



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Diseño de una instalación experimental de caracterización de la transmisión de calor por convección de aire en conductos.

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

Autor: Álvaro González Sánchez.

Director: Francisco Javier Sánchez Velasco.

Codirector: Ramón A. Otón Martínez.



Universidad
Politécnica
de Cartagena

Cartagena, 2020

ÍNDICE DE CONTENIDOS

CAPÍTULO 1. INTRODUCCIÓN.

1.1- Transferencia de Calor por Convección.....	4
1.1.1-Transferencia de calor por convección forzada.....	4
1.1.2-Convección forzada en el interior de un tubo para los casos de flujo de calor contante ($q''=cte$) y temperatura constante en la pared ($T_s = cte.$).....	6
1.1.3-Cálculo del coeficiente de transmisión del calor h	8
1.1.4-Conceptos importantes en transferencia de calor por convección forzada en flujo interno...8	
1.1.5-Transferencia de calor por convección forzada en régimen turbulento.....	9
1.1.6-Correlaciones existentes en la bibliografía para transferencia de calor por convección forzada para flujo interno.....	11
1.2- Objetivos Trabajo Fin de Grado.....	14
1.3- Fases del trabajo.....	15

CAPÍTULO 2. INSTALACIÓN EXPERIMENTAL.

2.1- Definición de la instalación experimental y descripción instrumentación experimental.	16
2.2- Variables fluidodinámicas a medir.....	18
2.2.2- Área de mojado y área transversal.....	19
2.2.3- Temperatura superficial interior y exterior del tubo.....	19
2.2.4- Temperatura del fluido a la entrada.....	19
2.2.5- Temperatura del fluido a la salida.....	20
2.2.6- Temperatura media del fluido.....	20
2.2.7- Caudal másico.....	20
2.2.8- Calor específico.....	21
2.3- Descripción de equipos auxiliares para suministro de potencia y medida.....	21
2.3.1- Termopares.....	21
2.3.2- Manta eléctrica.....	21
2.3.3- Multímetro.....	21
2.3.4- Sensor de presión diferencial.....	22
2.3.5- Aislante exterior.....	22

CAPÍTULO 3. MONTAJE Y PUESTA A PUNTO DE LAS INSTALACIONES23

CAPÍTULO 4. SIMULACIÓN NUMÉRICA DEL PROBLEMA CON OPENFOAM.

4.1- Incidencia sobre el cambio de estrategia en la resolución del caso.....	24
4.2- Objetivos a cumplir con OpenFoam.....	24
4.3- Ecuaciones del modelo físico resuelto con Open Foam.....	24
4.3.1- Ecuaciones RANS: Reynolds Averaged Navier-Stokes y el modelo k-epsilon.....	24
4.3.2- Resolución por el método de volúmenes finitos.....	25
4.4- Introducción para el uso de OpenFoam.....	26
4.4.1- Descarga del programa y accesos directos.....	26
4.4.2- Pasos previos antes de simular un caso.....	26
4.4.3- Comandos más comunes y su utilidad.....	27
4.4.4- Paraview y sus aplicaciones más usadas.....	28
4.4.5- Tutorial para entender OpenFoam.....	30
4.5- Generación de la malla.....	36
4.6- Condiciones de contorno.....	41
4.6.1- Tutorial con SimpleFoam.....	41
4.6.1.1- k- epsilon model.....	45

4.6.1.2- k- omega model.....	53
4.6.1.3- Comparación entre modelos.....	55
4.6.2.- Tutorial “buoyantBoussinesqPimpleFoam” para introducir condiciones de contorno.....	61
CAPÍTULO 5. RESULTADOS DEL ANÁLISIS DEL PROBLEMA.	
5.1- Preparación del caso para simulación del problema de transmisión de calor por convección en el interior de un tubo.....	69
5.2- Análisis de resultados del caso velocidad de entrada 2 m/s con 75% de la potencia y $dT/dr=118.68^{\circ}C/m$.....	71
5.3- Análisis de resultados del caso velocidad de entrada 6.66 m/s con 75% de la potencia y $dT/dr=118.68^{\circ}C/m$.....	73
5.4- Análisis de resultados del caso velocidad de entrada 6.66 m/s con 95% de la potencia y $dT/dr=118.68^{\circ}C/m$.....	76
5.5- Análisis de resultados del caso con velocidad teórica de 6.66m/s con 75% de la potencia y $dT/dr=99651.94^{\circ}k/m$.....	78
5.6- Influencia de la velocidad de entrada y del gradiente de temperaturas.....	81
5.6.1- Temperatura media a la salida.....	82
5.6.2-Espesor capa límite térmica.....	82
5.6.3- Temperatura teórica media a la salida.....	83
5.6.4- Coeficiente de transmisión de calor y número de Nusselt.....	84
CAPÍTULO 6. CONCLUSIÓN DEL TRABAJO.....	88
CAPÍTULO 7. BIBLIOGRAFÍA.....	89
CAPÍTULO 8. ANEXOS.....	90

CAPÍTULO 1. INTRODUCCIÓN.

1.1- Transferencia de Calor por Convección.

La convección es conocida como “*el proceso de transferencia de energía entre un sólido en contacto con un fluido debido al movimiento de este*” (Madrid y Navarro, 2012, 112).

Se puede distinguir entre convección natural y convección forzada, y distinguir entre convección por flujo interno, que encontramos cuando introducimos un fluido con una velocidad y a una temperatura específicas en un conducto, y flujo externo, que se da cuando este mismo fluido se mueve por la superficie exterior de un cuerpo a tratar.

Tradicionalmente, la transferencia de calor por convección se ha modelado a través de la ecuación de enfriamiento de Newton:

$$q = h \cdot A \cdot (T_s - T_f) \quad (1.1)$$

En la ecuación que representa la Ley de enfriamiento de Newton, está presente el coeficiente de transmisión de calor por convección, h , también llamado coeficiente de película. T_s se refiere a la temperatura de la superficie sólida en contacto con el fluido, T_f es la temperatura del fluido fuera de la capa límite y A es el área correspondiente

En el inicio del estudio de la transmisión de calor por convección este coeficiente se suponía conocido. Esta suposición atrasó los estudios sobre la convección, ya que este valor depende de más variables relativas a la geometría y el movimiento del fluido. En realidad, h no es un coeficiente sino una función que depende de muchos parámetros, y que encapsula todo el proceso de transferencia de energía entre fluido y superficie. Por tanto, para su determinación de manera precisa se requiere la resolución de las ecuaciones de conservación de energía, masa y cantidad de movimiento para el fluido, las llamadas ecuaciones de Navier-Stokes, con la aplicación de las condiciones de contorno e iniciales correspondientes.

Por tanto, el tratamiento sobre mencionado coeficiente de calor solo se puede obtener a través de soluciones analíticas en geometrías muy sencillas, debido a la dificultad de este estudio. En el caso de flujo en régimen turbulento, las ecuaciones se complican pues es necesario modelar la turbulencia en el fluido. La turbulencia por definición es caótica, es decir, la determinación de la solución de las ecuaciones de la dinámica del fluido en régimen turbulento depende de instantes de tiempo posteriores al actual y por tanto requieren un tratamiento de tipo estadístico. En nuestro caso intentaremos calcular el coeficiente h en una de estas geometrías más sencillas como puede ser la de un tubo circular.

1.1.1-Transferencia de calor por convección forzada.

Podemos definir dicha convección de la siguiente manera:

“La convección forzada tiene lugar cuando una fuerza motriz exterior mueve un fluido sobre una superficie sólida a una temperatura mayor o menor que el fluido” (Madrid y Navarro, 2012, 112).

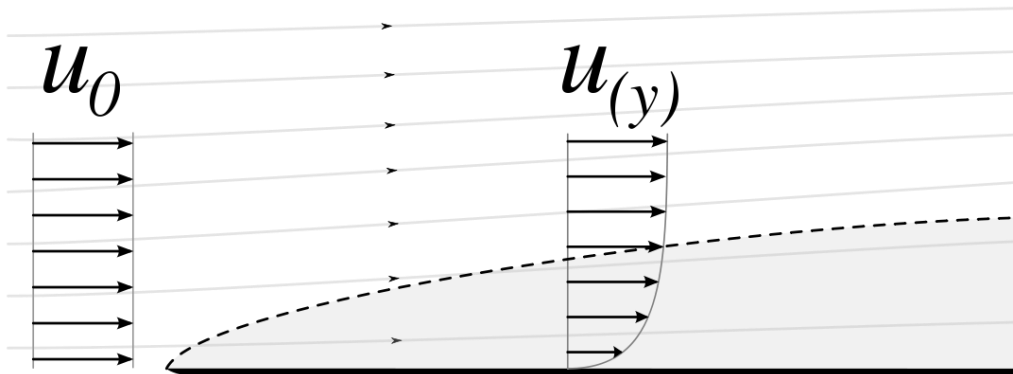
En nuestro caso a estudiar, el fluido de trabajo será aire, la fuerza estará ejercida por un ventilador. Las especificaciones del mismo serán detalladas más adelante en este documento.

Se puede especificar dentro de esta definición que una corriente es creada en general por diferencias de presiones, que mediremos. Cuando la gravedad es despreciable frente a estas fuerzas de presión es cuando ocurre la convección forzada pura. El fluido intercambia calor con la

superficie, es decir, puede ser calentado o enfriado por dicha superficie.

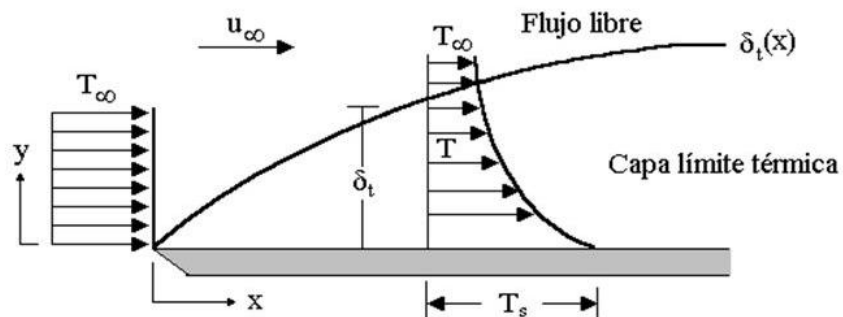
La potencia calorífica transferida q (W) está definida mediante la ley de enfriamiento de Newton (Ecuación 1.1). Esta transferencia tiene lugar en la capa límite.

Cuando hablamos de capa límite nos referimos al efecto que produce la viscosidad del fluido en una zona cercana a la pared del cuerpo, esto afecta a que la velocidad inicial en el punto de creación de la capa será 0, e irá aumentando progresivamente a medida que avancemos. Hay un punto en el que el efecto de la viscosidad es insignificante.



Capa límite laminar. [Wikipedia enciclopedia libre]1.1

Al igual que este caso que afecta a la velocidad del fluido, afecta al gradiente de temperatura, creando una capa límite térmica cerca de las paredes.



Capa límite térmica. [Introducción a la convección por Reina Payan]1.2

1.1.2-Convección forzada en el interior de un tubo para los casos de flujo de calor constante ($q''=cte$) y temperatura constante en la pared ($T_s = cte$.)

En nuestro caso a estudiar tendremos que hacer un balance de energía con la especificación de flujo constante de calor en la superficie, ya que tendremos una superficie rodeada por una manta eléctrica que aportará este calor por igual al tubo ($q'' = cte \text{ W/m}^2$). En la bibliografía a este flujo de calor también se le suele denominar \dot{q}_s .

Se podrá calcular el calor transmitido en la unidad de tiempo con la siguiente expresión:

$$q = q'' \cdot A = \dot{m} \cdot c_p \cdot (T_{fe} - T_{fs}) = h \cdot A \cdot (T_s - T_m) \quad (1.2)$$

En este caso mediremos la temperatura del fluido a la entrada y a la salida, \dot{m} se refiere al caudal másico del fluido (kg/s), c_p al calor específico del mismo (J/(kg·K)) y T_s, T_m, T_{fe}, T_{fs} se refieren a la temperatura de la superficie, a la media del fluido, a la del fluido a la entrada y la del fluido a la salida respectivamente (en °C ó K). Esta última también se llama temperatura de mezcla y se suele representar en la bibliografía como T_b .

Podemos utilizar la definición de la potencia calorífica en función del salto de temperaturas logarítmicas medias, ΔT_m , como:

$$q = h \cdot A \cdot \Delta T_m \quad (1.3)$$

En la zona de entrada hasta una longitud térmica, que definiremos posteriormente, tendremos un coeficiente de transmisión de calor que será función de x . Es a partir de esa longitud donde tanto el gradiente de temperaturas como dicho coeficiente h se vuelven constantes.

En este caso de flujo de calor constante podemos establecer las siguientes relaciones:

$$q = q''A = \dot{m} \cdot c_p \cdot (T_{fe} - T_{fs}) \rightarrow T_{fs} = T_{fe} + \frac{q''A}{\dot{m} \cdot c_p} ; q'' = h_x(T_s - T_m) \rightarrow T_s = T_m + \frac{q''}{h_x} \quad (1.4)$$

Teniendo el valor del calor del flujo podremos definir la temperatura de la superficie y la temperatura de salida.

Si realizamos un balance de energía en un volumen de control definido por dx del conducto, tendremos:

$$\dot{m}c_p T_m = q''(pdx) \rightarrow \frac{dT_m}{dx} = \frac{q''p}{\dot{m}c_p}, \text{ si } q'', h \rightarrow cte \quad (1.5)$$

Esto nos indica que la variación temperatura de mezcla con x sigue una ley lineal para el caso de un flujo de calor constante en la pared. Si el flujo está totalmente desarrollado térmicamente, h no será función de x y por tanto la derivada de T_b respecto de x será igual a la derivada de la temperatura de la superficie respecto de x . En la región de entrada térmica h será función de x y por tanto, en base a la ecuación 1.4, la derivada de T_b respecto de x no será igual a la derivada de T_{sup} respecto de x . Esto se demuestra si tenemos en cuenta que se debe cumplir que la temperatura adimensional T permanece constante en la región del flujo totalmente desarrollado, obteniendo:

$$\frac{\partial}{\partial x} \left(\frac{T_s - T}{T_s - T_m} \right) = 0 \rightarrow \frac{1}{T_s - T_m} \left(\frac{\partial T_s}{\partial T_x} - \frac{\partial T}{\partial x} \right) = 0 \rightarrow \frac{\partial T_s}{\partial T_x} = \frac{\partial T}{\partial x} ; \frac{\partial T_s}{\partial T_x} = \frac{\partial T}{\partial x} = \frac{\partial T_m}{\partial T_x} = \frac{q''p}{\dot{m}c_p} = cte \quad (1.6)$$

Esta expresión demuestra que en flujo completamente desarrollado sujeto a un flujo de calor constante en su superficie, el gradiente de temperaturas es independiente de x y así, el perfil no cambia a lo largo del tubo.

