



Universidad Técnica Federico Santa María

IP468 - DINÁMICA DE FLUIDOS COMPUTACIONAL

PHD. ING. ROMAIN GERS

TAREA DE VOLÚMENES FINITOS 2023:

"Calentamiento de un flujo plano"

Autor: Javiera Espinoza M.¹

22 de noviembre, 2023

¹Correo: javieraespinozam@usm.cl - Rol: 201941036-9

Índice

Índice de Figuras	1
Índice de Tablas	2
1 Marco Teórico	3
1.1 Advección en Dinámica de Fluidos	3
1.2 Difusión en Dinámica de Fluidos	3
1.3 Método de Volúmenes Finitos (FVM)	3
2 Presentación del problema	3
3 Discretización de las ecuaciones	4
3.1 Método de resolución a utilizar	4
3.2 Ecuaciones discretizadas	5
3.3 Malla regular	5
3.3.1 Esquema explícito en el tiempo, UDS en la advección y CDS en la difusión	6
3.4 Malla irregular	7
3.4.1 Esquema explícito en el tiempo, UDS en la advección y CDS en la difusión	7
4 Condiciones de borde	8
5 Perfiles iniciales	9
5.1 Perfil inicial de velocidades	9
5.2 Perfil inicial de temperaturas	9
6 Resultados	10
7 Validación	12
7.1 Advección pura en x	13
7.2 Difusión pura en x	14
7.3 Difusión pura en y	16
7.4 Permanencia del régimen	17
7.4.1 Malla regular	17
7.4.2 Malla irregular	18
7.5 Escalas de tiempo asociadas	18
8 Conclusiones	19

Índice de Figuras

1 Representación del flujo.	4
2 Método a utilizar	5
3 Malla regular.	6
4 Malla irregular.	7

5	Perfil de velocidad.	9
6	Perfil de Temperatura inicial.	10
7	Solución para el tiempo de convergencia.	11
8	Solución para el tiempo de convergencia.	12
9	Mallado unidimensional para la advección.	13
10	Solución para $t = 500$ [s].	14
11	Solución para $t = 500$ [s].	15
12	Solución para $t = 500$ [s].	16
13	Transporte de temperatura en malla regular.	17
14	Transporte de temperatura en malla irregular.	18

Índice de Tablas

1	Parámetros generales asociados	10
---	--	----

1 Marco Teórico

El estudio del transporte de temperatura ha experimentado un notable crecimiento en las últimas décadas, impulsado por la necesidad de comprender y prever los cambios térmicos para optimizar la eficiencia en diversos procesos. Dos fenómenos cruciales en este campo son la advección y la difusión, los cuales desempeñan un papel fundamental en la propagación de las propiedades térmicas de un fluido en su entorno. A pesar de la relevancia de estos fenómenos, su resolución matemática puede ser tanto compleja como computacionalmente exigente. En por ello que se han desarrollado métodos avanzados de resolución, como lo es el método de volúmenes finitos. Este método trabaja con el dominio de manera eficiente, permitiendo abordar problemas de transporte térmico de manera rápida y efectiva, y contribuyendo así a mejorar la comprensión y el rendimiento de los procesos asociados.

1.1 Advección en Dinámica de Fluidos

La advección es un fenómeno clave en la Dinámica de Fluidos que describe el transporte de propiedades como masa, cantidad de movimiento o energía debido al movimiento del fluido. En el contexto de CFD, la advección se modela mediante ecuaciones diferenciales parciales (EDP) que representan la variación temporal y espacial de estas propiedades en un dominio fluido. La ecuación de advección modela cómo un campo escalar o vectorial es transportado por el flujo.

1.2 Difusión en Dinámica de Fluidos

La difusión representa la dispersión de propiedades en el espacio debido a gradientes de concentración o temperatura. En CFD, la difusión se modela mediante ecuaciones diferenciales que describen la variación espacial de la propiedad difundida.

1.3 Método de Volúmenes Finitos (FVM)

El Método de Volúmenes Finitos es una técnica numérica ampliamente utilizada para resolver ecuaciones diferenciales en forma integral, razón por la cual se hará uso de ella en el presente proyecto. Para su uso, el dominio se divide en celdas o volúmenes finitos, y las ecuaciones diferenciales se integran sobre estos volúmenes.

2 Presentación del problema

Se desea estudiar mediante el método de Volúmenes finitos el calentamiento de un flujo plano representado en la siguiente ilustración:

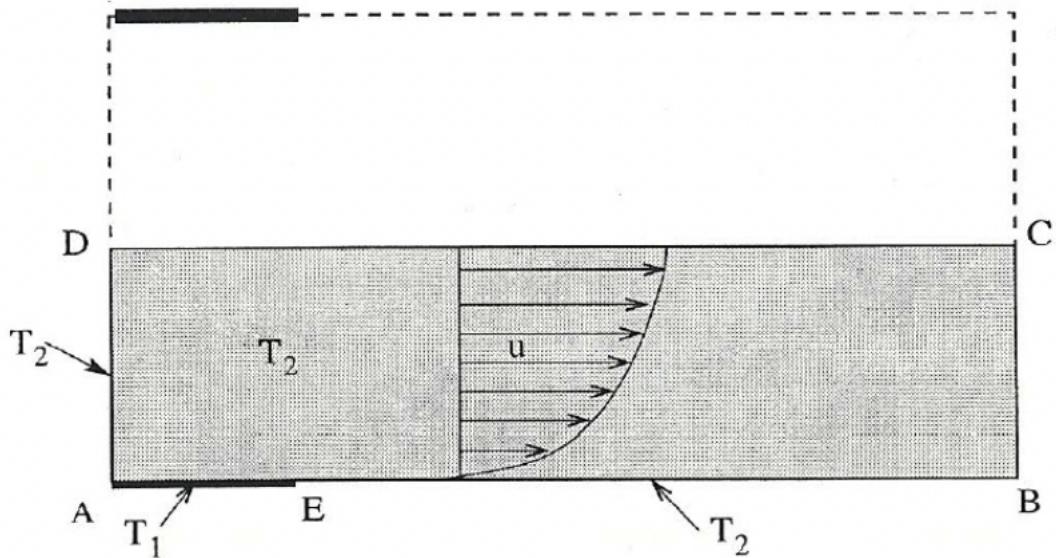


Figura 1: Representación del flujo.

El flujo es simétrico con relación al eje DC. Se puede restringir el dominio de cálculo al volumen grisado ABCD. La parte AE de la pared AB es calentada y mantenida a una temperatura constante.

La ecuación que gobierna el transporte advectivo y difusivo de la temperatura T se escribe :

$$\frac{\partial T}{\partial t} + \nabla(\vec{v}T - D\nabla T) = 0 \quad (1)$$

donde \vec{v} es el vector velocidad, D el coeficiente de difusividad térmica (isotropa), A es el origen del sistema de referencia, AB una parte del eje de las abscisas x, AD una parte del eje de las ordenadas y. Se impone un campo de velocidad estacionario, independiente de x pero dependiente de y. El perfil se define por la siguiente ecuación:

$$v(y) = v_0(1 - \frac{y - l(y)^2}{l_y^2}) \quad (2)$$

Donde $l_y = AD$

3 Discretización de las ecuaciones

3.1 Método de resolución a utilizar

El método de volúmenes finitos es una técnica numérica utilizada para resolver ecuaciones diferenciales, en el cual se divide el dominio mediante un mallado de carácter regular o irregular. Este mallado permite que, mediante la información de los nodos vecinos, se calcule la solución en cada volúmen del dominio, lo cual se puede ver ilustrado en la siguiente imagen:

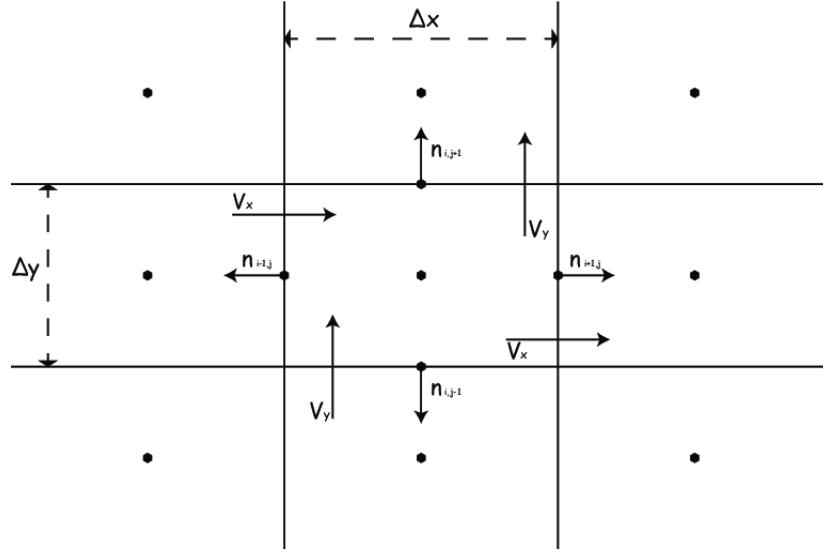


Figura 2: Método a utilizar

Para facilitar la comprensión, se utilizará un sistema de referencia dado por la rosa de los vientos estándar, en la cual los subíndices O, N, E, S representan el oeste, norte, este y sur respectivamente.

3.2 Ecuaciones dicretizadas

Sea la advección en dos dimensiones:

$$(\vec{F}_a)_k = (\vec{F}_{ax} + \vec{F}_{ay})_k = (\rho\phi\vec{u})_k + (\rho\phi\vec{v})_k \quad (3)$$

De la misma manera, sea la difusión:

$$(\vec{F}_d)_k = (\vec{F}_{dx} + \vec{F}_{dy})_k = -(\rho D \frac{\partial \phi}{\partial x} \vec{e}_x)_k - (\rho D \frac{\partial \phi}{\partial y} \vec{e}_y)_k \quad (4)$$

Suponiendo $\rho = \text{constante}$, la ecuación Advección - Difusión en dos dimensiones es:

$$\begin{aligned} \rho \frac{\partial \phi}{\partial t} V_{i,j} &= (\rho u \Delta y)_{i+\frac{1}{2},j} - (\rho u \Delta y)_{i-\frac{1}{2},j} \\ &+ (\rho v \Delta x)_{i,j+\frac{1}{2}} - (\rho v \Delta x)_{i,j-\frac{1}{2}} \\ &- (\rho D \frac{\partial \phi}{\partial x} \Delta y)_{i+\frac{1}{2},j} + (\rho D \frac{\partial \phi}{\partial x} \Delta y)_{i-\frac{1}{2},j} \\ &- (\rho D \frac{\partial \phi}{\partial y} \Delta x)_{i,j+\frac{1}{2}} + (\rho D \frac{\partial \phi}{\partial y} \Delta x)_{i,j-\frac{1}{2}} \end{aligned} \quad (5)$$

3.3 Malla regular

Sea la siguiente malla regular y ortogonal, la cual se compone de veinticinco volúmenes en el eje x, y 30 volúmenes en el eje y, todos equiespaciados entre si:

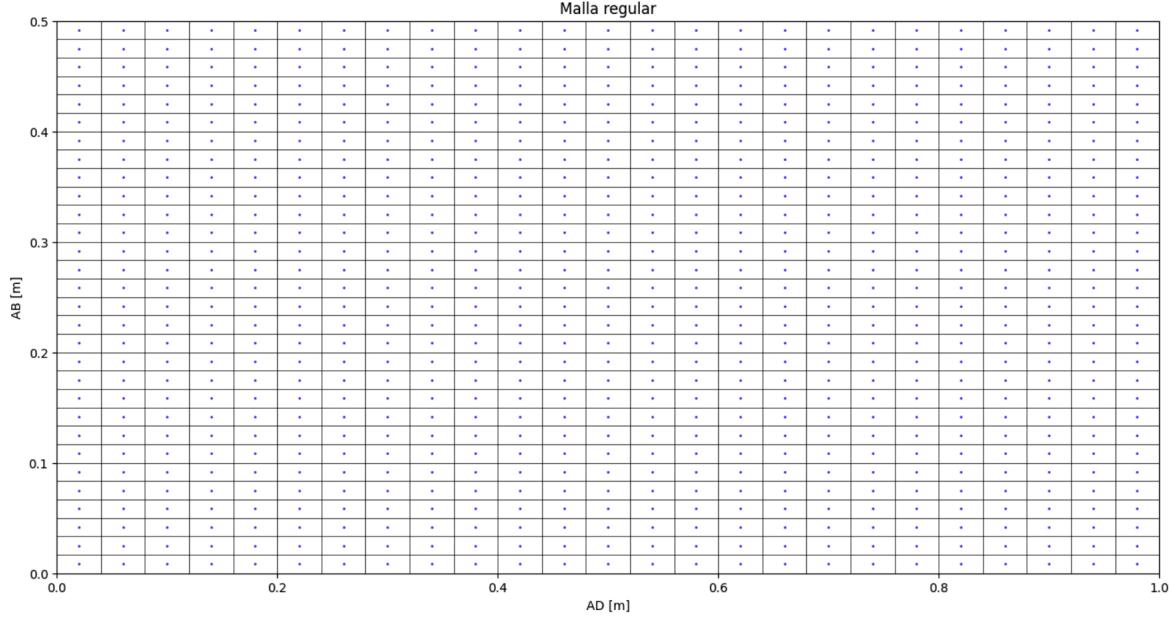


Figura 3: Malla regular.

