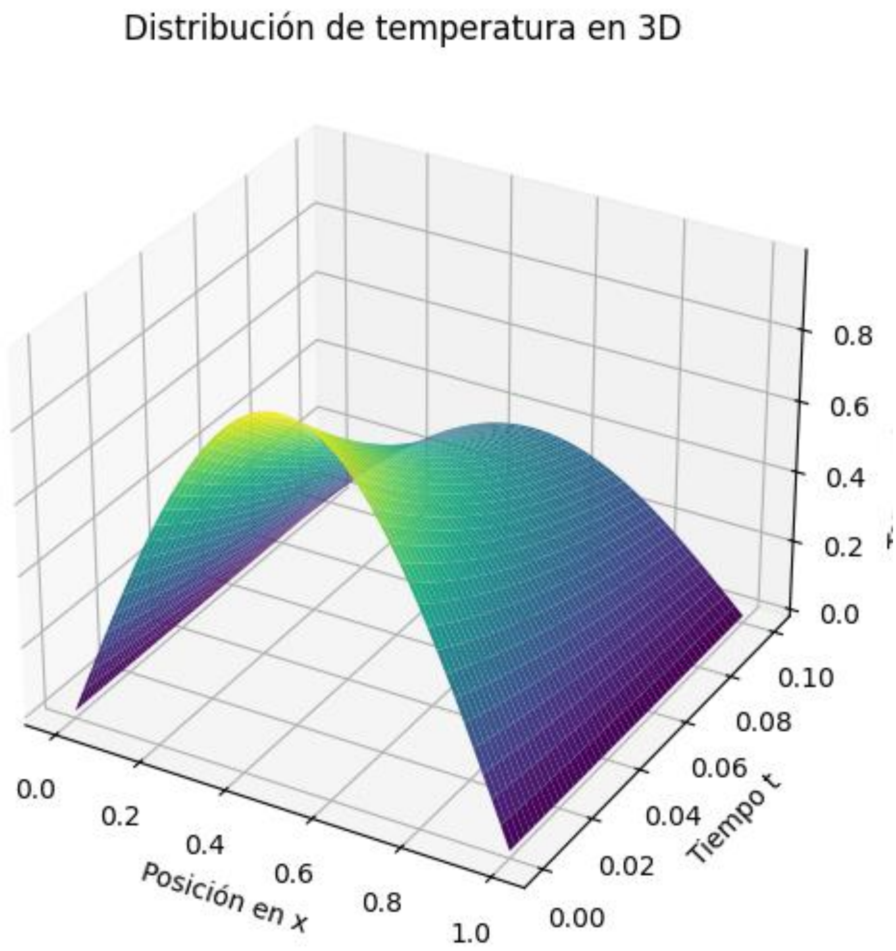
Esta línea grafica la posición de la onda en distintos momentos, permitiendo observar su propagación a lo largo del dominio.

Resultados

Los resultados se visualizan en gráficas 2D y 3D, mostrando la evolución de las soluciones en el tiempo y en el espacio.

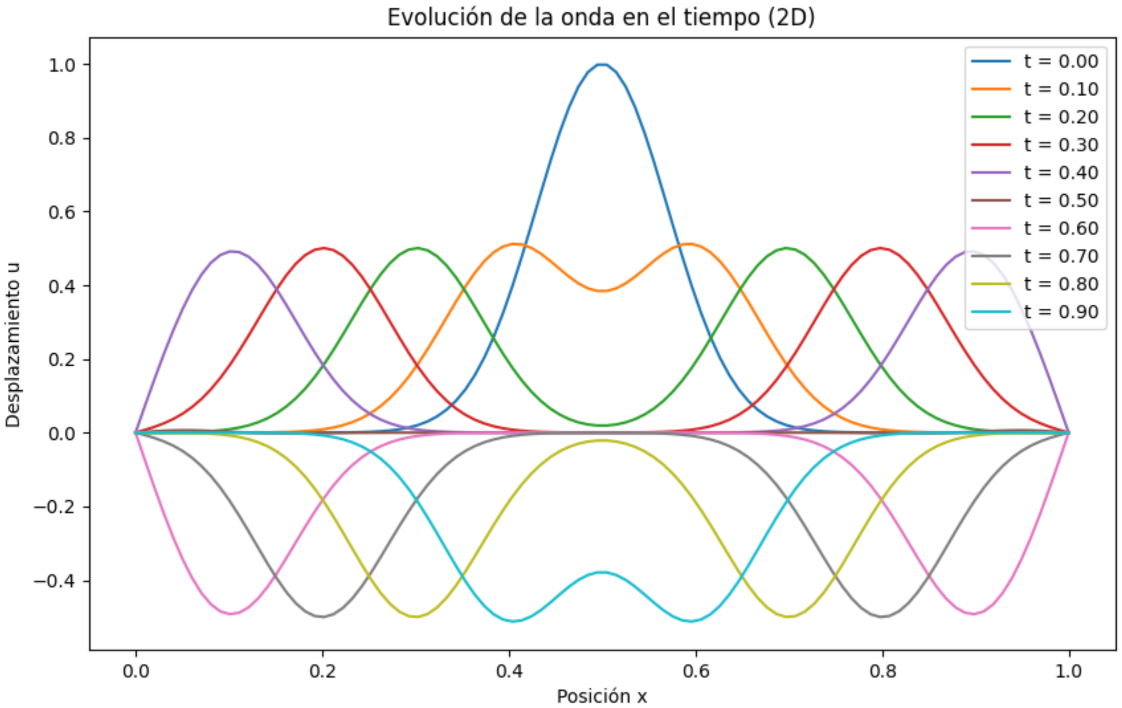
1. **Ecuación de Calor**:
   - La gráfica 2D muestra cómo la temperatura se difunde en el dominio, partiendo de una condición inicial senoidal. La gráfica 3D ilustra la distribución de temperatura en el tiempo y el espacio, evidenciando la disipación del calor en el tiempo.