En cuanto al caso de temperatura constante en la pared ($T_s = \text{cte}$), podemos obtener la solución cogiendo un diferencial de área dA con una variación de espesor dx de área en el conducto, tendremos una variación de temperaturas que sabiendo que el flujo tiene la forma expresada anteriormente:

$$\dot{m}c_p dT_m = h(T_s - T_m)dA \tag{1.7}$$

Donde sabemos que:

$$\frac{d(T_s - T_m)}{(T_s - T_m)} = -\frac{hp}{\dot{m}c_p} dx \tag{1.8}$$

Cogeremos la diferencial de la temperatura e integraremos la parte derecha de la igualdad respecto a x :

$$\int_{T_s - T_{fe}}^{T_s - T_m} \frac{d(T_s - T_m)}{(T_s - T_m)} = -\frac{hp}{\dot{m}c_p} \int_0^x dx \rightarrow T_{fs} = T_s - (T_s - T_{fe})e^{-\frac{hAp}{\dot{m}c_p}} \tag{1.9}$$

Podemos poner entonces:

$$\ln \frac{(T_s - T_{fs})}{(T_s - T_{fe})} = -\frac{hAp}{\dot{m}c_p} \tag{1.10}$$

Si despejamos de aquí $\dot{m}c_p$ y lo introducimos en (1.2) obtendremos la definición de diferencia de temperatura logarítmica media.

Suele considerarse como diferencia media de temperaturas, ΔT_m , la denominada diferencia de temperatura logarítmica media definida como:

$$\Delta T_{ln} = \frac{\Delta T_e - \Delta T_s}{\ln \frac{\Delta T_e}{\Delta T_s}} \tag{1.11}$$

siendo $\Delta T_e = T_s - T_{fe}$ y $\Delta T_s = T_s - T_{fs}$.

También podría considerarse como ΔT_m la media aritmética de la diferencia de temperaturas:

$$\Delta T_{ma} = \frac{\Delta T_e + \Delta T_s}{2} \tag{1.12}$$

Siempre que $\Delta T_{ln} < \Delta T_{ma}$ y si la razón entre ΔT_e y ΔT_s no es mayor de 0,5 la ΔT_{ln} puede aproximarse a ΔT_{ma} con un error del 1,4%.

Todo esto será comprobado posteriormente con las simulaciones realizadas.

1.1.3-Cálculo del coeficiente de transmisión del calor h .

Para calcular este coeficiente podemos enfocarlo de dos formas, la primera sería utilizando los parámetros adimensionales presentes en la convección como son el número de Reynolds y el número de Prandlt, de los que depende el número que nos permitiría calcular h , el de Nusselt dado que se puede definir como:

$$Nu = \frac{h \cdot x}{k_f} \tag{1.13}$$

Donde x es la longitud característica geométrica del problema (para un tubo es el diámetro) y k_f la conductividad térmica del fluido.

Y despejando nos quedaría:

$$h = \frac{Nu \cdot k_f}{x} \tag{1.14}$$

Con esta expresión se puede obtener h a partir de correlaciones de Nusselt de la bibliografía. La utilizaremos para comparar los resultados de nuestras estimaciones de h con las de la bibliografía.

Sin embargo, dispondremos del valor de q'' mediante la potencia eléctrica aportada para calentar el tubo. Valor que será comprobado con diferentes mediciones y atendiendo a la relación previamente descrita:

$$q'' \cdot A = \dot{m} \cdot c_p \cdot (T_{fe} - T_{fs}) \tag{1.15}$$

Por lo que podemos despejar de una forma más sencilla la h :

$$h = \frac{q}{A \cdot (T_s - T_m)} = \frac{q}{A \cdot \Delta T_m} \tag{1.16}$$

Siendo T_s y T_m ,la temperatura superficial y la media del fluido respectivamente en una determinada sección del tubo. Ésta última la calcularemos de la siguiente forma:

$$T_m = \frac{T_{fe} + T_{fs}}{2} \tag{1.17}$$

1.1.4-Conceptos importantes en transferencia de calor por convección forzada en flujo interno.

Podemos definir los siguientes conceptos:

- Velocidad media en flujo interno en cualquier sección transversal, que es la que solemos usar en los cálculos. En un conducto la velocidad es dependiente del radio, ya que en las paredes la velocidad se hace nula, y se crea un perfil de velocidades con el máximo situado en $r=0$, la podemos expresar:

$$U_m = \frac{1}{A} \int_{Area} u dA \tag{1.18}$$

- Para la temperatura, de forma similar, se crea un perfil de temperaturas, y usaremos de nuevo una variable media, que definimos como:

$$T_m = \frac{\int_{Area} c_p \rho T(r) U(r) dA}{c_p \dot{m}} \quad (1.19)$$

- El número de Reynolds es el parámetro que define el régimen que se encuentra el flujo, que depende de la velocidad, la viscosidad y el diámetro del conducto:

$$Re = \frac{U_m D}{\nu} \quad (1.20)$$

Si no tenemos un conducto de sección circular deberemos calcular el diámetro equivalente llamado diámetro hidráulico.

$$Re = \frac{U_m D_h}{\nu} \quad (1.21)$$

$$D_h = 4 \frac{A}{P} \quad (1.22)$$

Donde A es el área de la sección y P el perímetro mojado.

Podremos definir, en función del número de Reynolds, tres regímenes.

1. $Re < 2300$. Se trata del régimen laminar.
 2. $2300 < Re < 10000$. Régimen de transición.
 3. $Re > 10000$. Régimen turbulento.
- Longitud hidráulica de entrada, de adaptación a la capa límite.

$$L_{h,laminar} = 0.05 Re D \quad (1.23)$$

$$L_{h,turbulento} = 1.359 Re^{1/4} \quad (1.24)$$

- Longitud térmica de entrada.

$$L_{t,laminar} = 0.05 Re D Pr ; L_{t,turbulento} = 10 D \quad (1.25)$$

1.1.5-Transferencia de calor por convección forzada en régimen turbulento.

En el problema que vamos a estudiar tenemos régimen turbulento.

Las ecuaciones que utilizamos en este trabajo son del tipo RANS, que se obtienen mediante el promediado de las ecuaciones de Navier-Stokes en una escala mayor que la de turbulencia. Por lo tanto, las ecuaciones resultantes describen la velocidad, presión, temperatura y transporte medio en el flujo, pero no los detalles de las fluctuaciones turbulentas. Una cantidad instantánea se descompone en sus cantidades promediadas en el tiempo y fluctuantes. El efecto de la turbulencia en el flujo medio se introduce a través de los tensores de Reynolds.

Las ecuaciones integrales de conservación de la masa, cantidad de movimiento y energía para un volumen de control $V_c(t)$ son:

$$\frac{d}{dt} \int_{V_c(t)} \rho dV + \int_{\Sigma_c(t)} \rho (\vec{v} - \vec{v}_c) \cdot \vec{n} d\sigma = 0, \quad (1.26)$$

$$\frac{d}{dt} \int_{V_c(t)} \rho \vec{v} dV + \int_{\Sigma_c(t)} \rho \vec{v} (\vec{v} - \vec{v}_c) \cdot \vec{n} d\sigma = - \int_{\Sigma_c(t)} p \vec{n} d\sigma + \int_{\Sigma_c(t)} \tau' \cdot \vec{n} d\sigma + \int_{V_c(t)} \rho \vec{f}_m dV \quad (1.27)$$

$$\frac{d}{dt} \int_{V_c(t)} \rho \left(e + \frac{1}{2} v^2 \right) dV + \int_{\Sigma_c(t)} \rho \left(e + \frac{1}{2} v^2 \right) (\bar{v} - \bar{v}_c) \cdot \bar{n} d\sigma = - \int_{\Sigma_c(t)} \bar{v} \cdot \bar{p} \bar{n} d\sigma + \int_{\Sigma_c(t)} \bar{v} \cdot \bar{\tau}' \cdot \bar{n} d\sigma + \int_{V_c(t)} \rho \bar{f}_m \cdot \bar{v} dV - \int_{\Sigma_c(t)} \bar{q} \cdot \bar{n} d\sigma + \int_{V_c(t)} Q_r dV \quad (1.28)$$

Poniendo estas ecuaciones en forma diferencial:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \quad (1.29)$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_j u_i) = - \frac{\partial \rho}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \quad (1.30)$$

$$\frac{\partial}{\partial t} (\rho E) + \frac{\partial}{\partial x_j} (\rho u_j H) = \frac{\partial \rho}{\partial x_j} (u_j \tau_{ij}) + \frac{\partial}{\partial x_j} \left(k \frac{\partial T}{\partial x_j} \right) \quad (1.31)$$

Donde el término u representa la componente velocidad. Los componentes del tensor de esfuerzos viscosos, τ_{ij} , los podemos expresar como:

$$\tau_{ij} = 2\mu S_{ij} + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij} = 2\mu S_{ij} - \left(\frac{2\mu}{3} \right) \frac{\partial u_k}{\partial x_k} \delta_{ij} \quad (1.32)$$

Los componentes del tensor de deformaciones:

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (1.33)$$

Definimos así mismo el tensor de rotación como:

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (1.34)$$

La energía total E y la entalpía total H se expresan como:

$$E = e + \frac{1}{2} u_i u_i \quad (1.35)$$

$$H = h + \frac{1}{2} u_i u_i \quad (1.36)$$

Descomponiendo las variables en una parte media y otra turbulenta, y considerando las propiedades de operaciones con los términos fluctuantes, podemos obtener las siguientes relaciones para la conservación de la masa y la cantidad de movimiento:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (1.37)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} (\bar{\tau}_{ij} - \rho \overline{u'_i u'_j}) \quad (1.38)$$

Con

$$\bar{\tau}_{ij} = \mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (1.39)$$

Cuando tenemos en cuenta las fluctuaciones, es el tensor de esfuerzo de Reynolds el que representa la transferencia de momento debido a las fluctuaciones turbulentas, definido como:

$$\tau_{ij}^R = \rho \overline{u'_i u'_j} \quad (1.40)$$

A la hora de resolver las ecuaciones del flujo principal, es necesario emplear un modelo de

turbulencia para tener en cuenta los esfuerzos turbulentos. La solución es establecer una analogía entre el transporte viscoso y el turbulento definiendo una viscosidad turbulenta, μ_t , que simplemente se añade a la viscosidad molecular. Esta idea es la que sustenta el modelo de “Eddy viscosity”, que toma el esfuerzo proporcional a la deformación, es decir:

$$\tau_{turb} = \mu_t \frac{\partial \bar{u}}{\partial y} \quad (1.41)$$

Donde μ_t es la llamada Eddy viscosity o viscosidad turbulenta.

De este modo, el esfuerzo cortante resulta:

$$\tau = (\mu + \mu_t) \frac{\partial \bar{u}}{\partial y} \quad (1.42)$$

Las ecuaciones anteriores no se pueden resolver analíticamente y que, por tanto, es preciso encontrar soluciones parciales a las mismas mediante aproximaciones experimentales o numéricas, como haremos en este trabajo.

1.1.6-Correlaciones existentes en la bibliografía para transferencia de calor por convección forzada para flujo interno.

Flujo de calor constante

- Conducto circular

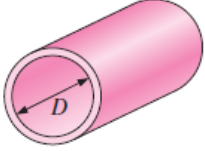
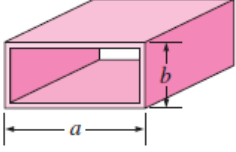
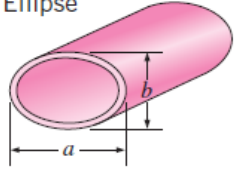
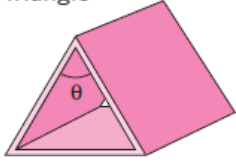
$$h = 4.36 \frac{k}{D}; \quad Nu = 4.36 \quad (1.43)$$

- Temperatura constante

$$h = 3.66 \frac{k}{D}; \quad Nu = 3.66 \quad (1.44)$$

- Región de entrada térmica. Coeficiente de convección en la región de entrada con flujo laminar

$$Nu = 3.66 + \frac{0.065(D/L) Re Pr}{1 + 0.04[(D/L) Re Pr]^{2/3}} \quad (1.45)$$

Tube Geometry	ab or θ°	Nusselt Number		Friction Factor f
		$T_s = \text{Const.}$	$\dot{q}_s = \text{Const.}$	
Circle 	—	3.66	4.36	64.00/Re
Rectangle 	$\frac{a}{b}$ 1 2 3 4 6 8 ∞	2.98 3.39 3.96 4.44 5.14 5.60 7.54	3.61 4.12 4.79 5.33 6.05 6.49 8.24	56.92/Re 62.20/Re 68.36/Re 72.92/Re 78.80/Re 82.32/Re 96.00/Re
Ellipse 	$\frac{a}{b}$ 1 2 4 8 16	3.66 3.74 3.79 3.72 3.65	4.36 4.56 4.88 5.09 5.18	64.00/Re 67.28/Re 72.96/Re 76.60/Re 78.16/Re
Triangle 	θ 10° 30° 60° 90° 120°	1.61 2.26 2.47 2.34 2.00	2.45 2.91 3.11 2.98 2.68	50.80/Re 52.28/Re 53.32/Re 52.60/Re 50.96/Re

Diferentes valores según la sección [Cengel] 1.3

- Región de entrada térmica. N° de Nusselt para grandes diferencias de temperaturas (Sieder y Tate):

$$Nu = 1.86 \left(\frac{D}{L} Re Pr \right)^{1/3} \left(\frac{\mu_b}{\mu_s} \right)^{0.14} \quad (1.46)$$

Propiedades a T_b , excepto μ_s a T_s

Flujo turbulento desarrollado en tubos suaves.

- Relación de Colburn:

$$f = 0.184 Re^{-0.2} \quad Nu = 0.023 \cdot Re^{0.8} \cdot Pr^{1/3} \quad \left(\begin{array}{l} 0.7 \leq Pr \leq 160 \\ Re > 10000 \end{array} \right) \quad (1.47)$$

Propiedades determinadas a $T_m = 0.5(T_{ent} + T_{sal})$

- Relación de Dittus-Boelter:

$$Nu = 0.023 \cdot Re^{0.8} \cdot Pr^n \quad \left(\begin{array}{l} 0.7 \leq Pr \leq 160 \\ Re > 10000 \end{array} \right) \quad (1.48)$$

$n=0.4$ cuando se calienta, $n=0.3$ cuando se enfría. Propiedades se determinadas a $T_m = 0.5(T_{ent} + T_{sal})$

Flujo turbulento desarrollado en tubos suaves cuando la diferencia de temperatura entre el fluido y la superficie es grande.

- Relación de Sieder y Tate

$$Nu = 0.027 Re_D^{4/5} \cdot Pr^{1/3} \left(\frac{\mu}{\mu_s} \right)^{0.14} \quad (0.7 = Pr = 16700; \quad Re > 10000; \quad L/D \geq 10) \quad (1.49)$$

Las propiedades del fluido están evaluadas a $T_m = 0.5(T_{ent} + T_{sal})$, excepto μ_s que se evalúa a la temperatura de la superficie.

Flujo turbulento desarrollado en tubos suaves.