Haciendo uso de la ecuación de Advección - Difusión anteriormente descrita se tiene:

$$\frac{\partial T}{\partial t} V_{i,j} = -(u \Delta y T)_{i+\frac{1}{2},j} + (u \Delta y T)_{i-\frac{1}{2},j} + D \left(\left(\frac{\partial T}{\partial x} \Delta y \right)_{i+\frac{1}{2},j} - \left(\frac{\partial T}{\partial x} \Delta y \right)_{i-\frac{1}{2},j} + \left(\frac{\partial T}{\partial y} \Delta x \right)_{i,j+\frac{1}{2}} - \left(\frac{\partial T}{\partial y} \Delta x \right)_{i,j-\frac{1}{2}} \right) \quad (6)$$

3.3.1 Esquema explícito en el tiempo, UDS en la advección y CDS en la difusión

Utilizando un esquema explícito en el tiempo, atrasado en la advección (UDS) y centrado en el espacio (CDS) en la difusión, se tienen los siguientes términos:

$$\frac{\partial T}{\partial t} V_{i,j} = \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \Delta x \Delta y \quad (7)$$

$$-(u \Delta y T)_{i+\frac{1}{2},j} = -(u(y)(T_C)) \quad (8)$$

$$-(u \Delta y T)_{i-\frac{1}{2},j} = -(u(y)T_O) \quad (9)$$

$$D \left(\frac{\partial T}{\partial x} \Delta y \right)_{i+\frac{1}{2},j} = D \left(\frac{T_E - T_C}{\Delta x} \right) \Delta y \quad (10)$$

$$D \left(\frac{\partial T}{\partial x} \Delta y \right)_{i-\frac{1}{2},j} = D \left(\frac{T_E - T_C}{\Delta x} \right) \Delta y \quad (11)$$

$$D \left(\frac{\partial T}{\partial x} \right) \Delta y_{i-\frac{1}{2},j} = D \left(\frac{T_C - T_O}{\Delta x} \right) \Delta y \quad (12)$$

$$D\left(\frac{\partial T}{\partial x}\right)\Delta x)_{i,j+\frac{1}{2}} = D\left(\frac{T_N - T_C}{\Delta y}\right)\Delta x \quad (13)$$

$$D\left(\frac{\partial T}{\partial x}\right)\Delta x)_{i,j-\frac{1}{2}} = D\left(\frac{T_C - T_S}{\Delta y}\right)\Delta x \quad (14)$$

Reemplazando los términos en la ecuación Advección - Difusión se tiene:

$$\begin{aligned} T_{i,j}^{n+1} = & -\frac{\Delta t}{\Delta x \Delta y} (\Delta y (u T_P - u T_O) \\ & + D\left(\frac{T_E - T_C}{\Delta x}\right)\Delta y - \frac{T_C - T_O}{\Delta x}\Delta y + \frac{T_N - T_C}{\Delta y}\Delta x - \frac{T_C - T_S}{\Delta y}\Delta x) + T_{i,j}^n \end{aligned} \quad (15)$$

3.4 Malla irregular

3.4.1 Esquema explícito en el tiempo, UDS en la advección y CDS en la difusión

Haciendo uso de la siguiente malla ortogonal, la cual posee 25 volúmenes en el eje x, y 30 volúmenes en el eje y, irregulares dependientes de una progresión producida por una productoria e iniciada en el eje de coordenadas $(x, y) = (0,0)$, se tiene:

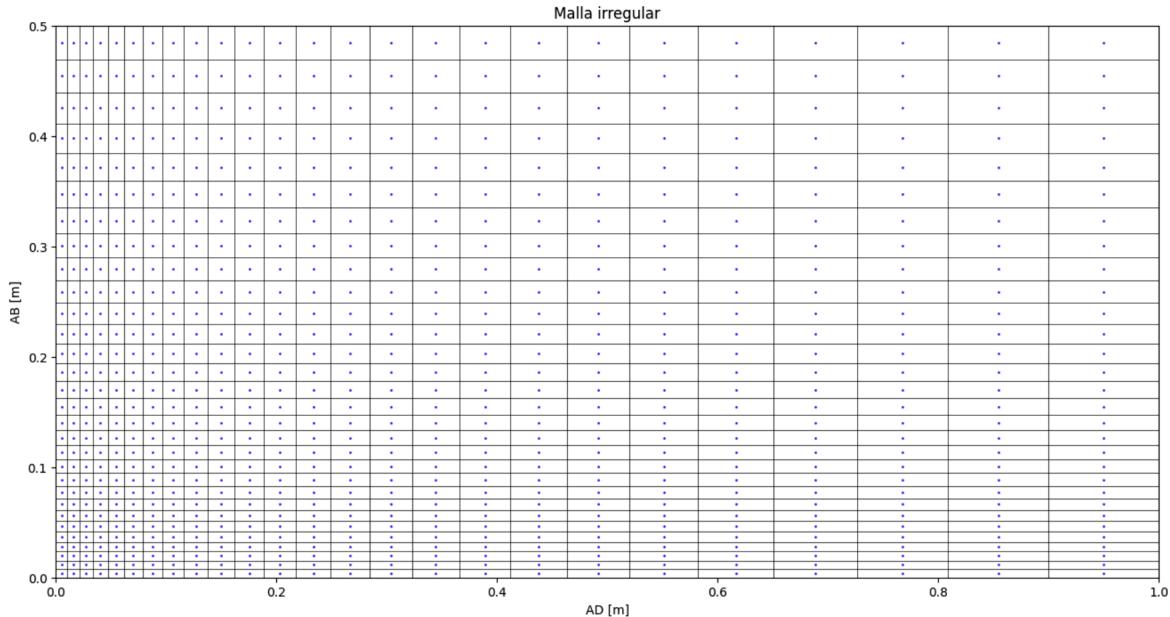


Figura 4: Malla irregular.

Haciendo uso de las ecuación de Advección - Difusión anteriormente descrita se tiene:

$$\frac{\partial T}{\partial t} V_{i,j} = -(u \Delta y T)_{i+\frac{1}{2},j} + (u \Delta y T)_{i-\frac{1}{2},j} + D\left(\left(\frac{\partial T}{\partial x}\Delta y\right)_{i+\frac{1}{2},j} - \left(\frac{\partial T}{\partial x}\Delta y\right)_{i-\frac{1}{2},j} + \left(\frac{\partial T}{\partial y}\Delta x\right)_{i,j+\frac{1}{2}} - \left(\frac{\partial T}{\partial y}\Delta x\right)_{i,j-\frac{1}{2}}\right) \quad (16)$$

Utilizando un esquema explícito en el tiempo, atrasado en la advección (UDS) y centrado en el espacio (CDS) en la difusión, se tienen los siguientes términos:

$$\frac{\partial T}{\partial t} V_{i,j} = \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \Delta x \Delta y \quad (17)$$

$$-(u \Delta y T)_{i+\frac{1}{2},j} = -u((y_j - y_{j-1}) T_{i-1,j}) \quad (18)$$

$$(u \Delta y T)_{i-\frac{1}{2},j} = u((y_{j+1} - y_j) T_{i-1,j}) \quad (19)$$

$$D(\frac{\partial T}{\partial x} \Delta y)_{i+\frac{1}{2},j} = D(\frac{T_E - T_C}{x_{i+1} - x_i})(y_{i,j+\frac{1}{2}} - y_{i,j-\frac{1}{2}}) \quad (20)$$

$$D(\frac{\partial T}{\partial x} \Delta y)_{i-\frac{1}{2},j} = -D(\frac{T_C - T_O}{x_i - x_{i-1}})(y_{i,j+\frac{1}{2}} - y_{i,j-\frac{1}{2}}) \quad (21)$$

$$D(\frac{\partial T}{\partial x} \Delta y)_{i,j+\frac{1}{2}} = D(\frac{T_N - T_C}{y_{j+1} - y_j})(x_{j+\frac{1}{2}} - x_{i-\frac{1}{2}}) \quad (22)$$

$$D(\frac{\partial T}{\partial x} \Delta y)_{i,j-\frac{1}{2}} = -D(\frac{T_C - T_S}{y_j - y_{j-1}})(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) \quad (23)$$

Sea la ecuación general que rige al fenómeno:

$$\begin{aligned} T_{i,j}^{n+1} &= T_{i,j}^n + \frac{dt}{(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})(y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})} + D(\frac{T_E - T_C}{x_{i+1} - x_i}(y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}) - (\frac{T_C - T_O}{x_i - x_{i-1}})(y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})) \\ &+ \frac{T_N - T_C}{y_{j+1} - y_j}(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}) - \frac{T_C - T_S}{y_j - y_{j-1}}(x_{i+\frac{1}{2},j} - x_{i-\frac{1}{2},j}) - u((y_{j+1} - y_j)T_C - (y_i - y_{i-1})T_O) \end{aligned} \quad (24)$$

$$\begin{aligned} \rho \frac{\partial \phi}{\partial t} V_{i,j} &= (\rho u \Delta y)_{i+\frac{1}{2},j} - (\rho u \Delta y)_{i-\frac{1}{2},j} + (\rho v \Delta x)_{i,j+\frac{1}{2}} - (\rho v \Delta x)_{i,j-\frac{1}{2}} \\ &- (\rho D \frac{\partial \phi}{\partial x} \Delta y)_{i+\frac{1}{2},j} + (\rho D \frac{\partial \phi}{\partial x} \Delta y)_{i-\frac{1}{2},j} - (\rho D \frac{\partial \phi}{\partial y} \Delta x)_{i,j+\frac{1}{2}} + (\rho D \frac{\partial \phi}{\partial y} \Delta x)_{i,j-\frac{1}{2}} \end{aligned} \quad (25)$$

4 Condiciones de borde

Con el fin de facilitar la resolución del problema, se imponen las siguientes condiciones de borde:

- **Frontera Norte:** Se impone la condición de simetría respecto al eje y.
- **Frontera Sur:** Se impone la condición $T_{x=[0,c],j} = T_1$ para todos los nodos pertenecientes a la placa localizada bajo en el eje x. Para aquellos que no, se impone la condición $T_{[c:],j} = T_2$
- **Frontera Este:** Se impone la condición de conservación de flujo difusivo, con lo cual el volumen en el borde $(i - \frac{1}{2}, j)$ y el flujo fuera del borde $(i + \frac{1}{2}, j)$ deben cumplir la siguiente relación:

$$D \frac{\partial T}{\partial x}_{i+\frac{1}{2},j} = D \frac{\partial T}{\partial x}_{i-\frac{1}{2},j} \quad (26)$$

$$\frac{T_{i+1,j} - T_{i,j}}{x_{i+1} - x_i} = \frac{T_{i,j} - T_{i-1,j}}{x_i - x_{i-1}} \quad (27)$$

Con lo cual, la condición de borde resultante queda como:

$$T_{nx+1}^n = \frac{x_{i+1} - x_i}{x_i - x_{i-1}}(T_{i,j} - T_{i-1,j}) + T_{i,j} \quad (28)$$

- **Frontera Oeste:** Se impone la condición $T_{x=0,j} = T_2$ para los nodos de la pared AB.

5 Perfiles iniciales

5.1 Perfil inicial de velocidades

Sea el perfil inicial de velocidades descrito por la siguiente ecuación:

$$(v(x), u(y)) = (0, v(y)) = v_0 \left(1 - \frac{y - l(y)^2}{l_y^2}\right) \quad (29)$$

Con una velocidad inicial $v_0 = 0.0010 \frac{m}{s}$, se tiene el siguiente perfil de velocidad:

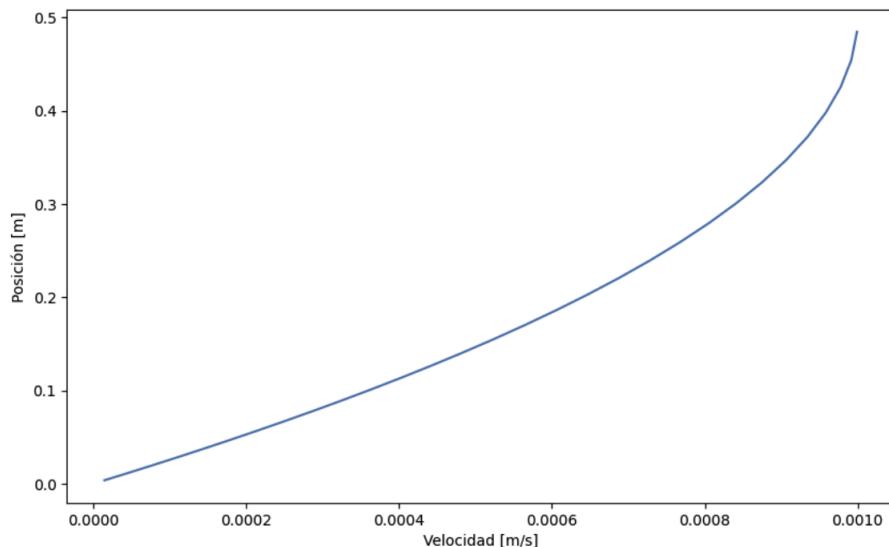


Figura 5: Perfil de velocidad.