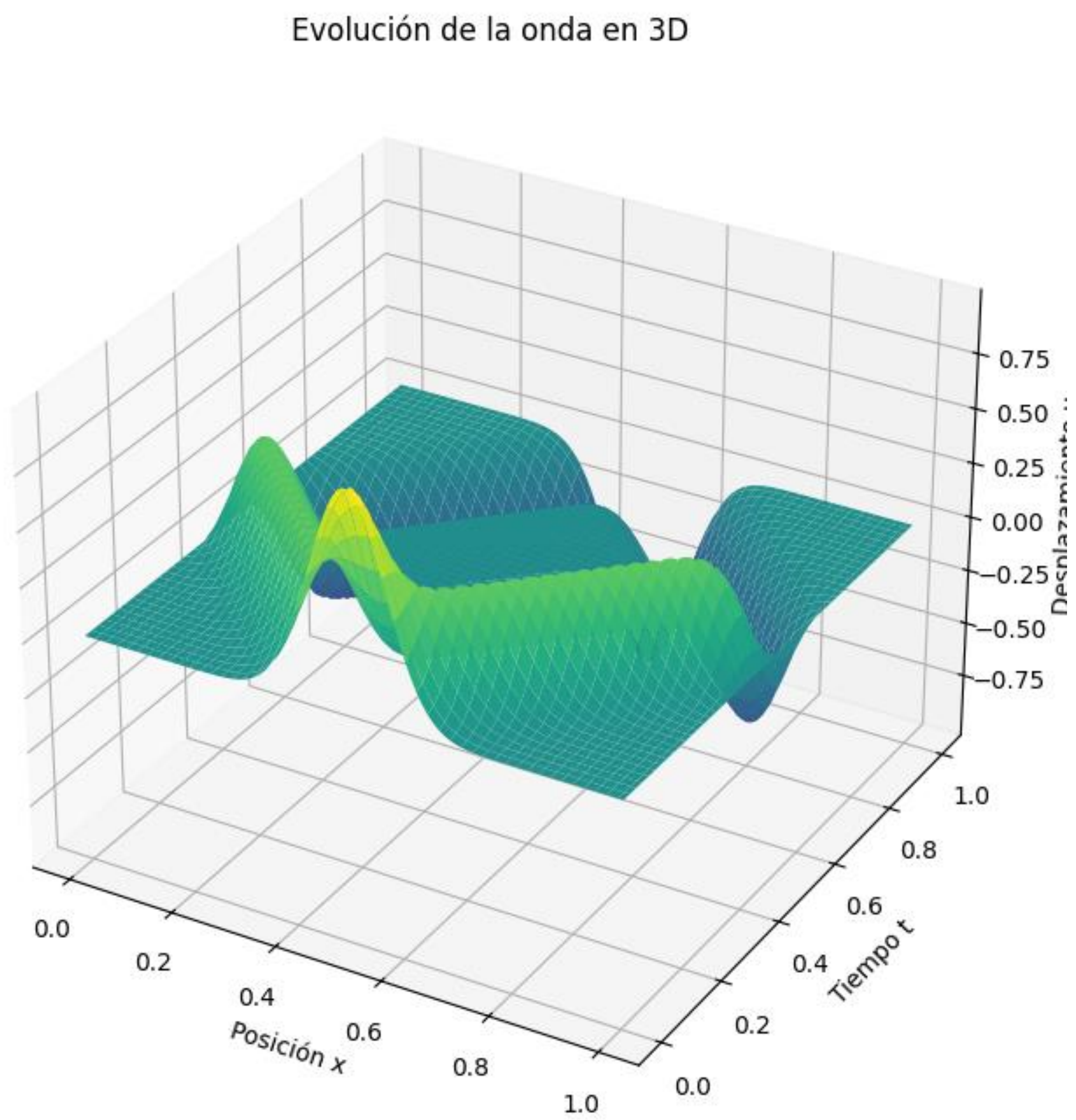




2. **Ecuación de Onda**:

   - La gráfica 2D de la evolución de la onda permite observar cómo la perturbación se propaga a lo largo del dominio. La gráfica 3D muestra la evolución de la onda en el tiempo, destacando la naturaleza de propagación y la influencia de las fronteras fijas.


Evolución de la onda en el tiempo (2D)
t = 0.00
t = 0.10
t = 0.20
t = 0.30
t = 0.40
t = 0.50
t = 0.60
t = 0.70
t = 0.80
t = 0.90
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
Desplazamiento u
0.0
0.2
0.4
0.6
0.8
1.0
Posición x



Conclusiones

En este informe, se ha demostrado la aplicación del método de diferencias finitas para resolver numéricamente las ecuaciones de calor y de onda en una dimensión. Los resultados muestran que este método es eficaz para modelar la difusión de calor y la propagación de ondas en un dominio espacial y temporal discretizado. Se evidenció que los parámetros de estabilidad son críticos para asegurar la precisión y estabilidad de la solución, especialmente en el caso de la ecuación de onda. Este análisis puede aplicarse en diversos problemas de ingeniería y física donde se requiera modelar fenómenos de difusión o propagación de ondas.