- Relación de Petukov.

$$Nu = \frac{(f/8) Re Pr}{1.07 + 12.7(f/8)^{0.5} (Pr^{2/3} - 1)} \quad \left(\begin{array}{l} 0.5 \leq Pr \leq 2000 \\ 10^4 < Re < 5 \cdot 10^6 \end{array} \right) \quad (1.50)$$

Propiedades se determinadas a $T_m = 0.5(T_{ent} + T_{sal})$

- Relación de Gnielinski.

$$Nu = \frac{(f/8)(Re - 1000) Pr}{1 + 12.7(f/8)^{0.5} (Pr^{2/3} - 1)} \quad \left(\begin{array}{l} 0.5 \leq Pr \leq 2000 \\ 3 \cdot 10^3 < Re < 5 \cdot 10^6 \end{array} \right) \quad (1.51)$$

Propiedades se determinadas a $T_m = 0.5(T_{ent} + T_{sal})$

- Metales líquidos

$$\left. \begin{aligned} \text{Nu} &= 4.8 + 0.0156 \text{Re}^{0.85} \text{Pr}_s^{0.93}; T_s = \text{cte} \\ \text{Nu} &= 6.3 + 0.0167 \text{Re}^{0.85} \text{Pr}_s^{0.93}; \dot{q}_s = \text{cte} \end{aligned} \right\} \begin{aligned} &(0.004 < \text{Pr} < 0.01) \\ &(10^4 < \text{Re} < 10^6) \end{aligned} \quad (1.52)$$

- Factor de fricción. Ec. De Colebrook para flujo turbulento

$$\frac{1}{\sqrt{f}} = -2 \ln \left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{\text{Re} \sqrt{f}} \right) \quad (1.53)$$

Flujo turbulento en conductos no circulares.

- Tubos anulares

D_i/D_o	Nu_i	Nu_o
0	—	3.66
0.05	17.46	4.06
0.10	11.56	4.11
0.25	7.37	4.23
0.50	5.74	4.43
1.00	4.86	4.86

Valores según la relación de D 1.4

$$\text{Nu} = 7.54 + \frac{0.03(D_h/L) \text{Re} \text{Pr}}{1 + 0.016[(D_h/L) \text{Re} \text{Pr}]^{2/3}}; \quad \text{Re} \leq 2800 \quad (1.54)$$

D_h : diámetro hidráulico

Flujo turbulento en conductos

- Factor fricción superficies suaves

$$f = (0.79 \ln(\text{Re}) - 1.64)^{-2} \quad 10^4 < \text{Re} < 10^6 \quad (1.55)$$

- N° de Nusselt

$$\text{Nu} = 0.125 f \cdot \text{Re} \cdot \text{Pr}^{1/3} \quad (1.56)$$

1.2- Objetivos Trabajo Fin de Grado.

En este TFG se pretende diseñar una nueva instalación experimental para la caracterización del proceso de transferencia de calor de aire en tubos y simular su comportamiento numéricamente a través de software de Mecánica de Fluidos Computacional, con el fin de apoyar el diseño en los resultados numéricos. Para el diseño se seleccionarán ventiladores, termopares, sensores de presión diferencial, resistencias para calentamiento de conductos entre otros elementos. Para la simulación numérica del comportamiento de la instalación se utilizará el software OpenFoam.

Los objetivos específicos de este TFG son los siguientes:

- Familiarizarse con el trabajo de laboratorio de diseño y montaje de una instalación experimental.
- Conocer y aprender a utilizar distinta instrumentación de medida en ingeniería térmica.
- Aprender a utilizar herramientas de software libre de mecánica de fluidos computacional

para simular problemas de transmisión de calor por convección.

- Investigar el proceso de transferencia de calor por convección entre un gas y una superficie sólida.

1.3- Fases del trabajo.

Las fases afrontadas en el desarrollo de este TFG han sido las siguientes:

1. Revisión del prediseño ya realizado.
2. Selección de material y diseño de distintos útiles de conexión para fabricación en el taller de la UPCT.
3. Familiarización con el software OpenFoam. Realización de tutoriales.
4. Desarrollo de una malla para la simulación numérica del problema.
5. Selección y ajuste de un modelo numérico que incluya el modelado de la turbulencia para flujo de aire en convección forzada en un tubo.
6. Implementación, en OpenFoam, de las condiciones de contorno de flujo de calor constante en la pared.
7. Simulación del problema y obtención, para distintas condiciones de operación, de temperaturas medias del aire de salida y coeficientes convectivos para distintos caudales y/o velocidades del aire.
8. Elaboración de la memoria.

CAPÍTULO 2. INSTALACIÓN EXPERIMENTAL.

2.1- Definición de la instalación experimental y descripción de la instrumentación.

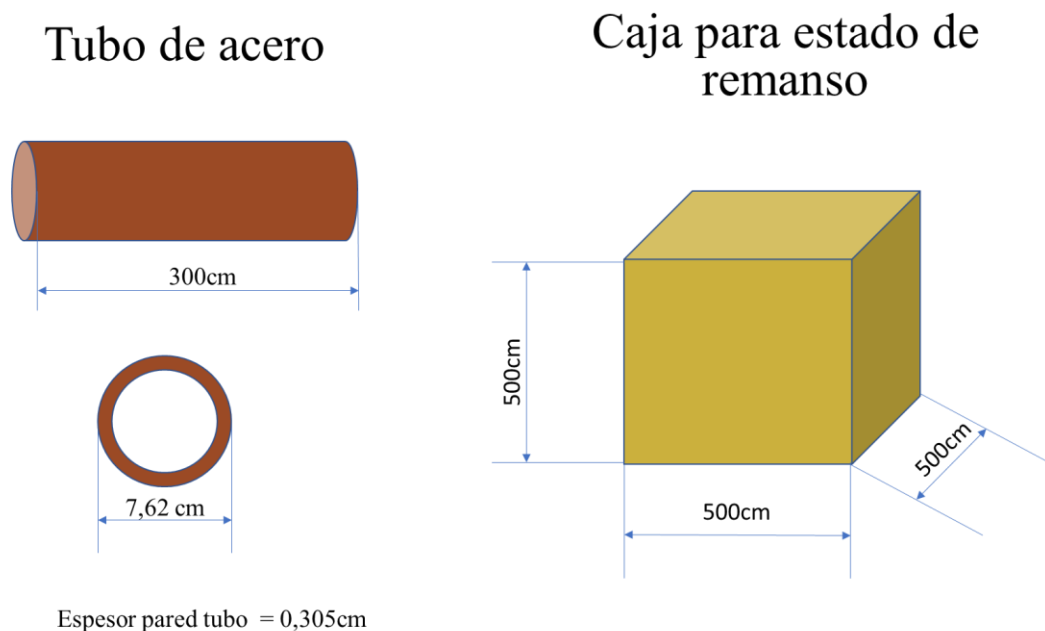
En primer lugar, haremos una descripción gráfica del equipo completo, así como las dimensiones de éste y las posiciones de los instrumentos de medida en el mismo.

Debemos tener en cuenta que, del equipo que se presenta a continuación, en la práctica nos encontraremos con dos réplicas del mismo, y que los instrumentos usados serán los mismos en ambos.

Los elementos principales de nuestro equipo se basan en un ventilador, un tubo de acero y una caja de remanso que utilizaremos para amortiguar las posibles fluctuaciones del ventilador y para medir el estado de remanso de presión y temperatura. A esta caja acoplaremos el ventilador. La función principal del ventilador será impulsar el fluido desde el estado de remanso hasta el exterior atravesando el tubo, de esta manera tendremos un flujo continuo en el que poder medir las diferentes variables necesarias en nuestro estudio.

Proporcionaremos calor al tubo de acero mediante la cara exterior de este con una manta eléctrica. Esta cara del tubo y la manta irán recubiertas por una capa de coquilla que actuará como aislante exterior, y supondremos que no existirán pérdidas por calor.

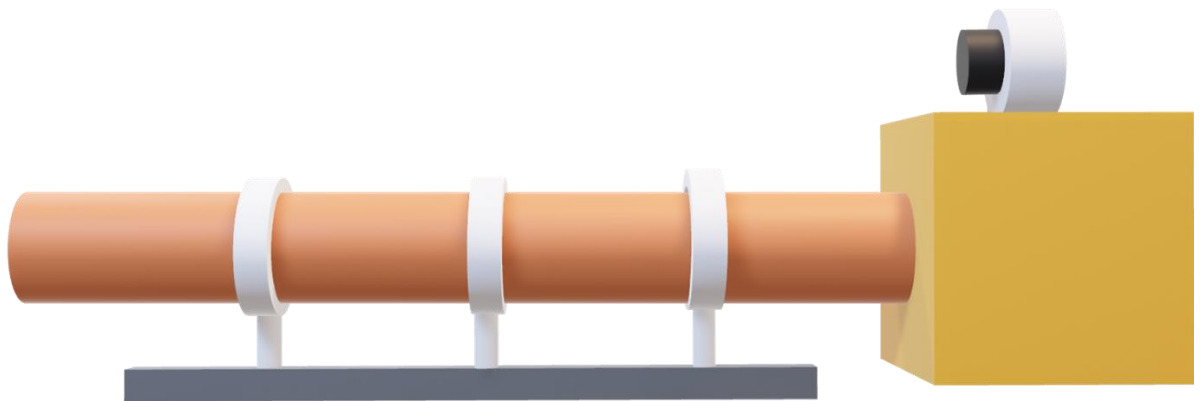
Las dimensiones y disposición del equipo serán:



Medidas tubo y caja 2.1

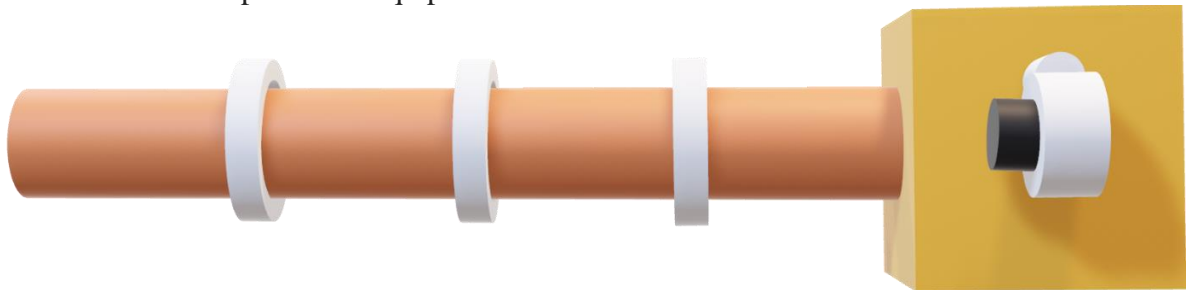
Ambos, junto al ventilador, una guía de posicionamiento y sujeciones para el tubo, formarán un conjunto, que estará dispuesto de la siguiente manera:

- Vista desde el alzado del equipo.



Alzado simulación del equipo 2.2

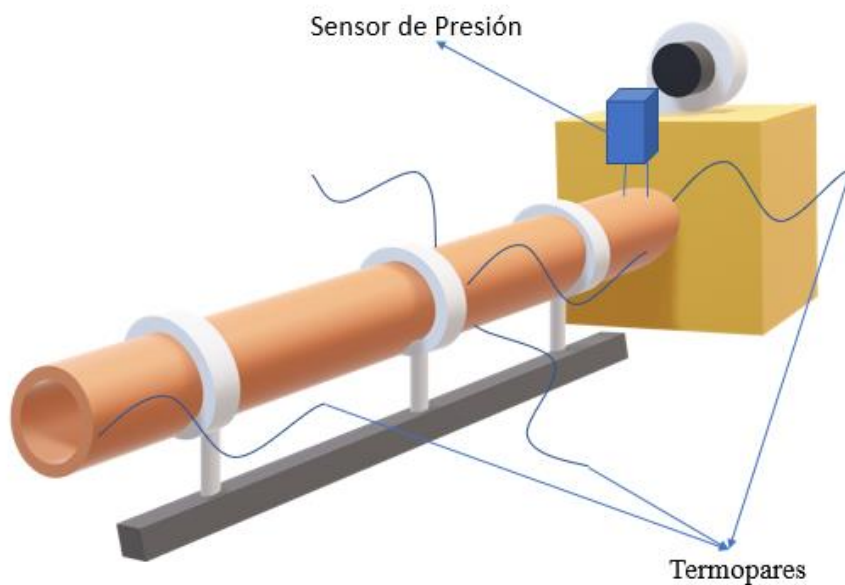
- Vista desde la planta del equipo.



Alzado simulación del equipo 2.3

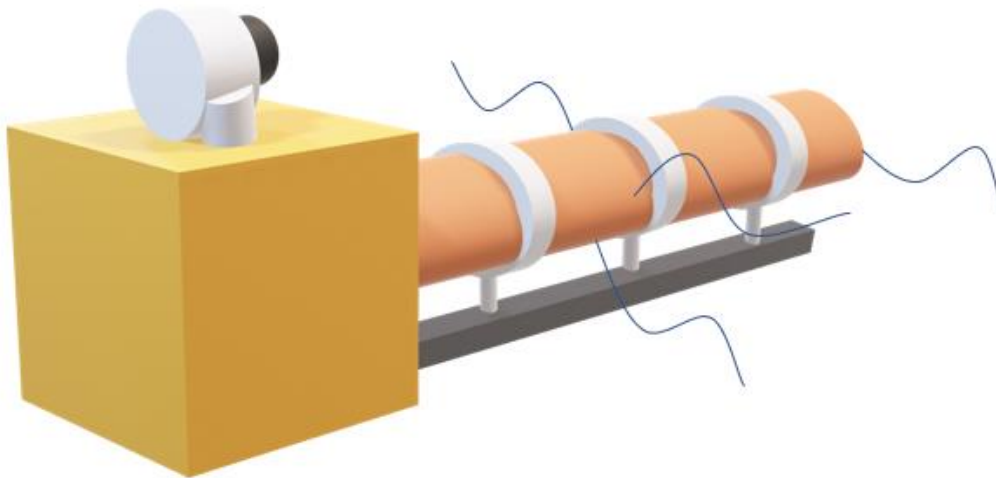
Este equipo vendrá complementado con los correspondientes instrumentos de medida que se implementarán desde una vista general de la siguiente forma:

- Vista delantera.



Vista delantera simulación 2.4

➤ Vista trasera.



Vista trasera simulación 2.5

Las especificaciones técnicas tanto de la guía de posicionamiento como del ventilador se pueden encontrar en los anexos.

La disposición de los mencionados equipos de medida la concretaremos más adelante, junto a la explicación detallada de cada uno de ellos, así como su funcionamiento y utilidad en este estudio.

2.2- Variables fluidodinámicas a medir.

Partiendo de que nuestro principal objetivo en este estudio será calcular el coeficiente de transmisión de calor h , tendremos que medir las siguientes propiedades para realizar este cálculo.