Es importante destacar que, para el desarrollo de este ejercicio, se considerará que todo el campo de velocidades descrito en la representación del flujo inicial está dentro de la capa límite, la cual acaba en el largo [AD].

5.2 Perfil inicial de temperaturas

Con el fin de facilitar el desarrollo de la solución al problema, se utilizarán los siguientes parámetros generales:

Tabla 1: Parámetros generales asociados

Parámetros generales	
Largo del canal	1 [m]
Ancho de la placa	0.2 [m]
Alto del canal	0.5 [m]
Volúmenes en el eje x	25
Volúmenes en el eje y	30
Espaciamiento espacial mínimo	0.04 [m]
Difusividad Térmica	5e-5 [m ² /s]
Velocidad inicial	0.001[m/s]
Temperatura de la placa (T1)	500 [°C]
Temperatura general (T2)	50 [°C]

Con ello, y haciendo uso de las condiciones de borde anteriormente descritas, se obtiene el siguiente perfil incial:

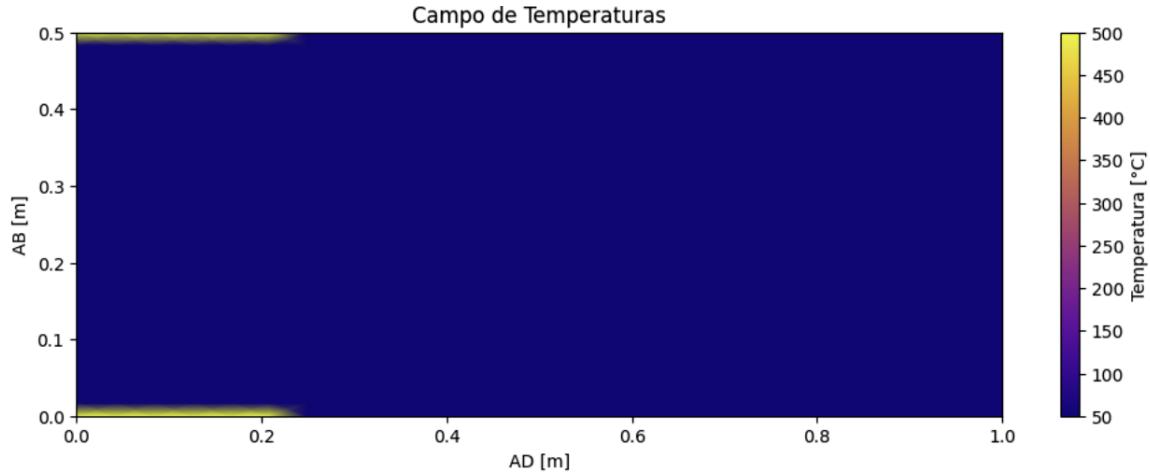


Figura 6: Perfil de Temperatura inicial.

En este se puede visualizar que todo el campo permanece a una temperatura t2 constante, mientras que el borde, en el cual se ubica la placa, se mantiene a una temperatura T_1 .

6 Resultados

Desarrollando la ecuación general de Advección - Difusión tanto en la malla regular como irregular, y calculando un tiempo de 1000 [s], se tienen las siguientes soluciones:

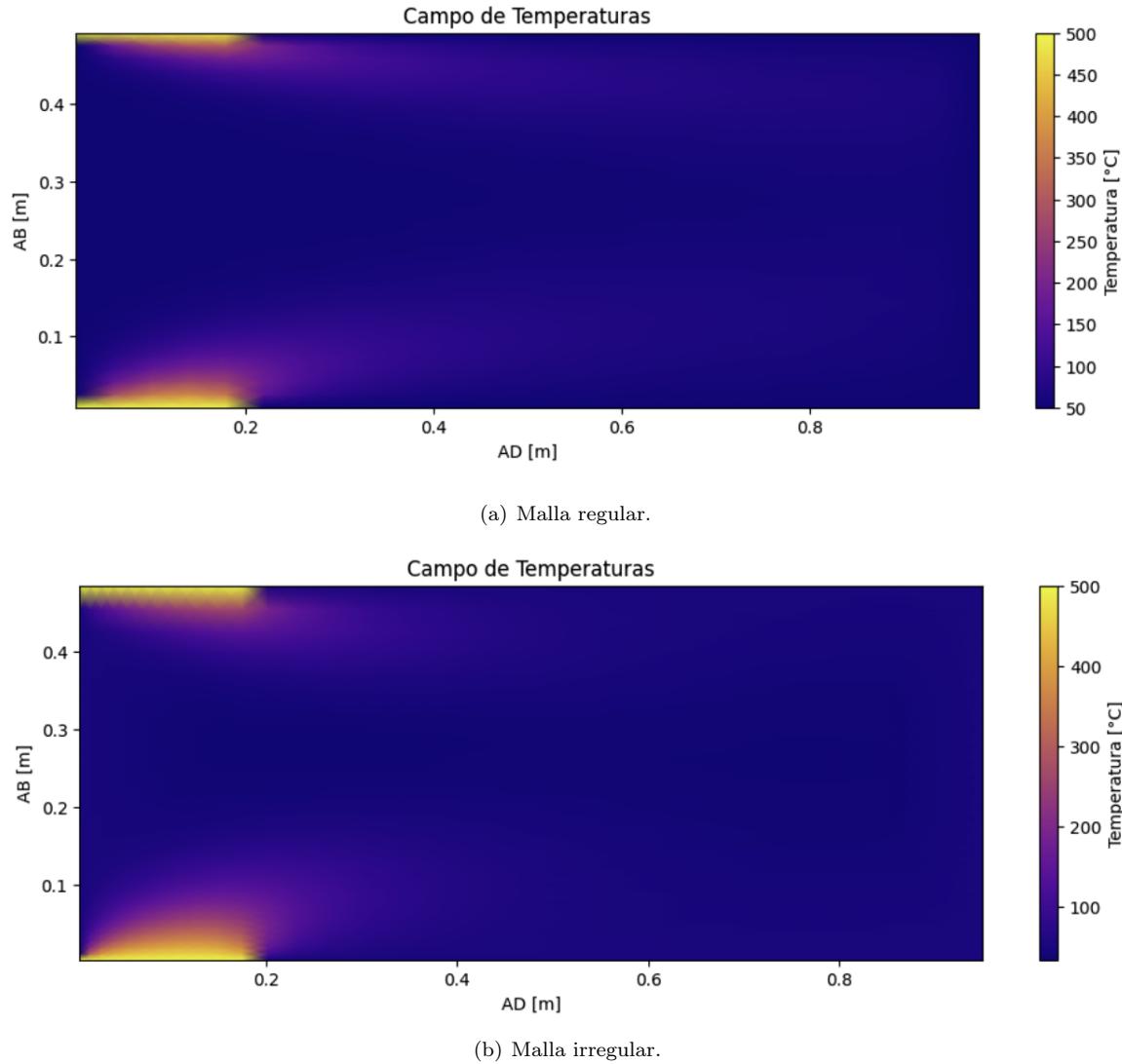


Figura 7: Solución para el tiempo de convergencia.

En ellas se puede notar que la solución de la malla regular es "más plana", es decir, está sufriendo menos difusión en el sentido del eje y, lo que se aprecia de sobemanera cercano al eje $(x,y) = (0,0)$. Esto se debe a que la escala el mallado regular no es lo suficientemente pequeña para notar las diferencias entre los volúmenes, lo que sí logra la malla irregular al poseer volúmenes de tamaño variable en ambas direcciones.

los tiempos de convergencia son los siguientes:

- **Advección - Difusión en mallado regular 2D:** 2577 [s]
- **Advección - Difusión en mallado irregular 2D:** 1917 [s].

Si se cambian las temperaturas y la longitud de la placa para visualizar un caso más exagerado, se obtiene:

$$T_1 = 1500[C], T_2 = 50[C], AE = 0.4[m]$$

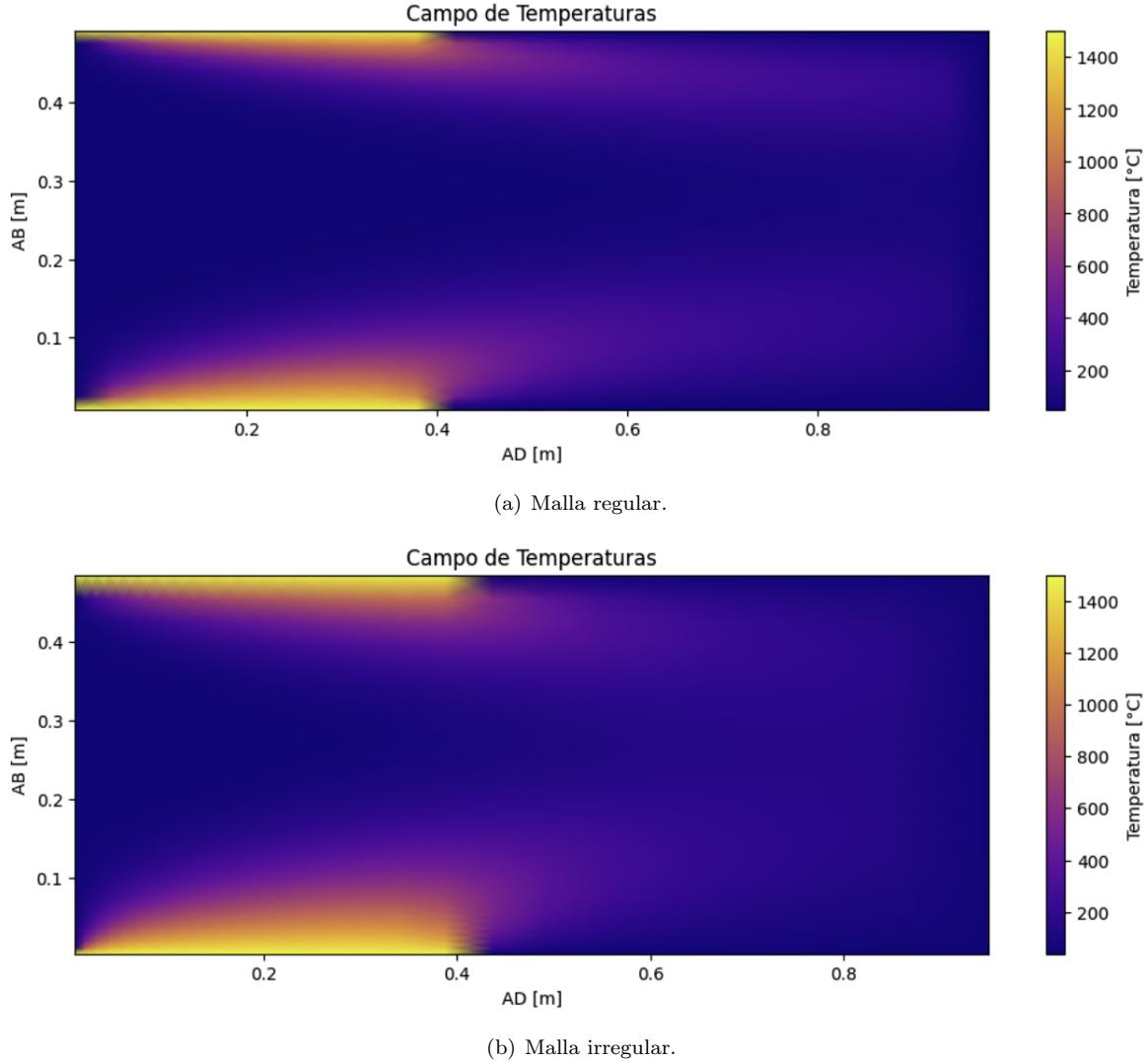


Figura 8: Solución para el tiempo de convergencia.

los tiempos de convergencia son los siguientes:

- **Advección - Difusión en mallado regular 2D:** 3639 [s]
- **Advección - Difusión en mallado irregular 2D:** 2605 [s].

En las imágenes anteriores se puede visualizar claramente que existe una pérdida difusiva en la malla regular, causada por el tamaño de los volúmenes, los cuales no son apropiados para la resolución del problema presentado.

7 Validación

Con el fin de validar el código anteriormente desarrollado con el caso 1D - Académico, se procede a estudiar los fenómenos de Advección y Difusión en sus direcciones existentes.

7.1 Advección pura en x

Considerando una malla unidimensional, la cual se compone de elementos en el eje x que componen a un vector del siguiente estilo:

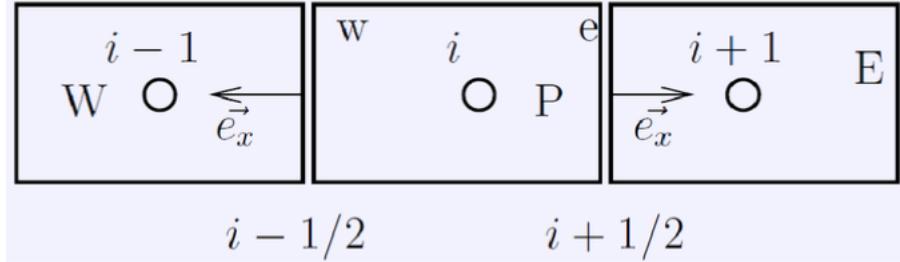


Figura 9: Mallado unidimensional para la advección.

Sea la ecuación de advección en una dimensión sin fuente:

$$\frac{\partial}{\partial t} \iiint_{\Omega_{CV}} \rho \phi d\Omega = - \oint_{A_{CV}} \vec{F} \cdot d\vec{A}_n \quad (30)$$

Desarrollando los términos matemáticos y considerando que $u > 0$ se tiene:

$$\frac{\partial}{\partial t} \rho \phi V_i = -[(\rho \phi \vec{u})_{i+\frac{1}{2}} \Delta y - (\rho \phi \vec{u})_{i-\frac{1}{2}} \Delta y] \quad (31)$$

Utilizando UDS en el espacio y Euler explícito en el tiempo:

$$T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x \Delta y} [(T_i^n \vec{u})_{i+\frac{1}{2}} \Delta y - (T_{i-1}^n \vec{u})_{i-\frac{1}{2}} \Delta y] \quad (32)$$

Simplificando los términos, y reemplazando según las direcciones de la rosa de los vientos antes mencionada:

$$T_i^{n+1} = T_i^n - \frac{\Delta t}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \vec{u} [(T_C - T_O)] \quad (33)$$

Con ello, y para diferentes CFL obtenidos con el cambio en el paso de tiempo, más no el tiempo total, se observan los siguientes resultados:

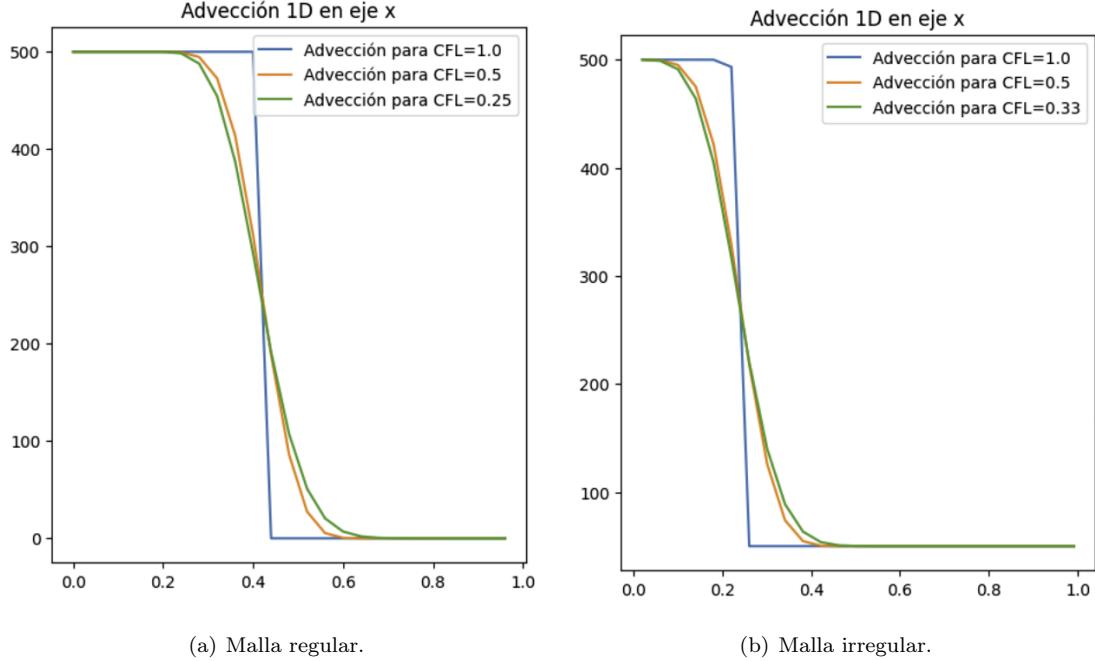


Figura 10: Solución para $t = 500$ [s].

En base a las figuras asociadas, se puede llegar a las siguientes conclusiones:

- Tal como la teoría lo indica, una vez que se llega al rango estable del número de Courant-Friedrichs-Levy ($CFL \leq 1$), las curvas se vuelven más precisas a menor valor. En este fenómeno en particular, la variación entre curvas no es significante, pues la advección es el fenómeno que menos aporta a la transferencia de calor en este problema.
- La velocidad de convergencia de la malla regular es superior a la de asociada a la malla irregular, lo que se puede visualizar en la posición x en el que estas se encuentran. Esto se debe a que los volúmenes en el mallado irregular poseen menores dimensiones, por lo que existen más volúmenes en el mismo espacio. La iteración de este mayor número de volúmenes "atrasa" la gráfica respecto al esquema regular.
- Respecto a las curvas como tal, estas poseen una forma esperada para el fenómeno estudiado. Si se deja correr el programa más tiempo, se podrá notar que para el tiempo del estado estacionario, todos los puntos del sistema habrán sufrido advección.

7.2 Difusión pura en x

Utilizando el mismo mallado anteriormente descrito, se procede a trabajar con la ecuación de la difusión unidimensional:

$$\frac{\partial}{\partial t} \iiint_{V_i} \rho \phi V_i = - \iint_{\Delta y} \rho D \frac{\partial \phi}{\partial x} \vec{e}_x dy \quad (34)$$

Desarrollando matemáticamente la ecuación se tiene:

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \Delta x \Delta y = -D \left(\frac{T_{i+1,j} - T_{i,j}}{x_{i+1,j} - x_{i,j}} \right) \Delta y + D \left(\frac{T_{i,j} - T_{i-1,j}}{x_{i,j} - x_{i-1,j}} \right) \Delta y \quad (35)$$

Finalmente, simplificando términos y reemplazando según la rosa de los vientos anteriormente descrita se tiene:

$$T_{i,j}^{n+1} = T_{i,j}^n \frac{\Delta t}{\Delta x} \left(-D \left(\frac{T_E - T_C}{x_{i+1,j} - x_{i,j}} \right) + D \left(\frac{T_C - T_O}{x_{i,j} - x_{i-1,j}} \right) \right) \quad (36)$$

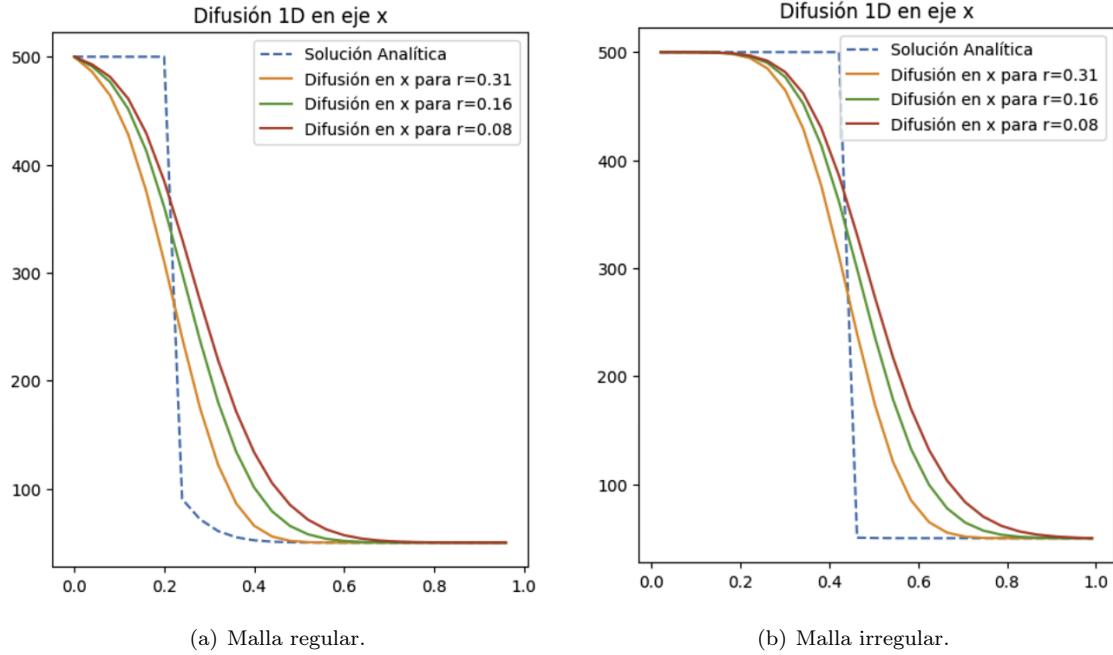


Figura 11: Solución para $t = 500$ [s].

En las figuras asociadas podemos ver que la solución analítica se ha adaptado a la condición de borde en la pared (c nodos con una temperatura T_1). Con ello, se pueden sacar las siguientes conclusiones:

- La solución analítica en la malla regular se ve afectada por la distancia entre volúmenes, lo que se nota en la zona inferior, cercano a los $0.4[m]$. Este fenómeno no se ve en la malla irregular, pues esta se adapta de mejor manera, perdiendo menos información.
- La solución irregular converge más rápido que la solución correspondiente a la malla irregular, lo que se nota en el avance de las curvas.
- En ambos casos se cumple la condición de estabilidad ($r <= 0.5$). Además, dentro del rango estable del número de Fourier ($r = 0.31, 0.16, 0.08$, las curvas poseen una forma similar, siendo más estables aquellas que poseen un r menor, tal como lo indica la teoría.
- Respecto a la forma de las curvas, se puede concluir que estas son coherentes con la teoría, ya que la difusión es dependiente del paso en el tiempo y espacio, por lo que esta tiene un efecto

más lento que el de la advección anteriormente vista. Si este sistema se deja correr más tiempo, se alcanzará el estado estacionario, en el cual todos los puntos serán afectados por la difusión.

7.3 Difusión pura en y

Siguiendo los mismos pasos que para la difusión en x, se tiene que la difusión en la dirección y será:

Finalmente, simplificando términos y reemplazando según la rosa de los vientos anteriormente descrita se tiene:

$$T_{i,j}^{n+1} = T_{i,j}^n \frac{\Delta t}{\Delta y} \left(-D \left(\frac{T_N - T_O}{y_{i,j+1} - y_{i,j}} \right) + D \left(\frac{T_C - T_S}{y_{i,j} - y_{i,j-1}} \right) \right) \quad (37)$$

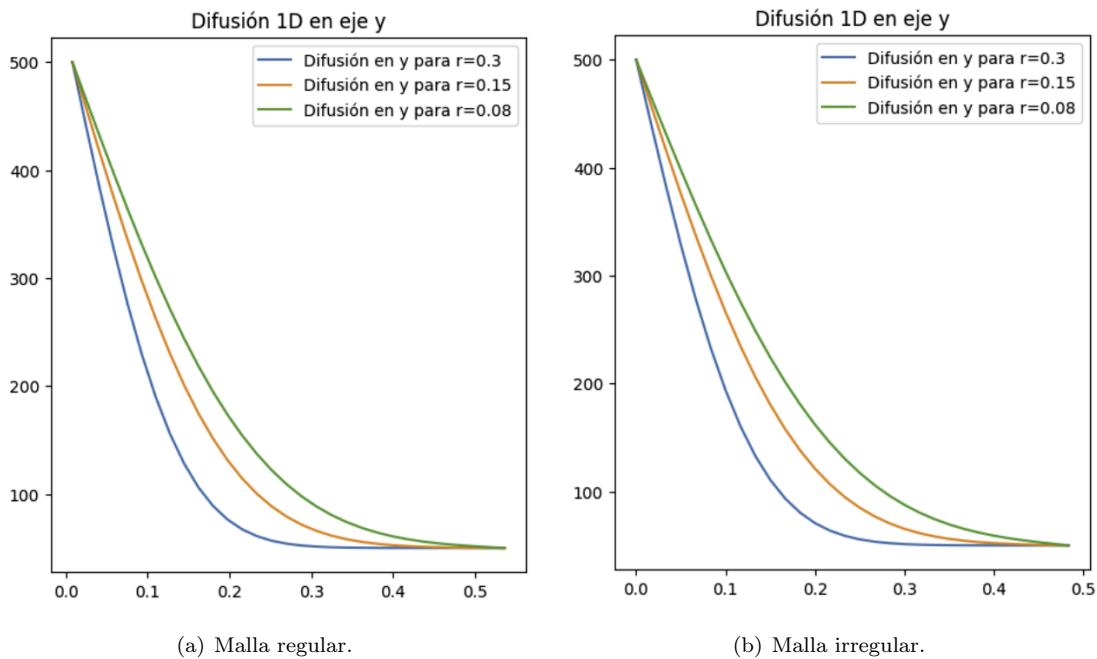


Figura 12: Solución para $t = 500$ [s].

A diferencia del caso en la dirección x, la difusión en la dirección y sólo se ve afectada por un nodo caliente en la posición $(x,y) = (0,0)$. Es por ello que, la curva "cae" desde la primera posición.

En base a las gráficas, se puede llegar a las siguientes conclusiones:

- A diferencia de los fenómenos anteriores, se puede visualizar que las curvas de la malla regular e irregular son casi idénticas. Esto se atribuye a que la distancia del canal en el sentido de y es menor, y posee un mayor número de volúmenes, razón por la cual $\Delta x > \Delta y$.
- Los números de Fourier asociados hacen variar la precisión, tal como lo indica la teoría. Al ser menores las variaciones en y, no existe una gran diferencia entre los pasos, a pesar de que los mallados sean diferentes.