Partimos de las ecuaciones que ya hemos presentado en la introducción para esquematizar este apartado.

$$q'' \cdot A = \dot{m} \cdot c_p \cdot (T_{fe} - T_{fs}) \quad (2.1)$$

$$h = \frac{q}{A \cdot (T_s - T_m)} \quad (2.2)$$

Calcularemos de dos diferentes formas el calor transmitido por unidad de tiempo. Como ya hemos dicho anteriormente, la primera será mediante la potencia de la manta térmica y la segunda siguiendo la primera ecuación 2.1.

Este procedimiento redundante lo haremos para comprobar los cálculos que realizamos.

2.2.1- Potencia calorífica suministrada por la manta térmica.

Con un multímetro mediremos intensidad y voltaje que suministra el autotransformador a la manta eléctrica, por lo que podremos calcular esta potencia calorífica.

2.2.2- Área de mojado y área transversal.

El área mojada se corresponde con la superficie del tubo en contacto con el aire que es $A_m = \pi LD$

El área transversal se corresponde al área de paso del fluido, como en nuestro caso se trata de aire, el área será el de la circunferencia interna del tubo del equipo. Por tanto, tendremos:

$$A = \frac{\pi \cdot d^2}{4} = \frac{\pi \cdot 7.315^2}{4} = 42.03 \text{ cm}^2 = 0.42 \text{ m}^2 \quad (2.3)$$

2.2.3- Temperatura superficial interior y exterior del tubo.

Con los termopares de contacto colocados en la superficie exterior del tubo podremos obtener directamente esta medida de temperatura de la superficie del tubo, T_{se} , que será constante por toda la superficie.

La temperatura de la superficie interior del tubo, T_{si} , la estimaremos a partir de la medida de la temperatura exterior del tubo, ya que no queremos realizar perforaciones en el mismo.

A partir de la temperatura exterior, estimaremos la temperatura interior del tubo asumiendo conducción unidimensional estacionaria a lo largo de la superficie del tubo.

Nuestro interés se centrará en la temperatura del aire que circula por el interior, por tanto, teniendo en cuenta las especificaciones anteriores, necesitaremos hacer una serie de operaciones con nuestros datos para poder hallar la temperatura interior.

Este proceso lo realizaremos partiendo de la Ley de Fourier, particularizada para el caso de la geometría descrita anteriormente, que podemos expresar como:

$$q = \frac{T_1 - T_2}{\frac{\ln(r_2/r_1)}{2\pi L k_{12}}} \quad (2.4)$$

Siendo T_1 y T_2 , las temperaturas en la pared interior T_{si} y la exterior respectivamente T_{se} ; r_1 y r_2 , el radio de la pared interna y la externa respectivamente; L la longitud del tubo de acero y por último k_{12} la conductividad térmica del material.

Podemos decir entonces que $T_{si} = T_1$ es la temperatura que nosotros estaremos midiendo en el interior del tubo y que $T_2 = T_{se}$.

Despejando la ecuación anterior nos queda:

$$T_1 = q \cdot \frac{\ln(r_2/r_1)}{2\pi L k_{12}} + T_2 \quad (2.5)$$

Podemos tomar el valor de q dado por la manta térmica.

Para comprobar si existen pérdidas de calor, aún con el aislante, podemos evaluar la transferencia de calor hacia el exterior.

2.2.4- Temperatura del fluido a la entrada.

La mediremos con un termopar ubicado en la caja de remanso.

2.2.5- Temperatura del fluido a la salida.

La diferencia de este caso con el anterior, a parte de la posición de toma de medidas en el tubo, ya que esta se medirá al final de este, es que ahora podemos medir directamente la temperatura del fluido colocando los termopares en el interior del tubo. Por esta parte nos encontramos con la posibilidad de poder realizarlo ya que esta parte está abierta al exterior.

2.2.6- Temperatura media del fluido.

Para esta medida utilizaremos las dos anteriores ya que, como hemos mencionado, en la introducción esta medida es una operación con dichas medidas. Recordamos el procedimiento de este cálculo usando la siguiente ecuación:

$$T_m = \frac{T_{fe} + T_{fs}}{2} \quad (2.6)$$

2.2.7- Caudal másico.

Primero definiremos este parámetro como:

$$\dot{m} = \rho \cdot v \cdot S \quad (2.7)$$

Entendiendo ρ como la densidad del aire, que la tenemos por las propiedades ya especificadas del aire y tomaremos como constante, v como la velocidad del fluido y S como la sección de paso, que tendrá el mismo valor que el área de mojado ya calculado anteriormente.

Lo que necesitamos entonces para poder calcular el caudal másico será la velocidad del aire, ésta la mediremos mediante un tubo de Pitot y un sensor de presión diferencial, este nos proporciona el valor de la diferencia de presiones entre dos puntos. Uno nos medirá la presión total y el otro la estática. Procedemos entonces de la siguiente manera.

Mediante la ecuación de Bernoulli:

$$p_1 + \frac{1}{2} \cdot \rho \cdot v_1^2 = p_2 + \frac{1}{2} \cdot \rho \cdot v_2^2 \quad (2.8)$$

Como podemos tomar $\rho = cte$, y sabiendo que por la ecuación de continuidad se tiene que cumplir que:

$$\rho \cdot v_1 = \rho \cdot v_2 \quad (2.9)$$

Por consiguiente, sabemos que $v_1 = v_2$.

Reajustamos la primera ecuación mostrada,

$$\Delta p = p_t - p_e = \frac{1}{2} \cdot \rho \cdot v^2 \quad (2.10)$$

Definimos el punto 1 como el que nos proporcionará la presión total y el 2 como el que lo hará con la estática.

Calcularemos la velocidad con el valor de diferencia de presión como:

$$v = \sqrt{\frac{2 \cdot \Delta p}{\rho}} \quad (2.11)$$

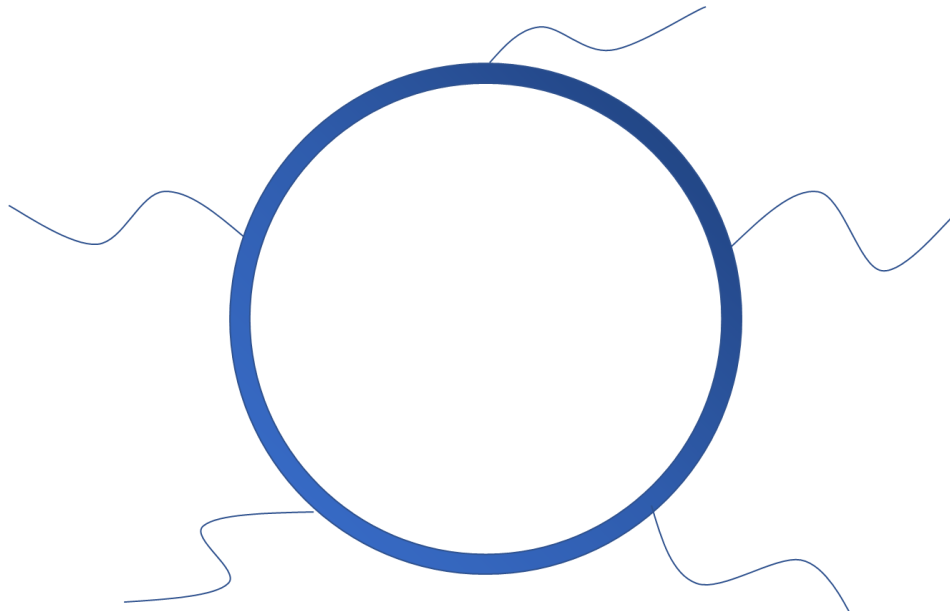
2.2.8- Calor específico.

Este valor pertenece a una propiedad del aire como fluido que obtendremos de los valores de tablas bibliográficas especificados para dicho fluido a la temperatura media del fluido, ya calculada previamente.

2.3- Descripción de equipos auxiliares para suministro de potencia y medida.

2.3.1- Termopares.

Para los termopares de contacto dispondremos de cinco para cada instalación, que utilizaremos para medir la temperatura de la superficie del tubo como ya hemos comentado anteriormente. Estos tendrán una disposición alrededor del tubo de forma equidistante:



Disposición termopares 2.6

Con los termopares de aire mediremos la temperatura del aire tanto en la caja de remanso, como a la entrada y a la salida del tubo.

Estas medidas serán tomadas por los correspondientes registradores para termopares que poseemos.

2.3.2- Manta eléctrica.

Este será el elemento que nos proporcionará el calor a lo largo de todo el tubo. Estará alimentado por el autotransformador del que disponemos.

El objetivo será que el tubo tenga un calor constante por toda su superficie.

2.3.3- Multímetro.

La utilidad del multímetro será para medir la potencia calorífica aportada por el autotransformador, como ya hemos comentado anteriormente.

2.3.4- Sensor de presión diferencial.

El sensor lo situaremos al principio del tubo, donde está conexionado con la caja de remanso y nos proporcionará la diferencia de presión en esa zona.

2.3.5- Aislante exterior.

Dicho aislante se tratará de “coquilla”, cuya función será disminuir la transferencia de calor hacia el exterior del tubo.

CAPÍTULO 3. MONTAJE Y PUESTA A PUNTO DE LAS INSTALACIONES.

La instalación experimental montada presenta el siguiente aspecto:



Vista frontal equipo 3.1



Vista trasera equipo 3.2

CAPÍTULO 4. SIMULACIÓN NUMÉRICA DEL PROBLEMA CON OPENFOAM.

4.1- Incidencia sobre el cambio de estrategia en la resolución del caso.

Debido al impedimento de poder realizar las tareas correspondientes de este trabajo en las instalaciones de la UPCT, debido al cierre de la misma ante la situación extraordinaria generada por el COVID-19, nuestro caso se ha resuelto mediante el programa libre de simulación numérica “OpenFoam”. En concreto, utilizaremos las librerías que tiene este software para Mecánica de Fluidos Computacional, basados en volúmenes finitos.

4.2- Objetivos a cumplir con OpenFoam.

Los objetivos a cumplir en el TFG con OpenFoam son:

- Generar una malla de 300.000 elementos, aproximadamente (debido a las limitaciones de capacidad de cálculo), que simule el volumen de aire en el interior del tubo de acero inoxidable que teníamos en el laboratorio.
- A partir de la malla, resolver las ecuaciones de conservación de masa energía y cantidad de movimiento al fluido que atraviesa el tubo imponiendo unas condiciones de contorno específicas para nuestro problema que nos permita simular la transmisión de calor y el cambio a lo largo del tubo de diferentes propiedades termodinámicas.
- Comparar los resultados de las simulaciones con la solución teórica del problema presentada en el capítulo 1.

4.3- Ecuaciones del modelo físico resuelto con Open Foam

4.3.1- Ecuaciones RANS: Reynolds Averaged Navier-Stokes y el modelo k-epsilon.

Como hemos explicado en la introducción de este trabajo, OpenFoam trabaja con las llamadas ecuaciones RANS que son un promedio de las ecuaciones de Navier-Stokes, cuando éstas últimas se escriben para variables fluidas que contienen una componente media y una componente pulsátil. Ya vimos la necesidad de definir una viscosidad turbulenta con el modelo “Eddy viscosity”. Una vez entendida la problemática, presentamos el modelo k-epsilon que utiliza este programa para resolver dicha viscosidad, lo hace utilizando dos ecuaciones para definir la misma.

Estas dos ecuaciones son ecuaciones de transporte que definen la energía cinética de la turbulencia, k , y la disipación turbulenta, ε .

Para k :

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho k u_i)}{\partial x_i} = \frac{\partial}{\partial x_j} \left[\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j} \right] + 2\mu_t E_{ij} E_{ij} - \rho \varepsilon \quad (4.1)$$

Para ε :

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \frac{\partial(\rho \varepsilon u_i)}{\partial x_i} = \frac{\partial}{\partial x_j} \left[\frac{\mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_j} \right] + C_{1\varepsilon} \frac{\varepsilon}{k} 2\mu_t E_{ij} E_{ij} - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \quad (4.2)$$

Donde u representa la velocidad en la dirección correspondiente, E_{ij} representa el componente de velocidad de deformación y μ_t la viscosidad Eddy, que podemos escribir como:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (4.3)$$

Las ecuaciones también cuentan con constantes propias del modelo.

Este sistema de ecuaciones se completa con las condiciones iniciales y de contorno siguientes:

Para las iniciales impondremos una velocidad en el conducto y una temperatura de 298 K, para las de contorno contamos con un gradiente de temperaturas que explicaremos más a fondo en la aplicación de nuestro caso con el programa.

4.3.2- Resolución por el método de volúmenes finitos.

En esta sección se describe brevemente el método numérico de volúmenes finitos que aplicará OpenFoam para resolver las ecuaciones matemáticas presentadas previamente. Básicamente, el método numérico sigue los siguientes pasos [6.6 en la bibliografía]:

1. Descomponer el dominio en pequeñas celdas o volúmenes de control.
2. Formular las ecuaciones integrales de conservación para cada volumen de control.
3. Aproximar numéricamente las integrales.
4. Aproximar los valores de las variables en las caras y las derivadas espaciales con la información de las variables nodales.
5. Aproximar las derivadas temporales por incrementos finitos.
6. Ensamblar y resolver el sistema algebraico obtenido.

El método de los volúmenes de control finitos permite discretizar y resolver numéricamente ecuaciones diferenciales. Es un método alternativo a los de diferencias finitas y elementos finitos.

Consideremos una malla de discretización del espacio fluido. En torno a cada punto de esta malla se construye un volumen de control que no se traslapa con los de los puntos vecinos. De esta forma el volumen total de fluido 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, de modo que permita 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. [6.6 y 6.15]

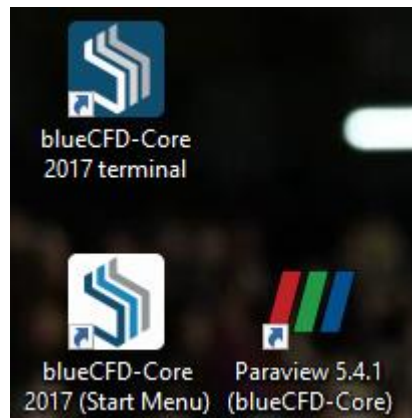
4.4- Introducción para el uso de OpenFoam.

Haremos un breve resumen del manejo del programa, de los pasos a seguir y de los comandos y acciones más importantes en el uso de este simulador.

4.4.1- Descarga del programa y accesos directos.

Primero hay que mencionar que el programa es descargable en la página oficial [6.14 en la bibliografía] de forma gratuita y que nos proporcionará la versión más reciente del mismo. Utilizaremos el paquete “blueCFD-2017”.

Una vez descargado podremos apreciar tres accesos directos en nuestro escritorio:



Ítems programa 4.1

El llamado “terminal” es el que abriremos para escribir todos los comandos que queramos ordenar al programa, “Start Menu” nos dirigirá a todos los archivos que se descargan junto al programa, entre ellos estarán el manual en formato pdf del programa y todos los archivos de simulación correspondiente a cada tutorial disponible. Por último, el acceso “Paraview” se abrirá directamente cuando simulemos una malla con o sin condiciones de contorno que nos permitirá comprobar el caso y sus variantes.

4.4.2- Pasos previos antes de simular un caso.

Antes de poder simular cualquier caso que deseemos debemos hacer varios pasos previos en el programa:

1. Crear un directorio llamado “run” donde acudiremos y guardaremos todos los archivos de cada simulación que hagamos, para el correcto funcionamiento del programa y que nos servirá para encontrar los archivos de una forma más simple. Una vez abierto el terminal del programa introduciremos el siguiente comando:
 - `mkdir -p $FOAM_RUN`
2. Podemos encontrar la ubicación (el programa nos llevará directamente) de dicho archivo introduciendo:
 - `echo $FOAM_RUN`
3. Ya creado este fichero, como ya hemos mencionado, llevaremos ahí todos los archivos que deseemos simular.

4.4.3- Comandos más comunes y su utilidad.

Podemos hacer una breve guía como la siguiente para familiarizarnos más rápidamente con el terminal de escritura y entender para qué utilizamos cada comando (después de escribir cada comando deberemos presionar la tecla “intro” para que se pueda ejecutar). El propio sitio web donde descargamos el programa nos ofrece un videotutorial y el manual del programa, pero como lo más destacado y lo que más utilizaremos podemos resaltar:

1. Para entrar en el directorio “run”, una vez ya creado, solo hace falta escribir el siguiente comando:
 - run
2. Para poder visualizar el contenido de cada carpeta en la que estemos:
 - ls
3. Si queremos entrar a una carpeta o una dirección que esté dentro de la carpeta en la que estemos situados:
 - cd “nombre de la carpeta en la que deseemos entrar”
4. También podemos ir a una dirección específica de nuestro archivo si ya conocemos el nombre de dichas carpetas:
 - cd “nombre/nombre/nombre”

En el terminal podremos ver en que carpeta estamos actualmente, y de donde está situada esa carpeta, poniendo los nombres de cada carpeta de forma continuada:

```
LENOVO@W8 MINGW64 OpenFOAM-5.x ~/blueCFD/ofuser-of5/run/cilindro
```

Dirección de carpetas 4.2

En este caso, para poner un ejemplo explicativo, la ruta habrá sido entrar en los archivos de blueCFD. A continuación, elegiremos la carpeta “run”, y dentro de esta, la carpeta “cilindro”.

5. Para retroceder utilizaremos:
 - cd ..
6. Podemos crear copias de archivos cambiando el nombre (más adelante veremos un ejemplo más claro de este comando):
 - cp -r
7. Si queremos abrir directamente algún archivo y poder modificarlo introduciremos:
 - nano “dirección del archivo”

Nos abrirá el archivo en un block de notas donde podremos cambiarlo.

8. Una vez retocados todos los archivos ajustándose a las condiciones de cada caso, deberemos mandar al programa que cree la malla correspondiente:
 - blockMesh
9. Cuando tengamos la malla creada, podremos abrir el caso en “Paraview”:

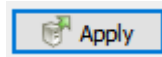
- paraFoam &

4.4.4- Paraview y sus aplicaciones más usadas.

Para poder manejarnos con más facilidad mostraremos de nuevo unos consejos de las opciones más usadas en este caso del ítem “Paraview”.

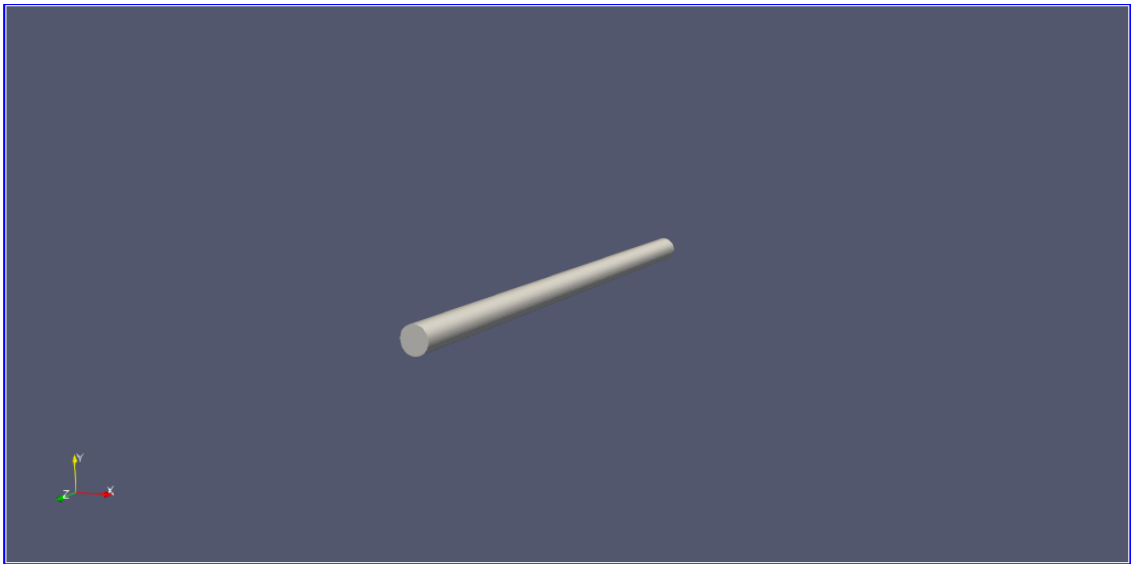
Una vez creada la malla y abierto el caso en Paraview, nos aparecerá una nueva ventana donde trabajar.

En la parte izquierda nos aparecerá nuestro caso, y deberemos hacer click en la pestaña “apply” para que simule nuestro caso.



Botón apply 4.3

A continuación, podremos ver la malla creada en la parte derecha de la pantalla, representada en 3D. Dentro de esta ventana podremos manejar la malla como deseemos.



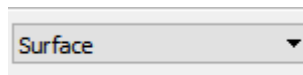
Pantalla de visualización 4.4

En la parte superior a esta ventana podemos observar varias opciones que nos colocarán la malla en diferentes posiciones con relación a los ejes.



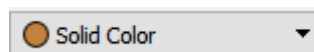
Botones para el manejo de ejes 4.5

El programa también nos da la opción de ver la malla con diferentes apariencias, esto lo podremos ver en la siguiente pestaña desplegable:



Pestaña para aspecto malla 4.6

Si queremos cambiar la propiedad que queremos ver, por ejemplo, de estar observando la presión en la malla a la velocidad, lo cambiaremos en:



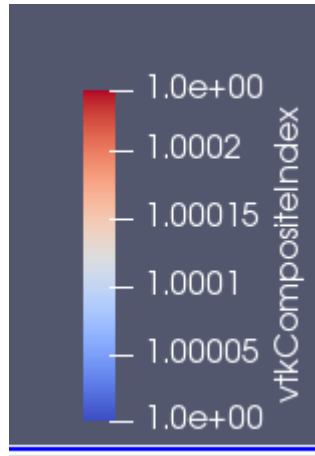
Pestaña para cambiar propiedades 4.7

Otra de las opciones más destacadas para poder estudiar los casos es la posibilidad de hacer un corte a la malla en la dirección que deseemos. En el programa se corresponde con el nombre “Slice”, que nos aparecerá al poner el cursor encima del símbolo:



Botón slice 4.8

Al estudiar las distintas propiedades que introduzcamos en la malla, en la misma ventana donde podemos ver la malla aparecerá una leyenda con el rango de valores y los colores con la que está representada dicha propiedad dentro de la malla:



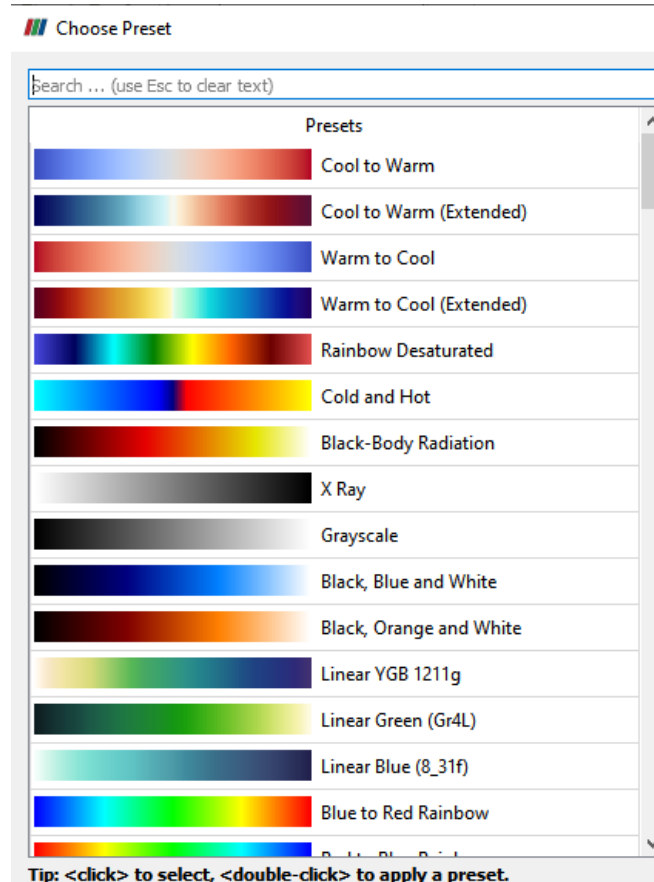
Leyenda propiedades 4.9

Podemos cambiar el rango de colores en la ventana izquierda inferior, en las propiedades de la malla en la siguiente opción:



Botón para cambiar colores 4.10

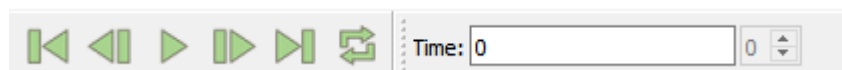
Que nos desplegará la siguiente pantalla con distintas opciones:



Opciones de colores 4.11

Normalmente elegiremos la opción “Blue to Red Rainbow”.

Y, por último, para poder simular los distintos intervalos de tiempo en la malla y observar cómo afectan las condiciones introducidas, lo haremos en:



Manejo del tiempo 4.12

4.4.5- Tutorial para entender OpenFoam.

Antes de resolver el caso o ver cómo funciona cualquier “solver” del programa, podemos realizar un tutorial guiado para entender mejor cómo tratar con el simulador. En nuestro caso hemos escogido un tutorial realizado por el profesor Ramón A. Otón Martínez. Es un tutorial estudiando distintas propiedades sobre un alabe, y usando dos “solver” distintos. Aunque este tutorial tenga más complejidad, nosotros lo usaremos para entender distintas variedades del programa y poder llevarlo luego a nuestro caso.

Los ficheros del caso que usaremos nos han sido facilitados, así no tendremos que crear una malla o los archivos de cada condición de contorno, simplemente manejaremos dichos archivos y podremos modificarlos. La malla de este caso proviene del tutorial ‘airfoil’ para flujo incompresible, que se encuentra en la distribución original de OpenFoam.

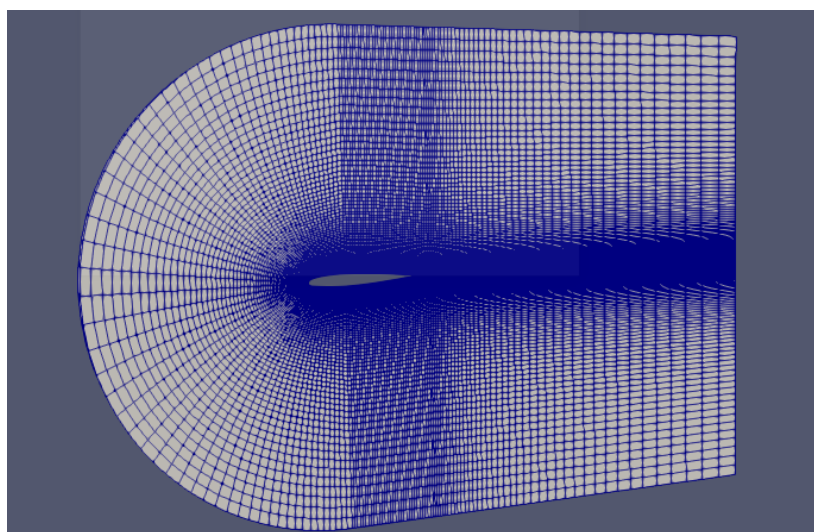
En primer lugar, haremos una visualización del caso con el “solver”: “simpleFoam”, que el propio programa nos facilita, dentro de un catálogo amplio de dichos “solver”. En este caso éste es usado para fluidos incompresibles en estado estacionario con un modelo turbulento. No entraremos a explicarlo profundamente en este apartado, pero sí más adelante cuando tengamos que imponer las condiciones de contorno de nuestra malla.

Los pasos por seguir serán:

1. Abriremos “blueCFD – core 2017 terminal”.
2. Como ya hemos explicado, crearemos la dirección “run”:
 - `mkdir -p $FOAM_RUN`
 - `echo $FOAM_RUN`
3. Copiaremos “scripts_practica” en esta dirección, se trata de los archivos facilitados para poder realizar este caso.
4. Entraremos en esta carpeta desde el terminal, y posteriormente en la carpeta “caso_base” que se encuentra en “incompresSimple”:
 - `run`
 - `cd scripts_practica/incompresSimple/caso_base`
 - `ls`
5. Creamos la malla:
 - `blockMesh`
6. Para ver la malla:
 - `paraFoam &`
 - Click



Veremos:



Visualización malla 4.13

La superficie blanca dividida en elementos por las líneas de color azul representa la malla que hemos creado- Es importante resaltar que la malla que representamos en este caso no es el álabe propiamente dicho. Se trata del espacio por donde circula el fluido y en el que queremos estudiar las propiedades y cómo se comporta ante ellas. En este caso concreto, el álabe es la parte central de lo que podemos observar en la imagen.

7. Establecemos las condiciones de contorno:

- nano 0/U &

Se nos abrirá el siguiente archivo:

```

13   object      U;
14 }
15 // * * * * *
16
17 inlet        (200 0 0);
18
19 dimensions   [0 1 -1 0 0 0];
20
21 internalField uniform $inlet;
22
23 boundaryField
24 {
25     inlet
26     {
27         type          freestream;
28         freestreamValue uniform $inlet;
29     }
30
31     outlet
32     {
33         type          freestream;
34         freestreamValue uniform $inlet;
35     }
36
37     aerofoil
38     {
39         type          noSlip;
40     }
41
42     // frontAndBack
43     // {
44     //     type          freestream;
45     // }
46 }
47

```

Archivo U 4.14

- Debemos cambiar el valor de la velocidad, en este caso, donde pone Uinlet (XXXX 0 XXXX), cambiamos a 200 en la dirección x: Uinlet (200 0 0).
- En cada archivo que abramos de esta forma nos aparecerán los límites o contornos de la malla, tendremos que especificar las condiciones en cada uno de ellos:

- inlet: freestream; (Se refiere a la parte de “entrada”)
 - outlet: freestream; (Para la “salida”)
 - aerofoil: noSlip; (Condición de adherencia en la pared del álabe)
 - frontAndBack: empty; (La parte frontal y la trasera)
8. En este caso dos parámetros a medir son dos coeficientes aerodinámicos (C_d y C_l) dependientes de la velocidad y de las dimensiones características del caso. Deberemos cambiar estas características también:

- nano system/forceCoeffs &
- magUInf XXXX; (Velocidad)
- rhoInf 1.25; (Densidad)
- lRef XXXX; (Longitud)
- Aref XXXX; // area = c x b

Pondremos los svalores:

```

44 magUInf 200;
45 rhoInf 1.25;
46 lRef 1;
47 Aref 0.1;
    
```

Archivo forceCoeffs 4.15

9. Hacemos correr la simulación, mediante un script previamente preparado para este tutorial, que ejecuta internamente los comandos de OpenFoam necesarios:

- ./Allrun_ROM

10. Una vez hecho esto, podemos visualizarlo:

- paraFoam &

Este tutorial también es útil desde el punto de vista de que nos enseña a cómo poder modificar la malla, en este caso lo que haremos será añadir elementos, esto nos permitirá hacer simulaciones más precisas y parecidas a la realidad.