- La curva es consistente con lo esperado, ya que el campo es un vector unidimensional, razón por la cual los puntos van sufriendo del fenómeno de difusión uno a la vez. Si se deja correr el programa más tiempo, probablemente se notará que todos los puntos están siendo afectados por la difusión.
- Es importante destacar que esta solución se realizó sin considerar una condición de simetría en el centro del vector. En caso de aplicarse $T_{j=0} = T_{j=ny} = T_1$, se verá que la convergencia lleva a una línea recta en el centro.

7.4 Permanencia del régimen

7.4.1 Malla regular

Respecto al régimen regular, se convergen las siguientes gráficas de transporte de temperatura, utilizando siguiente criterio:

$$T_{i,j}^{n+1} - T_{i,j}^n < 0.2 \quad (38)$$

Con ello, se obtienen las siguientes figuras:

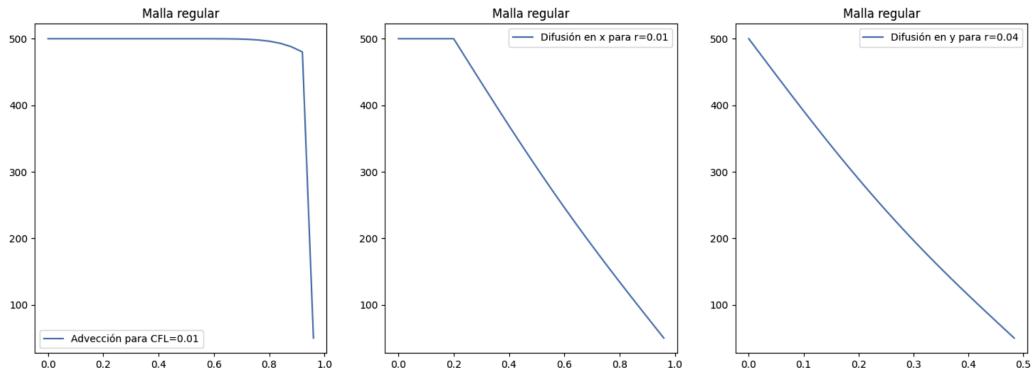


Figura 13: Transporte de temperatura en malla regular.

Los tiempos de convergencia asociados son:

- **Tiempo de convergencia de la advección en x:** 2577 [s].
- **Tiempo de convergencia de la difusión en x:** 16610 [s].
- **Tiempo de convergencia de la difusión en y:** 5712 [s].

Lo cual es concordante con el análisis de validación realizado con anterioridad.

Es importante destacar que la advección en x no se ve en su punto máximo más allá de la placa. Esto es un error que claramente indica que la dimensión de los volúmenes utilizados no es la apropiada para el estudio del fenómeno, pues se está perdiendo información en el vecindario cercano.

7.4.2 Malla irregular

Respecto al regimen irregular, se convergen las gráficas de transporte de temperatura bajo el mismo criterio de la malla regular.

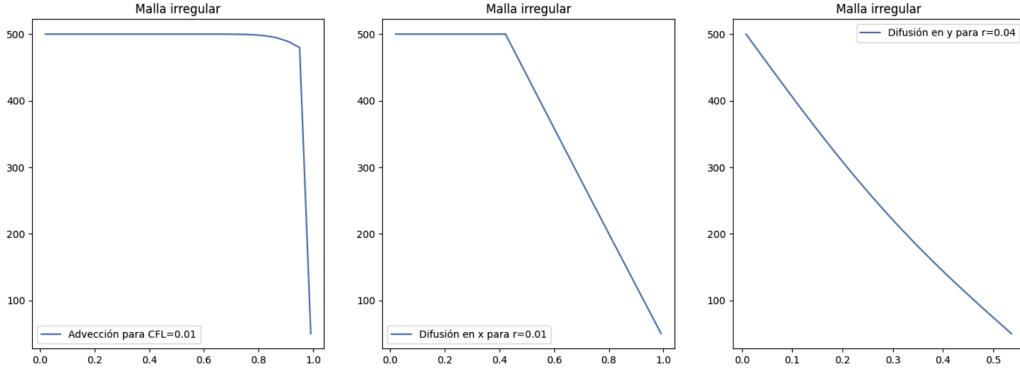


Figura 14: Transporte de temperatura en malla irregular.

Con ello, se obtienen los siguientes tiempos asociados:

Los tiempos de convergencia asociados son:

- **Tiempo de convergencia de la advección en x:** 1997 [s].
- **Tiempo de convergencia de la difusión en x:** 18339 [s].
- **Tiempo de convergencia de la difusión en y:** 6311 [s].

A diferencia de la malla regular, se puede visualizar que la convergencia de la difusión en x alcanza su punto máximo en más puntos que los pertenecientes a la placa inicial. Esto se asocia a que, como el mallado es mucho más fino, los puntos cercanos alcanzan la misma temperatura de la placa con el pasar del tiempo. Esto tiene cierto grado de posibilidad, pues el dominio es pequeño y el tiempo de convergencia muy grande.

7.5 Escalas de tiempo asociadas

Para caracterizar el dominio relativo de las componentes de Advección y Difusión en el fenómeno anteriormente estudiado, se procede a calcular el número de Péclet en el eje x:

$$Pe = \frac{c}{r} = \frac{u\Delta x}{D} = \frac{\tau_{Dif}}{\tau_{Adv}} \quad (39)$$

Con lo cual se tiene que:

- **Malla regular:**

$$Pe = \frac{\tau_{Adv}}{\tau_{Dif}} = \frac{2577[s]}{16610[s]} \approx 0.15[-] \quad (40)$$

- **Malla irregular:**

$$Pe = \frac{\tau_{Adv}}{\tau_{Dif}} = \frac{1997[s]}{18339[s]} \approx 0.10[-] \quad (41)$$

Con lo cual se concluye que independiente de la malla utilizada, el fenómeno que denomina el problema es la difusión en el eje x.

Es importante destacar que en el eje y el Péclet es igual a 0, ya que no existe velocidad y por consiguiente, no existe advección.

8 Conclusiones

Respecto al problema y solución presentadas se puede concluir:

- En el problema estudiado, el mayor impacto en el transporte de calor lo posee la difusión, lo cual se puede verificar con el número de Péclet antes mencionado. Esta difusión posee dos componentes, siendo la más en primera instancia aquella en la dirección y, pues esta converge más rápido que su compañera en x.

El impacto y tiempo de esta transferencia de calor depende directamente de los parámetros asociados.

- Es de suma relevancia utilizar el mallado apropiado para la resolución de problemas de Dinámica de Fluidos Computacional (CFD). Para poder apreciar correctamente los cambios de propiedades, se debe mallar más finamente cerca de singularidades. Para puntos alejados de estas, se puede trabajar con una malla más ancha, y/o una malla regular.
- La ecuación de Advección - Difusión permite estudiar fenómenos de transferencia de temperatura entre cuerpos. La predominancia de una o la otra depende intrínsecamente del problema en cuestión. Es por ello que, para optimizar soluciones y comprender fenómenos, se debe realizar una correcta clasificación con el uso de Números Adimensionales, como lo es el número de Péclet. Este desarrollo no sólo permitirá facilitar la comprensión de la permanencia del régimen, sino que también entregará la posibilidad de reescalar el problema sin mayores consecuencias.
- La velocidad de un flujo afecta al fenómeno de advección, más no a la difusión, la cual depende principalmente del paso temporal y espacial.
- Siempre se debe estudiar la estabilidad de los métodos computacionales (Ej: Número de Courant, Fourier, etc). En el caso de que se esté utilizando una malla con paso espacial variable, este estudio se debe realizar con el peor caso de todos, que corresponde al paso máximo dx . Esto permitirá asegurar que todos los puntos diferentes a el poseerán una estabilidad mayor, y asegurará el funcionamiento del esquema.

▼ Paquetes y funciones

```
#Importar paquetes necesarios
import numpy as np
from matplotlib import pyplot
import matplotlib.pyplot as plt
import imageio
import math
```

▼ Parámetros

```
#Dimensiones del canal
lx = 1
ly = 0.5
AE = 0.4

#Sea la cantidad de volúmenes en 2d
nx = 25
ny = 30
```

```
#Sea el mallado espacial
x = np.zeros(nx)
y = np.zeros(ny)

pmx = np.zeros(nx+1)
pmy = np.zeros(ny+1)

dx = lx/nx
dy = ly/ny
```

```
#Líneas medias
for i in range(nx+1):
    if i == 0:
        pmx[i] = 0
    else:
        pmx[i] = pmx[i-1] + dx

for j in range(ny+1):
    if j == 0:
        pmy[j] = 0
    else:
        pmy[j] = pmy[j-1] + dy
```

```
#Sean los centros de los volúmenes
for i in range(nx):
    x[i] = (pmx[i]+pmx[i+1])/2

for j in range(ny):
    y[j] = (pmy[j]+pmy[j+1])/2
```

```
#Coeficiente de difusividad térmica del agua
D = 1*(10**-5)

#Temperaturas [°C]
t1 = 1500
t2 = 50

#Velocidad inicial
v0 = 0.001

#Tiempo
nt = 1000
t = 1000
dt = t/nt

#Sea el mallado temporal
t = np.linspace(0, t, nt)

#Para aplicar la condición de borde
c = 0
for elementos in x:
    if elementos < AE:
        c = c+1

#Sea la malla
X,Y = np.meshgrid(x,y)

fig, ax = plt.subplots(figsize=(16, 14)) # Ancho x Altura en pulgadas

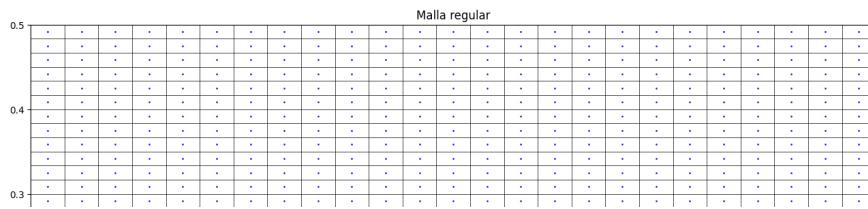
ax.vlines(pmx, ymin=0, ymax=ly, colors='black', linestyles='solid', linewidth=0.5, label='Líneas Verticales')
ax.hlines(pmy, xmin=0, xmax=lx, colors='black', linestyles='solid', linewidth=0.5, label='Líneas Horizontales')

ax.scatter(X,Y, color='blue', marker='o', s = 3, linewidth=0.1, label='Puntos')

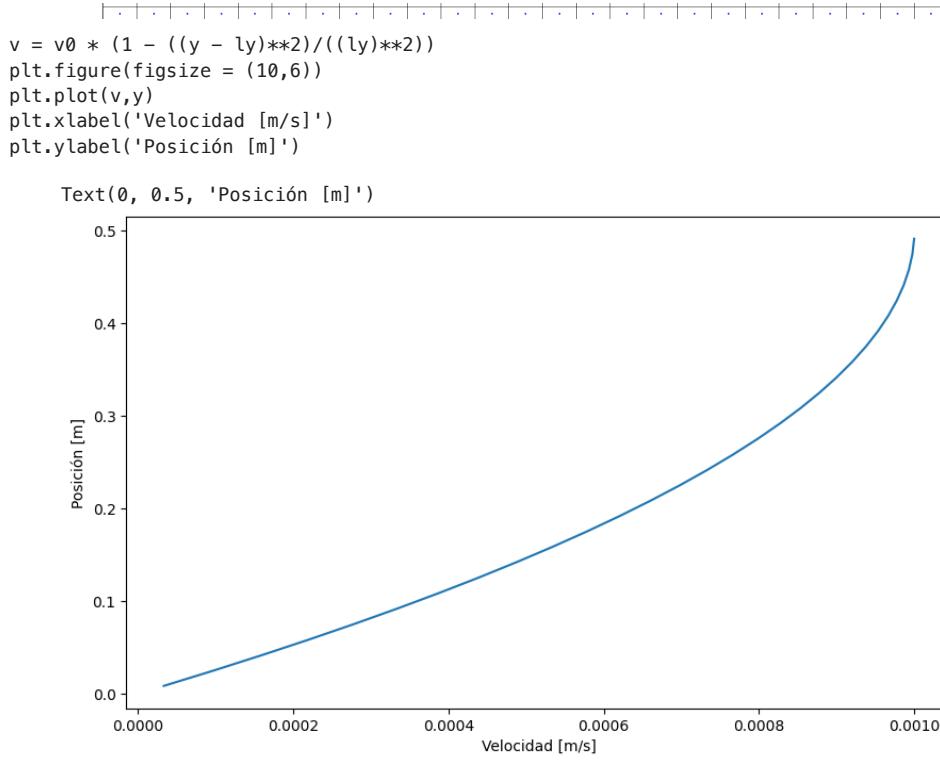
Ti = np.zeros((ny,nx))
plt.imshow(Ti, cmap = 'Greys', interpolation='bilinear', extent=[0, lx, 0, ly])

plt.title('Malla regular')
plt.xlabel('AD [m]')
plt.ylabel('AB [m]')

plt.show()
```