1. Copiaremos el caso base y limpiaremos:

- cd ..
- cp -r caso_base caso_refined
- cd caso_refined
- ./Allclean_ROM

2. Modificamos la malla en “blockMeshDict”:

- nano system/blockMeshDict &
- Escribimos:
- // Number of cells
zCells 320;
xUCells 120;

```
xMCells 120;
```

```
xDCells 160;
```

```
// Mesh grading
```

```
zGrading 60;
```

```
xUGrading 5;
```

```
leadGrading 0.2;
```

```
xDGrading 10;
```

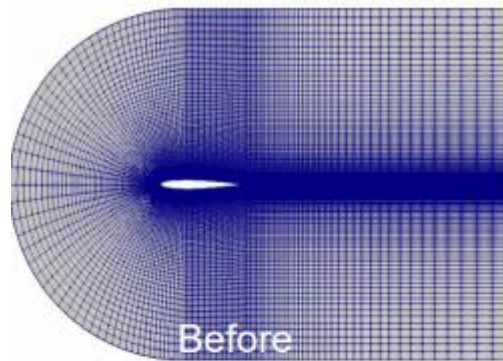
Esto significará un aumento de los elementos de la malla.

3. Cargamos la nueva malla en el programa

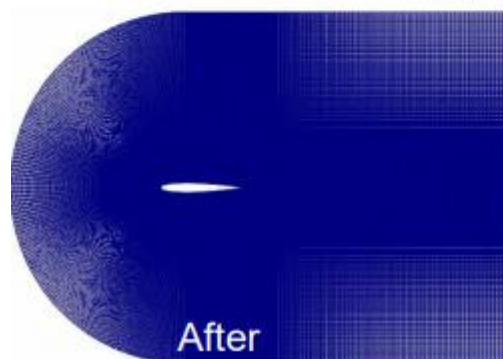
- blockMesh

4. Vemos el resultado en paraView

- paraFoam &



1:



Visualización aumento elementos 4.16

El mismo caso lo podemos realizar utilizando un “solver” distinto. En el tutorial utiliza “rhoSimpleFoam” que se usa para fluidos en régimen turbulento, pero en ese caso compresibles.

Para poder realizarlo deberemos hacer los mismos pasos de antes:

1. Ya abierto el terminal y creado el directorio de “run”, en el caso que no lo hayamos creado antes, vamos a la siguiente carpeta:
 - run
 - cd scripts_practica/rhoSimpleFoam/base _case
 - ls
 2. Creamos la nueva malla:
 - blockMesh
 3. Para poder verla como antes y comprobar que es la misma la abrimos en paraView:
 - paraFoam &
 4. Establecemos las condiciones de contorno:
 - nano 0/U &
- Se nos abrirá el siguiente archivo:

```

13   object      U;
14 }
15 // * * * * *
16
17   inlet       (200 0 0);
18
19   dimensions  [0 1 -1 0 0 0 0];
20
21   internalField  uniform $inlet;
22
23   boundaryField
24 {
25     inlet
26     {
27         type      freestream;
28         freestreamValue uniform $inlet;
29     }
30
31     outlet
32     {
33         type      freestream;
34         freestreamValue uniform $inlet;
35     }
36
37     aerofoil
38     {
39         type      noslip;
40     }
41
42     // frontAndBack
43     // {
44     //     type      freestream;
45     // }
46 }
47

```

Archivo U (2) 4.17

- Debemos cambiar el valor de la velocidad, en este caso cambiamos a 200 en la dirección x. Uinlet (XXXX 0 XXXX)
 - En cada archivo que abramos de esta forma nos aparecerán los límites o contornos de la malla, tendremos que especificar las condiciones en cada uno de ellos:
 - inlet: freestream; (Se refiere a la parte de “entrada”)
 - outlet: freestream; (Para la “salida”)
 - aerofoil: noSlip; (Condición de adherencia en la superficie del álabe)
 - frontAndBack: empty; (La parte frontal y la trasera)
5. Como en el caso anterior deberemos cambiar también:
- nano system/forceCoeffs &
 - magUInf XXXX; (Velocidad)
 - rhoInf 1.25; (Densidad)

- lRef XXXX; (Longitud)
- Aref XXXX; // area = c x b

Pondremos los svalores:

```
44 magUInf 200;
45 rhoInf 1.25;
46 lRef 1;
47 Aref 0.1;
```

Archivo forceCoeffs (2) 4.18

6. Hacemos correr la simulación, mediante un script similar al anterior:

- ./Allrun_ROM

7. Una vez hecho esto, podemos visualizarlo:

- paraFoam &

Una vez terminados los dos, podemos compararlos, tendrán diferencias en la simulación de las propiedades ya que como hemos comentado estamos usando dos “solver” diferentes. No entraremos en la complejidad de las diferencias de cada uno, la más importante es que tratamos con dos fluidos diferentes, uno se trata de un fluido incompresible y en el último caso de uno compresible. Esta diferencia se ve reflejada en las ecuaciones y el modelo que vienen integrados en cada “solver”.

Este tutorial nos es muy útil ya que vemos dos formas diferentes de resolver un caso para una misma malla, tenemos libertad ya que tenemos la capacidad de variar cada archivo. También hay que destacar la importancia de saber cambiar el fichero “blockMeshDict”, ya que lo llevaremos a la práctica con nuestro caso.

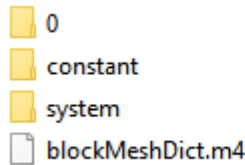
4.5- Generación de la malla.

Para poder simular nuestro tubo lo que haremos será generar una malla que constará de una pared cilíndrica y dos paredes, una de entrada y otra de salida. Con esto podremos emular a la parte interna del tubo de acero inoxidable por la que pasará el fluido.

Generar una malla desde cero es algo complejo y costoso, que no es el objeto de estudio de este trabajo, y teniendo en cuenta que el programa que estamos usando es de código abierto, hemos seleccionado una malla cilíndrica ya completada. Como hemos comentado anteriormente podemos modificar los archivos de la malla, por tanto, utilizaremos estos y los modificaremos para nuestro caso.

Obtenemos un archivo desde una página web de internet [7], llamado “blockMeshDict.m4”, este archivo lo guardamos en la carpeta en la que queramos guardar la malla de este caso. Copiamos igualmente en esta dirección las carpetas “0”, “constant” y “system”, desde un tutorial cualquiera. Tenemos la libertad de hacer esto ya que en este momento sólo es de nuestro interés la generación o modificación de nuestra malla, y no las condiciones iniciales, de contorno o los parámetros del método numérico.

Se nos quedará la carpeta elegida con este contenido:



Carpeta de la malla 4.19

Podemos abrir el archivo “blockMeshDict.m4” directamente pinchando en él y abriéndolo en el bloc de notas o podemos abrirlo desde el terminal del programa.

De cualquier forma, nos aparecerá un fichero con el siguiente contenido:

```

8 FoamFile
9 {
10     version      2.0;
11     `format'     ascii;
12     class        dictionary;
13     object       blockMeshDict;
14 }
15 // ***** //
16 // General macros to create cylinder mesh
17
18 changeCom(//)changeQuote([,])
19 define(calc, [esyscmd(perl -e 'use Math::Trig; print ($1)'])])
20 define(vcount, 0)
21 define(vlabel, [[// ]vertex $1 = vcount define($1, vcount)define([vcount], incr(vcount))])
22
23 define(hex2D, hex ($1b $2b $3b $4b $1t $2t $3t $4t))
24 define(btquad, ($1b $2b $2t $1t))
25 define(topquad, ($1t $4t $3t $2t))
26 define(bottomquad, ($1b $2b $3b $4b))
27

```

Definición de los vértices 4.20

En esta parte no cambiaremos nada, ya que lo que se definen aquí son los vértices de las diferentes geometrías que forman el cilindro.

A continuación de esta última parte veremos:

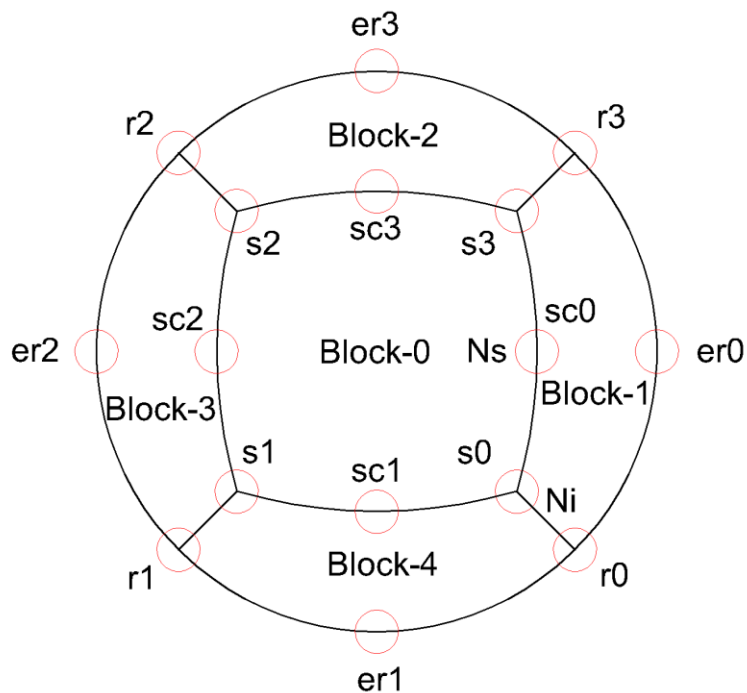
```

29
30 convertToMeters 0.001;
31
32 // Inner square side half
33 define(s, 18.28)
34
35 // Inner square side curvature
36 define(sc, 20.0)
37
38 // cylinder radius
39 define(r, 36.575)
40
41 // Height of cylinder
42 define(z, 1500.0)
43
44 // Base z
45 define(Zb, 0)
46
47 // outlet z
48 define(Zt, calc(Zb + z))
49
50 // Number of cells at inner square
51 define(Ns, 40)
52
53 // Number of cells between inner square and circle
54 define(Ni, 20)
55
56 // Number of cells in the cylinder height
57 define(Nz, 150)
58

```

Definición número celdas 4.21

Esta parte será la más importante ya que es donde introduciremos las medidas para nuestro caso. En primer lugar, la página web [7] nos ofrece una imagen con la vista previa de las bases del cilindro y las partes en las que se compone:



Partes de la base del cilindro 4.22

Teniendo en cuenta esta disposición procederemos a explicar y cambiar los comandos necesarios: En la primera línea nos encontramos con el comando “convertToMeters”, este comando sirve para

cambiar a sistema internacional los valores que introduzcamos, es decir, deberemos meter valores en milímetros, pues el valor que le acompaña es 0.001.

En “InnerSquareSideHalf” introduciremos la mitad del radio de nuestro cilindro.

El radio del cilindro lo introduciremos en “cylinder radius” y la altura o longitud del cilindro lo haremos en “Height of cylinder”.

Estos valores no cambiarán a lo largo de la simulación, sin embargo, el número de elementos de la malla sí.

Si nos fijamos en la figura, podremos introducir el número de celdas o elementos que tienen los lados “Ns” y “Ni”, y los elementos a lo largo de la longitud del cilindro.

Recordamos que, para obtener buenos resultados numéricos, deberemos intentar tener el mayor número de elementos. Empezamos construyendo una malla de tres cientos mil elementos. Por cuestiones de potencia y memoria del equipo disponible para realizar estos cambios, reduciremos la creación de la malla a la mitad de esos elementos.

El cálculo exacto de las celdas lo haremos a mano, para poder sacar cerca de ciento cincuenta mil elegiremos:

- $N_s=80$
- $N_i=40$
- $N_z=400$

Esto nos dará en el perímetro de la circunferencia de cada base:

$$80 \times 4 = 320 \text{ elementos} \quad (4.4)$$

Esto significará una cantidad de elementos en la pared cilíndrica de:

$$320 \times 400 = 128000 \text{ elementos} \quad (4.5)$$

Si calculamos el área de cada base, consta del rectángulo interior formado por los lados Ns, y las cuatro zonas exteriores a este:

$$80 \times 80 + 4 \times 80 \times 40 = 19200 \quad (4.6)$$

En total tendremos:

$$128000 + 2 \times 19200 = 166400 \text{ elementos} \quad (4.7)$$

En el equipo que hemos utilizado, esta cantidad requerirá un tiempo de espera al usuario, pero podemos simularlo para poder visualizar la malla. Aunque cuando queramos introducir condiciones de contorno en esta malla deberemos reducir los elementos, para que el equipo nos lo permita y para facilitar el estudio del problema.

Una vez introducidos estos cambios, guardaremos el archivo.

Como no hemos modificado directamente el documento responsable de crear la malla en el programa lo que deberemos hacer será lo siguiente:

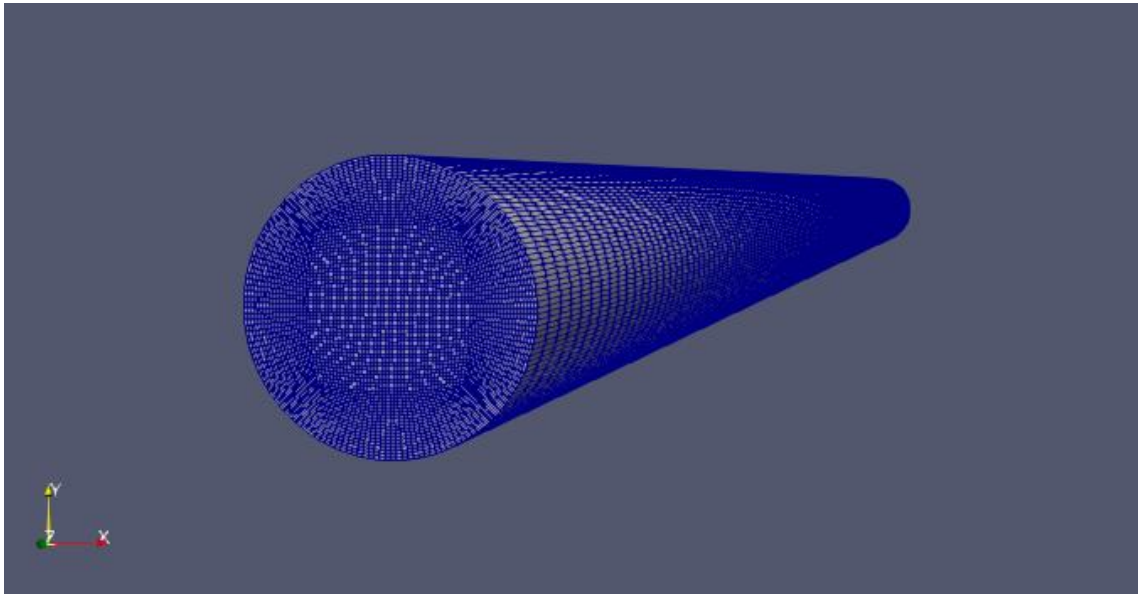
- run
- cd “nombre de la carpeta donde lo hayamos guardado”
- Borrarnos el archivo “blockMeshDict” que se encuentra en la carpeta “system”.
- m4 blockMeshDict.m4 > system/blockMeshDict

Lo que hará este comando será guardar el fichero que hemos modificado como el fichero

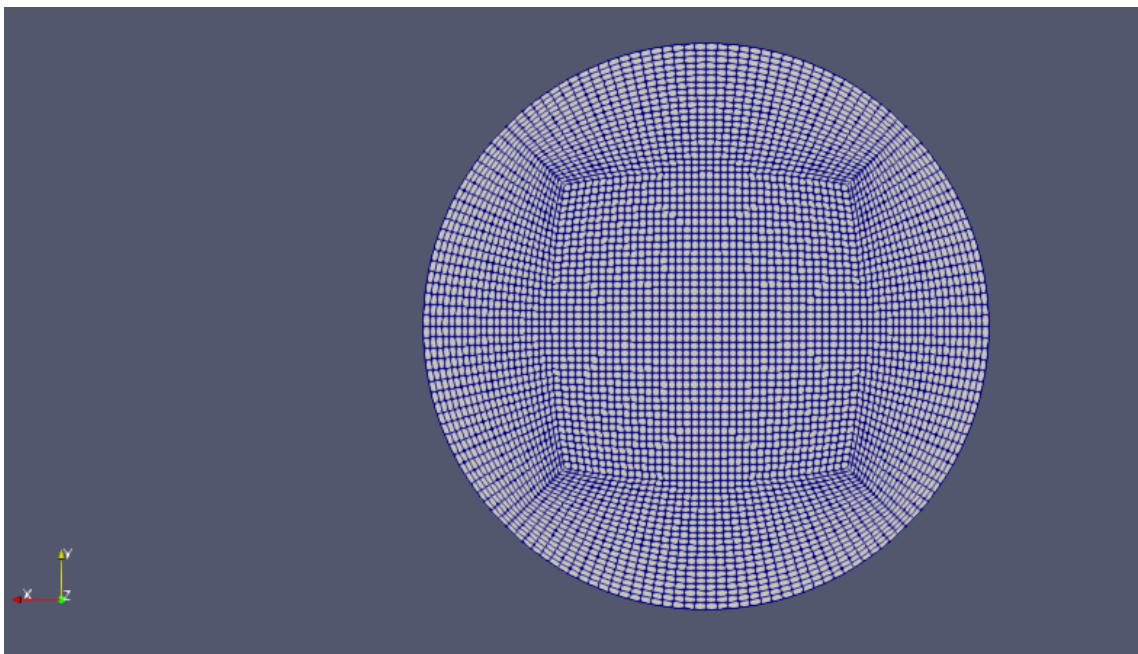
responsable de generar la malla.

- Comprobamos que se ha generado este archivo.
- Generamos la malla:
- blockMesh
- paraFoam &

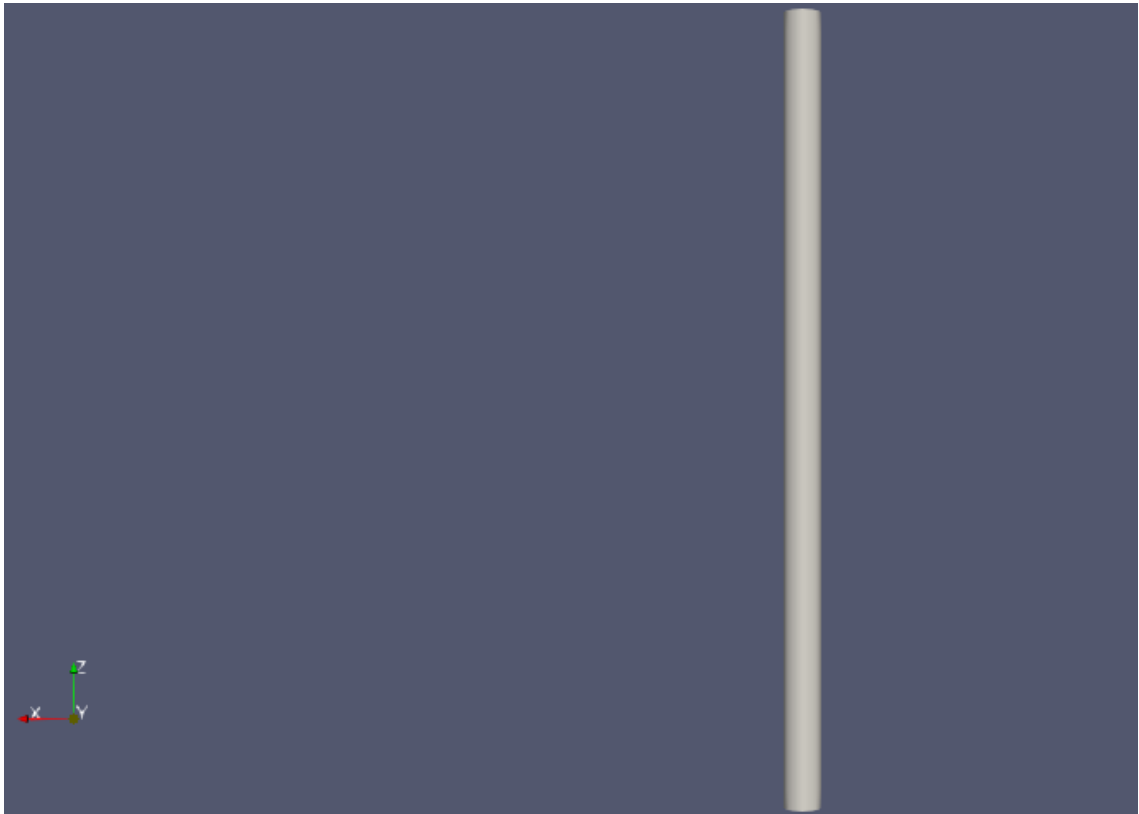
ParaView nos mostrará una imagen del mallado:



Visualización 3D cilindro 4.23



Vista base del cilindro 4.24



Vista superior cilindro 4.25

4.6- Condiciones de contorno.

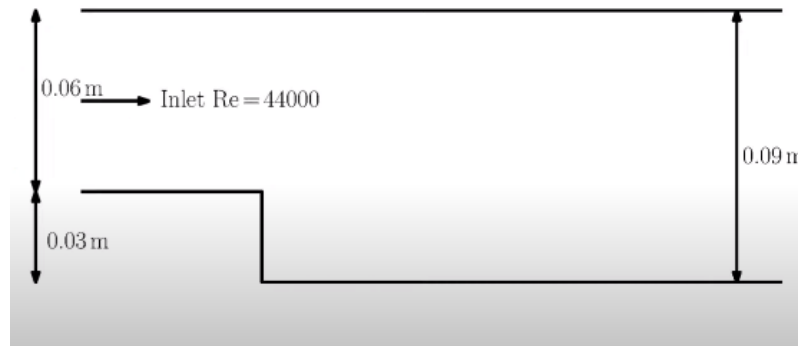
Una vez hemos obtenido la malla donde simular el flujo, deberemos imponer unas condiciones de contorno a la misma y utilizar un “solver” para poder simular las variaciones de tiempo en dichas condiciones.

4.6.1- Tutorial con SimpleFoam.

Para nuestra situación usaremos el “solver”: “bouyantBoussinesqPimpleFoam”, ya que es conveniente, pero antes de eso deberemos analizar como funciona más a fondo cada una de estas variantes.

Como guía de aprendizaje podemos seguir el tutorial realizado por József Nagy [6.9, 6.10 y 6.11 en la biografía]. Trata sobre la introducción a al modelado del régimen estacionario turbulento. Nos aporta también la malla generada para ese caso.

Al principio nos muestran el problema a resolver, del que se creará una malla para ser representado de tal forma que cumpla:



Problema tutorial [József Nagy]4.26

Es decir, una entrada de 0.06 m, en régimen turbulento con un número de Reynolds igual a 44000, y una salida de 0.09 m.

SimpleFoam se trata de un “solver” que se puede definir en distintos puntos:

1. Incompresible, ya que trata con fluidos incompresibles, y utiliza las siguientes ecuaciones:

- Ecuación de continuidad.

$$\nabla \cdot u = 0 \tag{4.8}$$

Donde u=velocidad.

- Ecuación de movimiento.

$$\nabla \cdot (uu) = -\frac{1}{\rho} \nabla \cdot \bar{p} + \nabla \cdot \nu[2S] + Ft \tag{4.9}$$

El primer término se trata del término convectivo que en el régimen turbulento será el dominante, el segundo término en la parte derecha de la igualdad se trata del término difusivo que será el menos dominante.

- Donde el termino de presión, que no así la presión, viene definido por:

$$p = \frac{1}{\rho} \bar{p} \tag{4.10}$$

Donde \bar{p} es la presión.

2. Estacionario, ya que no hay derivadas parciales de tiempo.
3. Turbulento, definido por el parámetro Ft de la ecuación de cantidad de movimiento.
4. Fase única, ya que no hay tratamientos adicionales.
5. Isotérmico, no hay ecuación de energía.

Dentro del propio “SimpleFoam” tendremos diferentes modelos RANS para resolver:

- k-epsilon model
- k-omega model

En el primero tendremos las siguientes variables principales:

1. k como la energía turbulenta:

$$k = \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) \quad (4.11)$$

2. ε que representa la disipación de la turbulencia.
3. Viscosidad turbulenta:

$$\nu t = 0.09 \frac{k^2}{\varepsilon} \quad (4.12)$$

En el segundo modelo las variables que usaremos son:

1. La misma k que en el anterior.
2. w como el rango específico de disipación turbulenta.
3. La viscosidad turbulenta vendrá dada ahora por:

$$\nu t = \frac{k}{w} \quad (4.13)$$

Como tener estos datos exactos es una tarea muy compleja, usaremos estimaciones aproximadas para este caso. Las condiciones de contorno en la entrada (“inlet”) que introduciremos son:

- Para k:

$$k = \frac{3}{2}(U_{ref} \cdot Ti)^2 \quad (4.14)$$

Donde U_{ref} es la velocidad del fluido a la entrada y Ti la intensidad de la turbulencia (%).

- Para ε :

$$\varepsilon = 0.09^{0.75} \frac{k^{1.5}}{l} \quad (4.15)$$

Donde $l = 0.07L$, L es la longitud característica.

- Para w:

$$w = \frac{\varepsilon}{k} \quad (4.16)$$

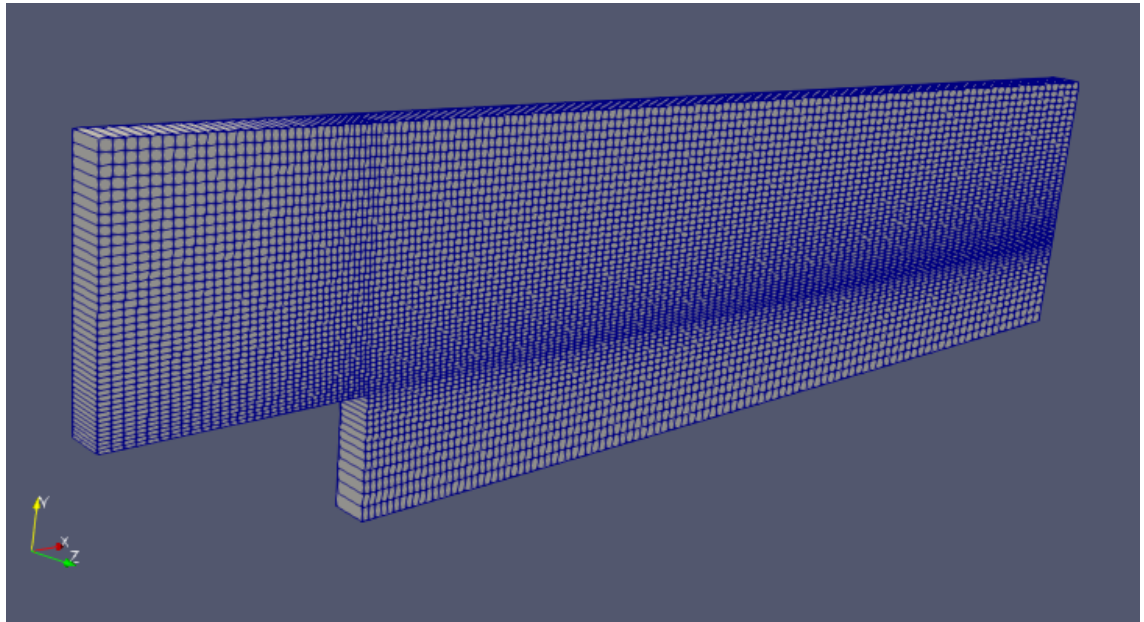
Una vez introducida la parte teórica podemos pasar a la parte práctica.

Primero usaremos k- epsilon model.

4.6.1.1- k- épsilon model.

Deberemos seguir los siguientes pasos para poder simular el caso (asumimos que la dirección run ya está creada):

1. Creamos en run una carpeta para el caso, que cogemos de “pitzDaily”:
 - run
 - cp -r pitzDaily backwardFacingStep_baseCase
 - cd backwardFacingStep_baseCase
2. Descargamos el “blockMeshDict” que nos ofrece el tutorial y lo sustituiremos en el de esta carpeta:
 - cp blockMeshDict backwardFacingStep_basecase/system/blockMeshDict
3. Creamos la malla:
 - blockMesh
 - paraFoam &



Malla simplefoam 4.27

4. Abrimos el fichero donde podremos ver los detalles de esta malla:

- nano constant/polyMesh/boundary
- Nos encontraremos:

```

22 (
23     outlet
24     {
25         type            patch;
26         nFaces          55;
27         startFace       13452;
28     }
29     inlet
30     {
31         type            patch;
32         nFaces          40;
33         startFace       13507;
34     }
35     lowerwall
36     {
37         type            wall;
38         inGroups        1(wall);
39         nFaces          148;
40         startFace       13547;
41     }
42     upperwall
43     {
44         type            wall;
45         inGroups        1(wall);
46         nFaces          133;
47         startFace       13695;
48     }
49     defaultFaces
50     {
51         type            empty;
52         inGroups        1(empty);
53         nFaces          13640;
54         startFace       13828;
55     }
56 )

```

Archivo boundary 4.28

Esto es importante, ya que podemos ver como se ha denominado a cada parte de la malla, “inlet” y “outlet” para la entrada y la salida respectivamente; “lowerwall” y “upperwall” para la parte inferior de la malla y para la superior.