▼ Perfil inicial de velocidades



▼ Perfil inicial de temperatura

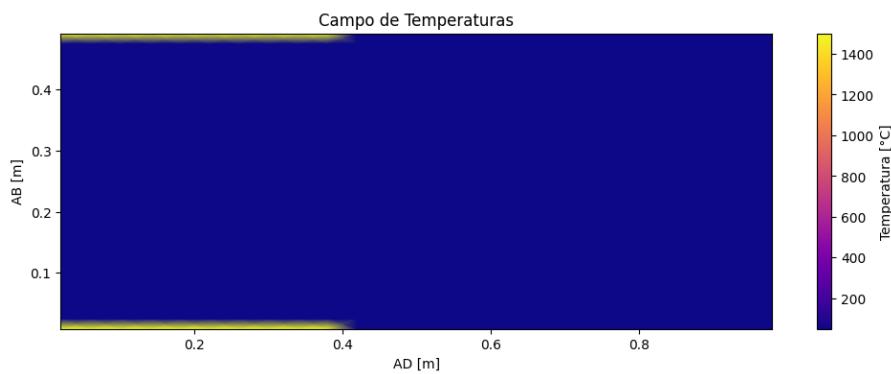
```
#Perfil inicial
Ti = np.zeros((ny, nx))
Ti[:, :] = t2
Ti[0, 0:c] = t1
Ti[ny-1, 0:c] = t1
T = Ti.copy()

fig, ax = plt.subplots(figsize=(12, 4))

pc = plt.pcolormesh(x, y, Ti, cmap='plasma', shading = 'gouraud')

plt.title('Campo de Temperaturas')
plt.xlabel('AD [m]')
plt.ylabel('AB [m]')
plt.colorbar(pc, label='Temperatura [°C]')

plt.show()
```



▼ Ecuación asociada

Sea la ecuación de convección lineal en una dimensión:

$$\frac{\partial T}{\partial t} \cdot \Delta x \cdot \Delta y = -(u \Delta y T)_{i+\frac{1}{2},j} + (u \Delta y T)_{i-\frac{1}{2},j} + D \left(\frac{\partial T}{\partial x} \Delta y_{i+\frac{1}{2},j} - \frac{\partial T}{\partial x} \Delta y_{i-\frac{1}{2},j} + \frac{\partial T}{\partial y} \Delta x_{i,j+\frac{1}{2}} - \frac{\partial T}{\partial y} \Delta x_{i,j-\frac{1}{2}} \right)$$

Utilizando un esquema explícito en el tiempo, atrasado en la advección y centrado en el espacio se tiene:

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \Delta x \Delta y = (\Delta y(uT_P - uT_O)) + D \left(\frac{T_E - T_C}{\Delta x} \Delta y - \frac{T_C - T_O}{\Delta x} \Delta y + \frac{T_N - T_C}{\Delta y} \Delta x - \frac{T_C - T_S}{\Delta y} \Delta x \right).$$

Despejando,

$$T_{i,j}^{n+1} = \frac{\Delta t}{\Delta x \Delta y} \left(D \left(\frac{T_E + T_O - 2T_C}{\Delta x} \Delta y + \frac{T_N + T_S - 2T_C}{\Delta y} \Delta x \right) - \Delta y(uT_C - uT_O) \right) + T_{i,j}^n$$

Separando los términos en advección y difusión, se tiene:

$$\begin{aligned} Adv &= (\Delta y(uT_C - uT_O)) \\ Diff &= -D \left(\frac{T_E - T_C}{\Delta x} \Delta y - \frac{T_C - T_O}{\Delta x} \Delta y + \frac{T_N - T_C}{\Delta y} \Delta x - \frac{T_C - T_S}{\Delta y} \Delta x \right) = -D \left(\frac{T_E + T_O - 2T_C}{\Delta x} \right) \Delta y \\ &\quad - D \left(\frac{T_N + T_S - 2T_C}{\Delta y} \right) \Delta x \\ T_{i,j}^{n+1} &= \frac{\Delta t}{\Delta x \Delta y} (Diff) + Adv + T_{i,j}^n \end{aligned}$$

▼ Aplicación de las ecuaciones: Código 2D

```

def temp(T, ny, v0, nx, ly, y, D, x, t2, t1, dt, t, diferencia):
    while diferencia > 1e-1:
        Tn = T.copy()
        t = t+1
        for j in range((ny-1)):

            v = v0 * (1 - ((y[j] - ly)**2)/((ly)**2))

            for i in range((nx-1)):
                #FLUJOS EN TODAS LAS DIRECCIONES
                fe = D* ( ((Tn[j,i+1] - Tn[j,i])/(dx) ) #FLUJO DIFUSIÓN ESTE
                fo = D* ( ((Tn[j,i] - Tn[j,i-1])/(dx) ) #FLUJO DIFUSIÓN OESTE
                fn = D* ( ((Tn[j+1,i] - Tn[j,i])/(dy) ) #FLUJO DIFUSIÓN NORTE
                fs = D* ( ((Tn[j,i] - Tn[j-1,i])/(dy) ) #FLUJO DIFUSIÓN SUR

                #NODO DIAGONAL INFERIOR IZQUIERDO
                if i == 0 and j == 0:
                    Tn[j, i-1] = t2
                    Tn[j-1,i] = t1
                    dif = fe - fo + fn - fs

                #NODO DIAGONAL INFERIOR DERECHO
                elif i == (nx-1) and j == 0:
                    if i < c:
                        Tn[j-1, i] = t1
                    else:
                        Tn[j-1, i] = t2
                    dif = fn - fs

                #NODO DIAGONAL SUPERIOR IZQUIERDO
                elif i == 0 and j == (ny-1):
                    Tn[j, i-1] = t2
                    Tn[j+1,i] = t1
                    dif = fe - fo + fn - fs

                #NODO DIAGONAL SUPERIOR DERECHO
                elif i == (nx-1) and j == (ny-1):
                    if i < c:
                        Tn[j-1, i] = t1
                    else:
                        Tn[j-1, i] = t2

                    dif = fn - fs

                #COLUMNA OESTE
                elif i == 0 and 0 < j < (ny-1):
                    Tn[j, i-1] = t2
                    dif = fe - fo + fn - fs

                #COLUMNA ESTE
                elif i == (nx-1) and 0 < j < (ny-1):
                    if i < c:
                        Tn[j-1, i] = t1
                    else:
                        Tn[j-1, i] = t2
                    dif = fn - fs

                #FILA SUR
                elif j == 0 and 0 < i < (nx-1):
                    if i < c:
                        Tn[j-1, i] = t1
                    else:
                        Tn[j-1, i] = t2
                    dif = fe - fo + fn - fs

                #FILA NORTE
                elif j == (ny-1) and 0 < i < (nx-1):
                    if i < c:
                        Tn[j+1, i] = t1
                    else:
                        Tn[j+1, i] = t2
                    dif = fe - fo - fs + fn

                else:
                    dif = fe - fo + fn - fs

            adv = v* (Tn[j,i] -Tn[j,i-1]) #ADVECCIÓN

```

```

T[j,i] = Tn[j,i] + (dt/ (dx)) * (dif - adv)

T[1:, 0] = t2
T[0, 0:c] = t1
T[0, c:] = t2
T[ny-1, 0:c] = t1
T[ny-1, c:] = t2
diferencia = np.sum(np.abs(T - Tn))
return T, t

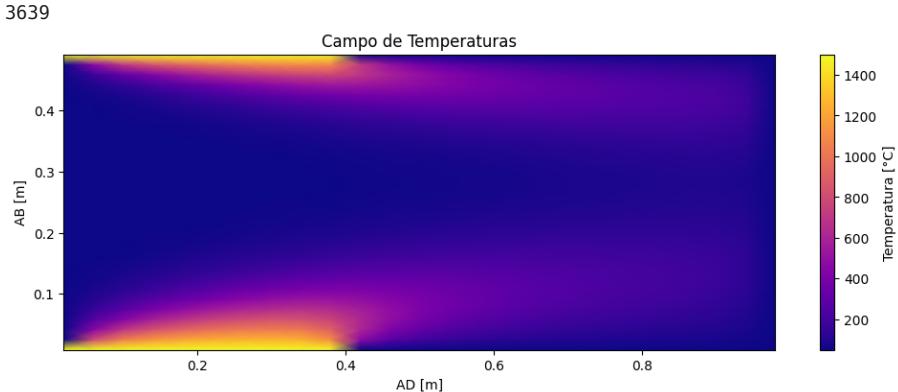
T = Ti.copy()
diferencia = 1
t = 0
T, t = temp(T, ny, v0, nx, ly, y, D, x, t2, t1, dt, t, diferencia)
print(t)
fig, ax = plt.subplots(figsize=(12, 4))

pc = plt.pcolormesh(x, y, T, cmap='plasma', shading = 'gouraud')

plt.title('Campo de Temperaturas')
plt.xlabel('AD [m]')
plt.ylabel('AB [m]')
plt.colorbar(pc, label='Temperatura [°C]')

plt.show()

```



▼ Validación del código

▼ Caso 1: Advección pura

Si el fluido sólo sufre advección con una velocidad constante de v_0 en el eje x, se tiene el siguiente código (Basado en las ecuaciones anteriormente descritas):

```
#Dimensiones del canal
lx = 1
ly = 0.5
AB = 0.2

#Temperaturas [°C]
t1 = 500
t2 = 50

#Velocidad inicial
v0 = 0.0004

D = 1e-5

#Sea la cantidad de volúmenes en 2d
nx = 25
ny = 30

#Sea el mallado espacial
x = np.arange(0, lx, lx/nx)
y = np.arange(0, ly, ly/ny)

#Sea el espaciado espacial
dx = x[1] - x[0]
dy = y[1] - y[0]
c = 0
for elementos in x:
    if elementos <= AB:
        c = c+1

def ADV1Dx(t, nx, v, dt, dx, Tadv):
    for ti in t:
        Tn = Tadv.copy()
        for i in range(1, nx-1):
            Tadv[i] = Tn[i] - (dt/dx)*v*(Tn[i] - Tn[i-1])
        diferencia = np.sum(np.abs(Tadv - Tn))
    return Tadv

def DIF1Dx(t, nx, dt, dx, D, Tdif, x):
    for ti in t:
        Tn = Tdif.copy()
        for i in range(1, nx-1):
            Tdif[i] = Tn[i] + (dt/(dx)) *D*(((Tn[i+1] - Tn[i]))/(dx)) - ((Tn[i] - Tn[i-1]))/(dx) )
    return Tdif

def DIF1Dy(t, nx, dt, dy, D, Tdif, y):
    for ti in t:
        Tn = Tdif.copy()
        for j in range(1, ny-1):
            Tdif[j] = Tn[j] + (dt/(dy)) *D*(((Tn[j+1] - Tn[j]))/(dy)) - ((Tn[j] - Tn[j-1]))/(dy) )
    return Tdif
```

```

fig, axs = plt.subplots(1, 3, figsize=(18, 6))

nt = [5, 10, 20]
# Gráfica para Advección en x
axs[0].set_title('Advección 1D en eje x')
for n in nt:
    t = 500
    dt = t / n
    t = np.linspace(0, t, n)
    CFL = (v0 * dt) / dx
    Tadv = np.zeros(nx)
    Tadv[:c] = t1
    Tadv = ADV1Dx(t, nx, v0, dt, dx, Tadv)
    axs[0].plot(x, Tadv, label=f'Advección para CFL={np.round(CFL, 2)}')
    axs[0].legend()

nt = [10, 20, 40]
# Gráfica para Difusión en x
axs[1].set_title('Difusión 1D en eje x')

Tdifx = np.zeros(nx)
Tdifx[:c] = t1
Tdifx[c:] = t2
legend_added = False

for n in nt:
    t = 500
    dt = t / n
    t = np.linspace(0, t, n)
    r = (dt * D) / (dx ** 2)
    Tdifx[:c] = t1
    Tdifx = DIF1Dx(t, nx, dt, dx, D, Tdifx, x)
    if not legend_added:
        Tan = np.zeros(nx)
        erf_vectorized = np.vectorize(math.erf)
        Tan = erf_vectorized((0.5 * x) / ((D * 1000) ** 0.5)) * (t2 - t1) + t1
        Tan[:c] = t1
        axs[1].plot(x, Tan, linestyle='--', label='Solución Analítica')
        legend_added = True

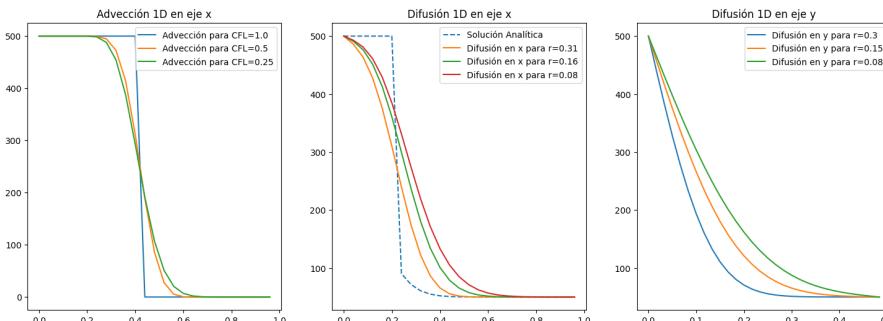
    axs[1].plot(x, Tdifx, label=f'Difusión en x para r={np.round(r, 2)}')
    #analítica
    axs[1].legend()

Tdify = np.zeros(ny)
Tdify[0] = t1
Tdify[1:] = t2
nt = [60, 120, 240]
# Gráfica para Difusión en y
axs[2].set_title('Difusión 1D en eje y')
for n in nt:
    t = 500
    dt = t / n
    t = np.linspace(0, t, n)
    r = dt * D / (dy ** 2)
    Tdify[0] = t1
    Tdify = DIF1Dy(t, nx, dt, dy, D, Tdify, y)

    axs[2].plot(y, Tdify, label=f'Difusión en y para r={np.round(r, 2)}')
    axs[2].legend()

plt.show()

```



▼ Tiempos de convergencia

```

def ADV1Dx(t, nx, v, dt, dx, Tadv, dif):
    while dif > 0.2:
        t = t+1
        Tn = Tadv.copy()
        for i in range(1, nx-1):
            Tadv[i] = Tn[i] - (dt/dx)*v*(Tn[i] - Tn[i-1])

        dif = np.sum(np.abs(Tadv - Tn))
        Tadv[:c] = t1
    return Tadv, t

def DIF1Dx(t, nx, dt, dx, D, Tdif, x, dif):
    while dif > 0.2:
        t = t+1
        Tn = Tdif.copy()
        for i in range(1, nx-1):
            Tdif[i] = Tn[i] + (dt/(dx**2)) *D*((Tn[i+1] - Tn[i]) - (Tn[i] - Tn[i-1]))

        dif = np.sum(np.abs(Tdif - Tn))
        Tdif[-1] = t2
        Tdif[:c] = t1
    return Tdif, t

def DIF1Dy(t, nx, dt, dy, D, Tdif, y, dif):
    while dif > 0.2:
        t = t+1
        Tn = Tdif.copy()
        for j in range(1, ny-1):
            Tdif[j] = Tn[j] + (dt/dy**2) *D*((Tn[j+1] - Tn[j]) - (Tn[j] - Tn[j-1]))

        dif = np.sum(np.abs(Tdif - Tn))
        Tdif[0] = t1
        Tdif[-1] = t2
    return Tdif, t

# Crear subgráficas para Advección y Difusión
fig, axs = plt.subplots(1, 3, figsize=(18, 6))

CFL = (v0 * dt) / dx
Tadv = np.zeros(nx)
Tadv[:c] = t1
Tadv[c:] = t2
dif = 1
t = 0
Tadv, t = ADV1Dx(t, nx, v0, dt, dx, Tadv, dif)
print(t)
axs[0].plot(x, Tadv, label=f'Advección para CFL={np.round(CFL, 2)}')
axs[0].set_title('Malla regular')
axs[0].legend()

axs[1].plot(x, Tn, label='Solución Analítica')
axs[1].plot(x, Tdif, label='Difusión en x para r=0.31')
axs[1].plot(x, Tn, label='Difusión en x para r=0.16')
axs[1].plot(x, Tn, label='Difusión en x para r=0.08')
axs[1].set_title('Difusión 1D en eje x')
axs[1].legend()

axs[2].plot(y, Tn, label='Difusión en y para r=0.3')
axs[2].plot(y, Tdif, label='Difusión en y para r=0.15')
axs[2].plot(y, Tn, label='Difusión en y para r=0.08')
axs[2].set_title('Difusión 1D en eje y')
axs[2].legend()

```

```
t = (ux * v) / (ux ** 2)
Tdifx = np.zeros(nx)
Tdifx[:c] = t1
Tdifx[c:] = t2
dif = 1
```

▼ Paquetes

```
#Importar paquetes necesarios
import numpy as np
from matplotlib import pyplot
import matplotlib.pyplot as plt
import imageio
import math
```

▼ Malla espacial

```
#Dimensiones del canal
lx = 1
ly = 0.5
AE = 0.4

#Sea la cantidad de volúmenes en 2d
nx = 25
ny = 30

#Sea el mallado espacial para UNA MALLA IRREGULAR
x = np.zeros(nx)
y = np.zeros(ny)

#sean las líneas de punto medio
pmx = np.zeros(nx+1)
pmy = np.zeros(ny+1)

#Matrices que guardan el distanciamiento entre nodos
dx = np.zeros(nx+1)
dy = np.zeros(ny+1)

for i in range(nx+1):
    if i == 0:
        dx[i] = 0
    if i == 1:
        dx[i] = lx/nx
        print(lx/nx)
    else:
        dx[i] = dx[i-1]*1.1

for j in range(ny+1):
    if j == 0:
        dy[j] = 0
    if j == 1:
        dy[j] = ly/ny
    else:
        dy[j] = dy[j-1]*1.05

#Normalizamos el vector para la longitud
dx = (dx/sum(dx))*lx
dy = (dy/sum(dy))*ly

for i in range(nx+1):
    pmx[i] = pmx[i-1] + dx[i]

for j in range(ny+1):
    pmy[j] = pmy[j-1] + dy[j]

#Sean los centros de los volúmenes
for i in range(nx):
    x[i] = (pmx[i]+pmx[i+1])/2

for j in range(ny):
    y[j] = (pmy[j]+pmy[j+1])/2
```

0.04

```
#Sea la malla
X,Y = np.meshgrid(x,y)

fig, ax = plt.subplots(figsize=(16, 14))

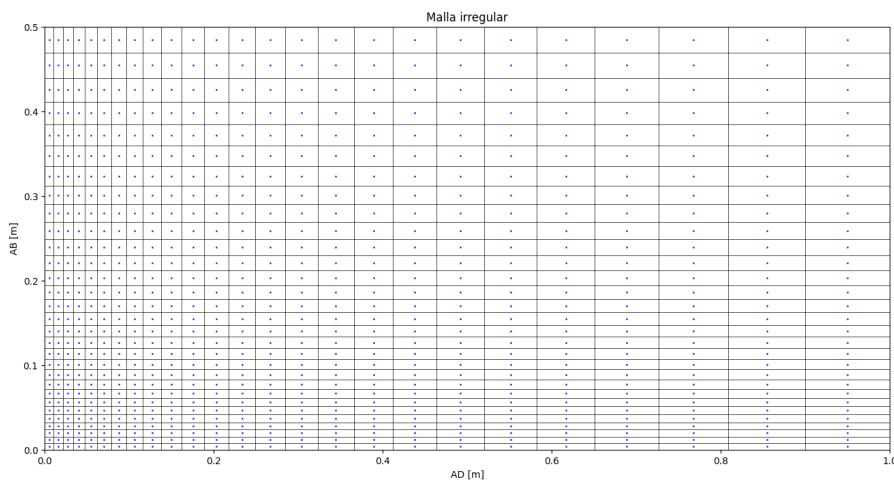
ax.vlines(pmx, ymin=0, ymax=ly, colors='black', linestyles='solid', linewidth=0.5, label='Líneas Verticales')
ax.hlines(pmy, xmin=0, xmax=lx, colors='black', linestyles='solid', linewidth=0.5, label='Líneas Horizontales')

ax.scatter(X,Y, color='blue', marker='o', s = 3, linewidth=0.1, label='Puntos')

Ti = np.zeros((ny,nx))
plt.imshow(Ti, cmap = 'Greys', interpolation='bilinear', extent=[0, lx, 0, ly])

plt.title('Malla irregular')
plt.xlabel('AD [m]')
plt.ylabel('AB [m]')

plt.show()
```



▼ Parámetros

```
#Dimensiones del canal
lx, ly, AE = 1, 0.5, 0.4

#Coeficiente de difusividad térmica del agua
D = 1*(10**-5)

#Temperaturas [°C]
t1, t2 = 1500, 50

#Velocidad inicial
v0 = 0.001

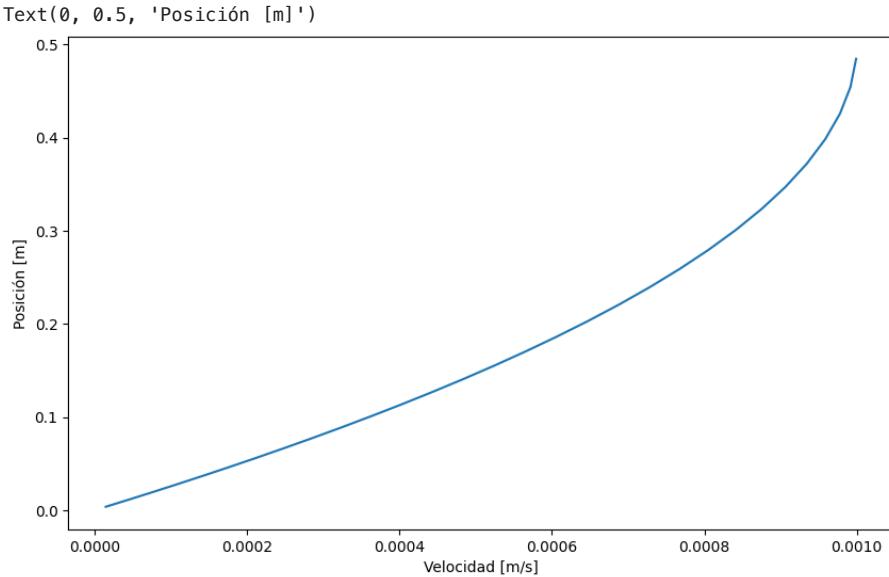
#Tiempo
nt = 1000
t = 1000
dt = t/nt

#Sea el mallado temporal
t = np.linspace(0, t, nt)

#Para aplicar la condición de borde
c = 0
for elementos in x:
    if elementos < AE:
        c = c+1
```

▼ Perfil inicial de velocidades

```
v = v0 * (1 - ((y - ly)**2)/((ly)**2))
plt.figure(figsize = (10,6))
plt.plot(v,y)
plt.xlabel('Velocidad [m/s]')
plt.ylabel('Posición [m]')
```



▼ Perfil inicial de temperaturas

```
#Perfil inicial
Ti = np.zeros((ny, nx))
Ti[:, :] = t2
Ti[0, 0:c] = t1
Ti[ny-1, 0:c] = t1
T = Ti.copy()
```

```

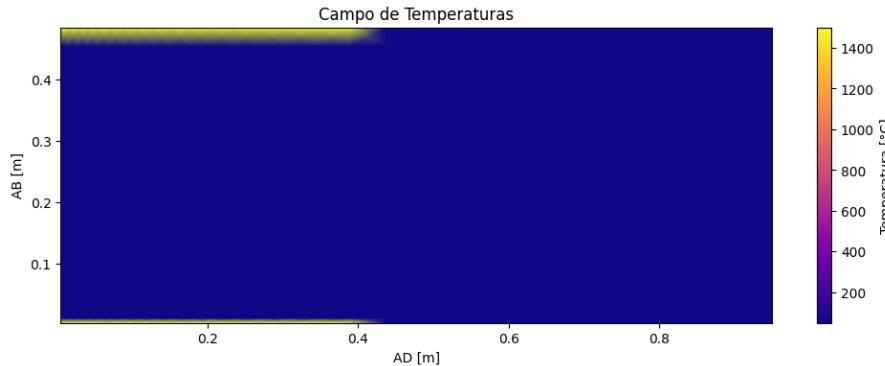
fig, ax = plt.subplots(figsize=(12, 4))

pc = plt.pcolormesh(x, y, Ti, cmap='plasma', shading = 'gouraud')

plt.title('Campo de Temperaturas')
plt.xlabel('AD [m]')
plt.ylabel('AB [m]')
plt.colorbar(pc, label='Temperatura [°C]')

plt.show()

```



▼ Discretización con UDS en la advección y difusión, y Euler explícito en el tiempo

Sea la ecuación general que rige al fenómeno:

$$\begin{aligned}
T_{i,j}^{n+1} &= T_{i,j}^n + \frac{dt}{(x_{i+\frac{1}{2},j} - x_{i-\frac{1}{2},j})(y_{i,j+\frac{1}{2}} - y_{i,j-\frac{1}{2}})} \\
&+ (D(\frac{T_{i+1,j}^n - T_{i,j}^n}{x_{i+1,j} - x_{i,j}})(y_{i,j+\frac{1}{2}} - y_{i,j-\frac{1}{2}}) - \frac{T_{i,j}^n - T_{i-1,j}^n}{x_{i,j} - x_{i-1,j}}(y_{i,j+\frac{1}{2}} - y_{i,j-\frac{1}{2}}) + \frac{T_{i,j+1}^n - T_{i,j}^n}{y_{i,j+1} - y_{i,j}}(x_{i+\frac{1}{2},j} - x_{i-\frac{1}{2},j}) \\
&- \frac{T_{i,j}^n - T_{i,j-1}^n}{y_{i,j} - y_{i,j-1}}(x_{i+\frac{1}{2},j} - x_{i-\frac{1}{2},j})) - u((y_{i+1,j} - y_{i,j})T_{i,j} - (y_{i,j} - y_{i-1})T_{i-1,j}))
\end{aligned}$$

```

def temp(T, ny, v0, nx, ly, y, D, x, pmy, pmx, t2, t1, dt, t, diferencia):
    while diferencia > 1e-1:
        Tn = T.copy()
        t = t+1
        for j in range((ny-1)):

            v = v0 * (1 - ((y[j] - ly)**2)/((ly)**2))

            for i in range((nx-1)):
                #FLUJOS EN TODAS LAS DIRECCIONES
                fe = D* ( ((Tn[j,i+1] - Tn[j,i])/(x[i+1]-x[i])) ) * (pmy[j+1]-pmy[j]) #FLUJO DIFUSIÓN ESTE
                fo = D* ( ((Tn[j,i] - Tn[j,i-1])/(x[i]-x[i-1])) ) * (pmy[j+1]-pmy[j]) #FLUJO DIFUSIÓN OESTE
                fn = D* ( ((Tn[j+1,i] - Tn[j,i])/(y[j+1]-y[j])) ) * (pmx[i+1]-pmx[i]) #FLUJO DIFUSIÓN NORTE
                fs = D* ( ((Tn[j,i] - Tn[j-1,i])/(y[j]-y[j-1])) ) * (pmx[i+1]-pmx[i]) #FLUJO DIFUSIÓN SUR

                #NODO DIAGONAL INFERIOR IZQUIERDO
                if i == 0 and j == 0:
                    Tn[j, i-1] = t2
                    Tn[j-1,i] = t1
                    dif = fe - fo + fn - fs

                #NODO DIAGONAL INFERIOR DERECHO
                elif i == (nx-1) and j == 0:
                    if i < c:
                        Tn[j-1, i] = t1
                    else:
                        Tn[j-1, i] = t2
                    T[j, i+1] = ((x[i+1] - x[i])/((x[i]-x[i-1])))*(Tn[j,i]-Tn[j-1,i]) + Tn[[j,i]]
                    dif = fe - fo + fn - fs

                #NODO DIAGONAL SUPERIOR IZQUIERDO
                elif i == 0 and j == (ny-1):
                    Tn[j, i-1] = t2
                    Tn[j+1,i] = t1
                    dif = fe - fo + fn - fs

                #NODO DIAGONAL SUPERIOR DERECHO
                elif i == (nx-1) and j == (ny-1):
                    T[j, i+1] = ((x[i+1] - x[i])/((x[i]-x[i-1])))*(Tn[j,i]-Tn[j-1,i]) + Tn[[j,i]]
                    dif = fe - fo - fs

                #COLUMNA OESTE
                elif i == 0 and 0 < j < (ny-1):
                    Tn[j, i-1] = t2
                    dif = fe - fo + fn - fs

                #COLUMNA ESTE
                elif i == (nx-1) and 0 < j < (ny-1):
                    T[j, i+1] = ((x[i+1] - x[i])/((x[i]-x[i-1])))*(Tn[j,i]-Tn[j-1,i]) + Tn[[j,i]]
                    dif = fe - fo + fn - fs

                #FILA SUR
                elif j == 0 and 0 < i < (nx-1):
                    if i < c:
                        Tn[j-1, i] = t1
                    else:
                        Tn[j-1, i] = t2
                    dif = fe - fo + fn - fs

                #FILA NORTE
                elif j == (ny-1) and 0 < i < (nx-1):
                    if i < c:
                        Tn[j+1, i] = t1
                    else:
                        Tn[j+1, i] = t2
                    dif = fe - fo - fs + fn

                else:
                    dif = fe - fo + fn - fs

                adv = v* ( (y[j+1] - y[j])*Tn[j,i] - (y[j]-y[j-1])*Tn[j,i-1] ) #ADVECCIÓN
                T[j,i] = Tn[j,i] + (dt/ ((pmx[i+1]-pmx[i])*(pmy[j+1]-pmy[j)))) * (dif - adv)

        T[1:, 0] = t2
        T[0, 0:c] = t1
        T[0, c:] = t2
        T[ny-1, 0:c] = t1

```

```

T[ny-1, c:] = t2
diferencia = np.sum(np.abs(T - Tn))
return T, t

T = Ti.copy()
diferencia = 1
t = 0
T, t = temp(T, ny, v0, nx, ly, y, D, x, pmy, pmx, t2, t1, dt, t,diferencia)
print(t)

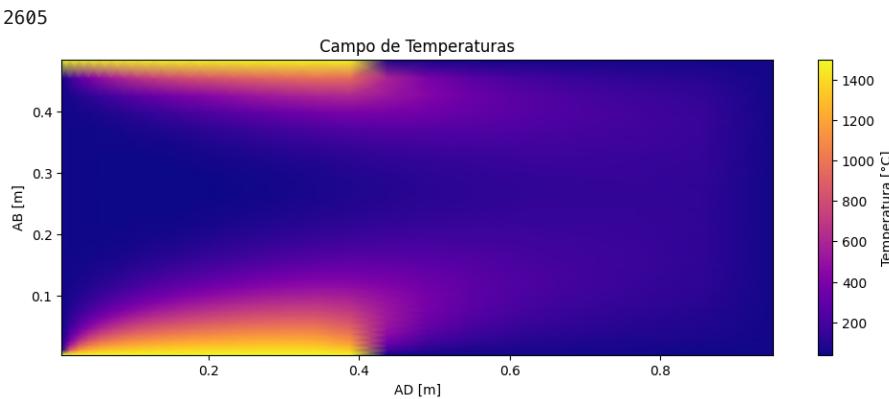
fig, ax = plt.subplots(figsize=(12, 4))

pc = plt.pcolormesh(x, y, T, cmap='plasma', shading = 'gouraud')

plt.title('Campo de Temperaturas')
plt.xlabel('AD [m]')
plt.ylabel('AB [m]')
plt.colorbar(pc, label='Temperatura [°C]')

plt.show()

```



Validación del código

Si el fluido sólo sufre advección con una velocidad constante de v_0 en el eje x, se tiene el siguiente código (Basado en las ecuaciones anteriormente descritas):

▼ Parámetros

```
#Dimensiones del canal
lx, ly, AE = 1, 0.5, 0.2

#Coeficiente de difusividad térmica del agua
D = 1*(10**-5)

#Temperaturas [°C]
t1, t2 = 500, 50

#Velocidad inicial
v0 = 0.0004

#Sea la cantidad de volúmenes en 1d
nx = 25

#Sea el mallado espacial para UNA MALLA IRREGULAR
x = np.zeros(nx)
y = np.zeros(ny)

#sean las líneas de punto medio
pmx = np.zeros(nx+1)
pmy = np.zeros(ny+1)

#Matrices que guardan el distanciamiento entre nodos
dx = np.zeros(nx+1)
dy = np.zeros(ny+1)

for i in range(nx+1):
    if i == 0:
        dx[i] = 0
    if i == 1:
        dx[i] = 0.04
    else:
        dx[i] = dx[i-1]*1.001

for j in range(ny+1):
    if j == 0:
        dy[j] = 0
    if j == 1:
        dy[j] = 0.0166666666
    else:
        dy[j] = dy[j-1]*1.006

for i in range(nx+1):
    pmx[i] = pmx[i-1] + dx[i]

for j in range(ny+1):
    pmy[j] = pmy[j-1] + dy[j]

#Sean los centros de los volúmenes
for i in range(nx):
    x[i] = (pmx[i]+pmx[i+1])/2

for j in range(ny):
    y[j] = (pmy[j]+pmy[j+1])/2
```

```

def ADV1Dx(t, nx, v, dt, dx, Tadv):
    for ti in t:
        Tn = Tadv.copy()
        for i in range(1, nx-1):
            Tadv[i] = Tn[i] - (dt/dx[i+1])*v*(Tn[i] - Tn[i-1])
    return Tadv

def DIF1Dx(t, nx, dt, dx, D, Tdif, x):
    for ti in t:
        Tn = Tdif.copy()
        for i in range(1, nx-1):
            Tdif[i] = Tn[i] + (dt/(dx[i+1])) * D*((Tn[i+1] - Tn[i])/(x[i+1]-x[i])) - ((Tn[i] - Tn[i-1])/(x[i]-x[i-1]))
    return Tdif

def DIF1Dy(t, nx, dt, dy, D, Tdif, y):
    for ti in t:
        Tn = Tdif.copy()
        for j in range(1, ny-1):
            Tdif[j] = Tn[j] + (dt/(dy[j+1])) * D*((Tn[j+1] - Tn[j])/(y[j+1]-y[j])) - ((Tn[j] - Tn[j-1])/(y[j]-y[j-1]))
    return Tdif

```

Haz doble clic (o pulsa Intro) para editar

```

fig, axs = plt.subplots(1, 3, figsize=(18, 6))

nt = [5, 10, 15]
# Gráfica para Advección en x
axs[0].set_title('Advección 1D en eje x')
for n in nt:
    t = 500
    dt = t / n
    t = np.linspace(0, t, n)
    CFL = (v0 * dt) / dx[1]
    Tadv = np.zeros(nx)
    Tadv[:1] = t1
    Tadv[1:] = t2
    Tadv = ADV1Dx(t, nx, v0, dt, dx, Tadv)
    axs[0].plot(x, Tadv, label=f'Advección para CFL={np.round(CFL, 2)}')
axs[0].legend()

nt = [10, 20, 40]
# Gráfica para Difusión en x
axs[1].set_title('Difusión 1D en eje x')

Tdify = np.zeros(nx)
Tdify[:c] = t1
Tdify[c:] = t2
legend_added = False
for n in nt:
    t = 500
    dt = t / n
    t = np.linspace(0, t, n)
    r = dt * D / (dx[1] ** 2)
    Tdify[:c] = t1
    Tdify = DIF1Dx(t, nx, dt, dx, D, Tdify, x)
    if not legend_added:
        Tan = np.zeros(nx)
        erf_vectorized = np.vectorize(math.erf)
        Tan = erf_vectorized((0.5 * x) / ((D * 1000) ** 0.5)) * (t2 - t1) + t1
        Tan[:c] = t1
        axs[1].plot(x, Tan, linestyle='--', label='Solución Analítica')
        legend_added = True

    axs[1].plot(x, Tdify, label=f'Difusión en x para r={np.round(r, 2)}')
#analítica
axs[1].legend()

nt = [60, 120, 240]
# Gráfica para Difusión en y
Tdify = np.zeros(ny)
Tdify[0] = t1
Tdify[1:] = t2
axs[2].set_title('Difusión 1D en eje y')
for n in nt:
    t = 500

```

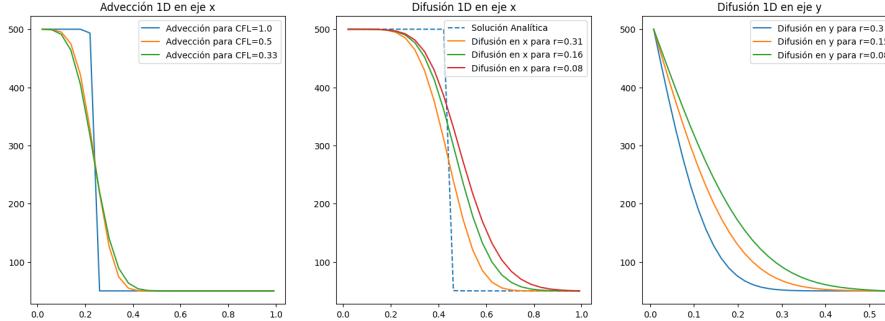
```

dt = t / n
t = np.linspace(0, t, n)
r = dt * D / (dy[1] ** 2)
Tdify[0] = t1
Tdify = DIF1Dy(t, nx, dt, dy, D, Tdify, y)

axs[2].plot(y, Tdify, label=f'Difusión en y para r={np.round(r, 2)}')
axs[2].legend()

plt.show()

```



▼ Tiempos de convergencia

```

def ADV1Dx(t, nx, v, dt, dx, Tadv, dif):
    while dif > 0.2:
        t = t+1
        Tn = Tadv.copy()
        for i in range(1, nx-1):
            Tadv[i] = Tn[i] - (dt/dx[i+1])*v*(Tn[i] - Tn[i-1])
        dif = np.sum(np.abs(Tadv - Tn))
        Tadv[:c] = t1
    return Tadv, t

def DIF1Dx(t, nx, dt, dx, D, Tdif, x, dif):
    while dif > 0.2:
        t = t+1
        Tn = Tdif.copy()
        for i in range(1, nx-1):
            Tdif[i] = Tn[i] + (dt/(dx[i+1]))*D*((Tn[i+1] - Tn[i])/(x[i+1]-x[i]) - (Tn[i] - Tn[i-1])/(x[i]-x[i-1]))
        dif = np.sum(np.abs(Tdif - Tn))
        Tdif[:c] = t1
        Tdif[-1] = t2
    return Tdif, t

def DIF1Dy(t, nx, dt, dy, D, Tdif, y, dif):
    while dif > 0.2:
        t = t+1
        Tn = Tdif.copy()
        for j in range(1, ny-1):
            Tdif[j] = Tn[j] + dt/(dy[j+1]) *D*((Tn[j+1] - Tn[j])/(y[j+1]-y[j]) - (Tn[j] - Tn[j-1])/(y[j]-y[j-1]))
        dif = np.sum(np.abs(Tdif - Tn))
        Tdif[0] = t1
        Tdif[-1] = t2
    return Tdif, t

```

```
fig, axs = plt.subplots(1, 3, figsize=(18, 6))

CFL = (v0 * dt) / dx[1]
Tadv = np.zeros(nx)
Tadv[:c] = t1
Tadv[c:] = t2
dif = 1
t = 0
Tadv, t = ADV1Dx(t, nx, v0, dt, dx, Tadv, dif)
print(t)
axs[0].plot(x, Tadv, label=f'Advección para CFL={np.round(CFL, 2)}')
axs[0].set_title('Malla irregular')
axs[0].legend()

r = (dt * D) / (dx[1] ** 2)
Tdifx = np.zeros(nx)
Tdifx[:c] = t1
Tdifx[c:] = t2
dif = 1
t = 0
```