En “defaultFaces” lo cambiaremos por “frontAndBack” para llamar así a los laterales de la malla que tienen la geometría que hemos visto en 2D al explicar la parte teórica del caso.

Podemos observar el número de elementos de cada parte de la malla en “nFaces”.

Tomaremos estos comandos escritos ya y los copiaremos para ir pegando en cada

propiedad a cambiar, ya que los archivos que usemos para simular las propiedades deberán contener el mismo nombre para las partes de la malla que en el archivo donde hemos creado la propia malla.

5. Así iremos cambiando cada propiedad, empezamos por la presión:

- cd 0
- nano p
- El contenido debe quedar así:

```
16
17 dimensions      [0 2 -2 0 0 0 0];
18
19 internalField    uniform 0;
20
21 boundaryField
22 {
23     inlet
24     {
25         type      zeroGradient;
26     }
27
28     outlet
29     {
30         type      fixedValue;
31         value     uniform 0;
32     }
33
34     upperwall
35     {
36         type      zeroGradient;
37     }
38
39     lowerwall
40     {
41         type      zeroGradient;
42     }
43
44     frontAndBack
45     {
46         type      empty;
47     }
48 }
```

Archivo presión 4.29

Sólo estableceremos un valor fijo en la salida, en este caso será 0.

6. Pasamos a cambiar la velocidad u.

- nano u

Deberemos establecer un valor a la entrada fijo, sabemos el número de Reynolds que deseamos establecer, por tanto, calculamos la velocidad.

$$Re = \frac{U \cdot L}{\nu} = \frac{x \cdot 0.06}{1 \cdot e^{-5}} = 44000 \quad (4.17)$$

$$U = 7.33 \text{ m/s} \quad (4.18)$$

Introduciremos esta velocidad en la entrada, en la dirección x.

```

17 dimensions      [0 1 -1 0 0 0];
18
19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     inlet
24     {
25         type      fixedValue;
26         value     uniform (7.333 0 0);
27     }
28
29     outlet
30     {
31         type      zeroGradient;
32     }
33
34     upperwall
35     {
36         type      noSlip;
37     }
38
39     lowerwall
40     {
41         type      noSlip;
42     }
43
44     frontAndBack
45     {
46         type      empty;
47     }

```

Archivo U (3) 4.30

7. Cambiamos k.

- nano k
- Calculamos el valor específico con el que inicializar la simulación.

$$k = \frac{3}{2}(U_{ref} \cdot Ti)^2 = 1.5 \cdot (7.33 \cdot 0.05)^2 = 0.2 \quad (4.19)$$

```
17
18 dimensions      [0 2 -2 0 0 0 0];
19
20 internalField    uniform 0.2;
21
22 boundaryField
23 {
24     inlet
25     {
26         type      fixedValue;
27         value      uniform 0.2;
28     }
29     outlet
30     {
31         type      zeroGradient;
32     }
33     upperwall
34     {
35         type      kqRwallFunction;
36         value      uniform 0.2;
37     }
38     lowerwall
39     {
40         type      kqRwallFunction;
41         value      uniform 0.2;
42     }
43     frontAndBack
44     {
45         type      empty;
46     }
47 }
```

Archivo k 4.31

8. Cambiamos ahora ϵ .

- nano epsilon
- Debemos calcular este valor también con las estimaciones especificadas anteriormente.

$$\epsilon = 0.09^{0.75} \frac{k^{1.5}}{l} = 0.09^{0.75} \frac{0.2^{1.5}}{0.07 * 0.06} = 3.54 \quad (4.20)$$

```
17
18 dimensions      [0 2 -3 0 0 0 0];
19
20 internalField   uniform 3.54;
21
22 boundaryField
23 {
24     inlet
25     {
26         type      fixedValue;
27         value     uniform 3.54;
28     }
29     outlet
30     {
31         type      zeroGradient;
32     }
33     upperwall
34     {
35         type      epsilonWallFunction;
36         value     uniform 3.54;
37     }
38     lowerwall
39     {
40         type      epsilonWallFunction;
41         value     uniform 3.54;
42     }
43     frontAndBack
44     {
45         type      empty;
46     }
47 }
48
```

Archivo *epsilon* 4.32

9. Una vez establecidas las condiciones de contorno, salimos y vamos a la carpeta “transportProperties”

- cd ..
- nano constant/transportProperties

```
17
18 simulationType RAS;
19
20 RAS
21 {
22     // Tested with kEpsilon, realizableKE, komega, komegasST, v2f,
23     // ShihQuadraticKE, LienCubicKE.
24     RASModel      kEpsilon;
25
26     turbulence     on;
27
28     printCoeffs   on;
29 }
30
```

Archivo Properties 4.33

En este caso, el modelo que está utilizando es el k-epsilon. Cuando deseemos utilizar otro deberemos cambiarlo ahí. También apreciamos que la turbulencia está activada.

10. Salimos y vamos a “system”

- cd ..
- nano controlDict

Aquí quitaremos la última parte, quedándonos el fichero de la siguiente forma:

```
17
18 application      simpleFoam;
19
20 startFrom        startTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          2000;
27
28 deltaT           1;
29
30 writeControl     timeStep;
31
32 writeInterval    100;
33
34 purgeWrite       0;
35
36 writeFormat      ascii;
37
38 writePrecision   6;
39
40 writeCompression off;
41
42 timeFormat       general;
43
44 timePrecision    6;
45
46 runtimeModifiable true;
47
```

Archivo controlDict 4.34

Lo más relevante de este archivo y que podemos modificar es que comenzamos el tiempo en 0 en “startTime”, acabará en 2000 en “endTime”, “deltaT” nos indica los intervalos de tiempo, y “writeInterval” cada cuanto tiempo escribirá o guardará una simulación en nuestro equipo.

11. Entramos en “fvSchemes” para cambiar la divergencia de épsilon.

- nano fvSchemes

Debe quedar:

```

28 divSchemes
29 {
30     default          none;
31     div(phi,u)       bounded Gauss linearUpwind grad(u);
32     div(phi,k)       bounded Gauss linearUpwind grad(k);
33     div(phi,epsilon) bounded Gauss linearUpwind grad(epsilon);
34     div(phi,omega)   bounded Gauss limitedLinear 1;
35     div(phi,v2)      bounded Gauss limitedLinear 1;
36     div((nuEff*dev2(T(grad(u)))) Gauss linear;
37     div(nonlinearStress) Gauss linear;
38 }

```

Archivo fvSchemes 4.35

12. Entraremos por último en “fvSolution”.

- nano fvSolution
- Ajustaremos los siguientes valores:

```

18 solvers
19 {
20     p
21     {
22         solver          GAMG;
23         tolerance       1e-06;
24         relTol          1e-06;
25         smoother        GaussSeidel;
26     }
27     "(u|k|epsilon|omega|f|v2)"
28     {
29         solver          smoothSolver;
30         smoother        symGaussSeidel;
31         tolerance       1e-05;
32         relTol          1e-05;
33     }
34 }
35 }

```

Archivo fvSolution 4.36

Debemos borrar los controles residuales, y estableceremos en los factores de relajación:

```

43 relaxationFactors
44 {
45     fields
46     {
47         p            0.3; //
48     }
49     equations
50     {
51         U            0.7; //
52         k            0.7; //
53         epsilon     0.7; //
54         R            0.7; //
55         nuTilda     0.7; //
56     }
57 }
    
```

Factores de relajación 4.37

En la parte de “equations” hemos puesto valores de 0.7 que nos favorece una simulación más estable, aunque la convergencia sea más lenta.

13. Ya podemos ver la simulación, aunque lo visualizaremos después para comparar los diferentes modelos de “SimpleFoam”.

4.6.1.2- k- omega model.

Una vez realizado los cambios necesarios, lo que haremos será copiar los ficheros y nombrarlos de forma diferente para cada modelo, ejecutaremos en el terminal:

- cp -r backwardFacingStep_baseCase/ backwardFacingStep_kEpsilon
- cp -r backwardFacingStep_baseCase/ backwardFacingStep_kOmega

Una vez realizado esto lo que haremos será entrar en el modelo “k-Omega” y cambiar las propiedades de este caso.

1. Entramos en el modelo:
 - cd backwardfacingStep_kOmega
2. Primero deberemos cambiar el modelo en las propiedades.
 - nano constant/turbulenceProperties

Hacemos el cambio:

```

18 simulationType RAS;
19
20 RAS
21 {
22     // Tested with kEpsilon, realizableKE, komega, komegaSST, v2f,
23     // shihQuadraticKE, LienCubicKE.
24     RASModel        kOmega;
25
26     turbulence      on;
27
28     printCoeffs    on;
29 }
    
```

Archivo Properties (2) 4.38

3. Ahora sólo hay que cambiar el archivo de omega.

- cd 0
- nano omega
- Debemos calcular el valor de omega que inicializa la simulación:

$$w = \frac{\varepsilon}{k} = \frac{3.54}{0.2} = 17.57 \quad (4.21)$$

Lo introducimos:

```

17
18 dimensions      [0 0 -1 0 0 0 0];
19
20 internalField   uniform 17.54;
21
22 boundaryField
23 {
24     inlet
25     {
26         type      fixedvalue;
27         value     $internalField;
28     }
29     outlet
30     {
31         type      zeroGradient;
32     }
33     upperwall
34     {
35         type      omegawallFunction;
36         value     $internalField;
37     }
38     lowerwall
39     {
40         type      omegawallFunction;
41         value     $internalField;
42     }
43     frontAndBack
44     {
45         type      empty;
46     }
47 }
```

Archivo omega 4.39

4. Cambiamos la divergencia de omega.

- cd ..
- cd system
- nano fvSchemes

```

28 divSchemes
29 {
30     default          none;
31     div(phi,u)       bounded Gauss linearUpwind grad(u);
32     div(phi,k)       bounded Gauss linearUpwind grad(k);
33     div(phi,epsilon) bounded Gauss linearUpwind grad(epsilon);
34     div(phi,omega)   bounded Gauss linearUpwind grad(omega);
35     div(phi,v2)      bounded Gauss limitedLinear 1;
36     div((nuEff*dev2(T(grad(u)))) Gauss linear;
37     div(nonlinearstress) Gauss linear;
38 }
    
```

Archivo fvSchemes (2) 4.40

4.6.1.3- Comparación entre modelos.

Pasamos ahora a comparar las diferencias entre usar un modelo y otro, lo haremos visualmente en “ParaView”, esto nos mostrará que podemos resolver el mismo caso de forma diferente, según definamos la viscosidad turbulenta y las variables que elijamos.

Para simular cada caso, una vez creada la malla lo que haremos será crear las diferentes carpetas de tiempo en nuestro equipo. Ejecutando el siguiente comando:

- SimpleFoam

El programa automáticamente nos creará en la carpeta del caso los archivos necesarios.

0	21/07/2020 11:40	Carpeta de archivos
100	21/07/2020 17:47	Carpeta de archivos
200	21/07/2020 17:47	Carpeta de archivos
300	21/07/2020 17:47	Carpeta de archivos
400	21/07/2020 17:47	Carpeta de archivos
500	21/07/2020 17:47	Carpeta de archivos
600	21/07/2020 17:47	Carpeta de archivos
700	21/07/2020 17:47	Carpeta de archivos
800	21/07/2020 17:47	Carpeta de archivos
900	21/07/2020 17:47	Carpeta de archivos
1000	21/07/2020 17:47	Carpeta de archivos
1100	21/07/2020 17:48	Carpeta de archivos
1200	21/07/2020 17:48	Carpeta de archivos
1300	21/07/2020 17:48	Carpeta de archivos
1400	21/07/2020 17:48	Carpeta de archivos
1500	21/07/2020 17:48	Carpeta de archivos
1600	21/07/2020 17:48	Carpeta de archivos
1700	21/07/2020 17:48	Carpeta de archivos
1800	21/07/2020 17:48	Carpeta de archivos
1900	21/07/2020 17:48	Carpeta de archivos
2000	21/07/2020 17:48	Carpeta de archivos
constant	29/07/2020 12:53	Carpeta de archivos
system	21/07/2020 11:40	Carpeta de archivos

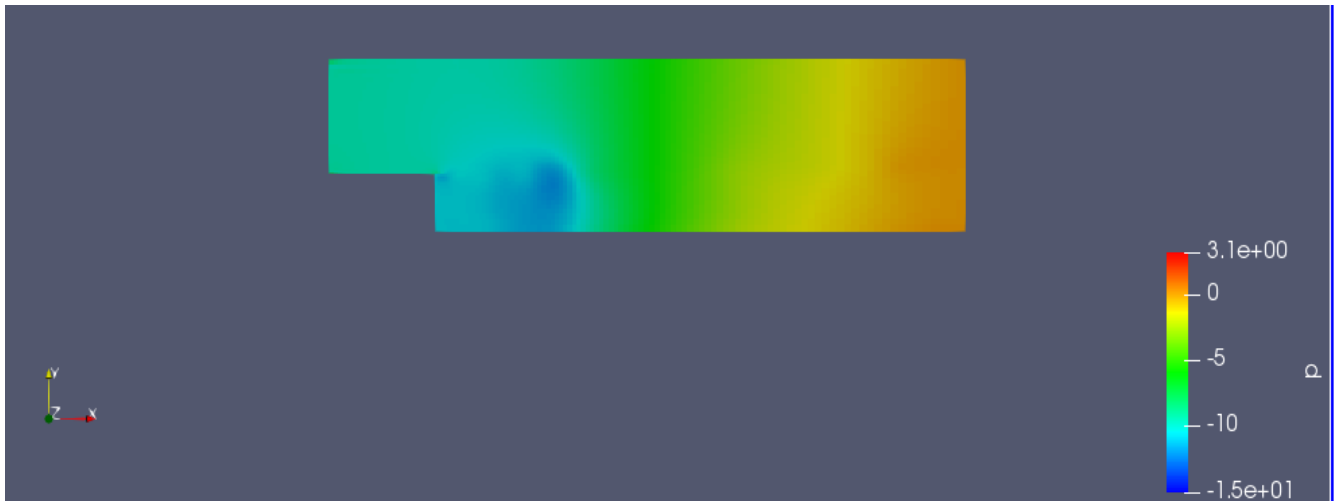
Carpetas generadas 4.41

Comprobaremos cada vez que ejecutemos un comando de este estilo que dichas carpetas se han guardado en la dirección correcta, ya que el programa simulará una a una éstas.

Una vez creadas estas carpetas, podemos ver la simulación de cada modelo. Tendremos la siguiente visualización de cada propiedad, hay que destacar que no varía significativamente a lo largo del tiempo al ser un caso estacionario, escogemos el último intervalo de tiempo:

1. Presión:

- k-epsilon model.



Presión k-ε model 4.42

- k-omega model.

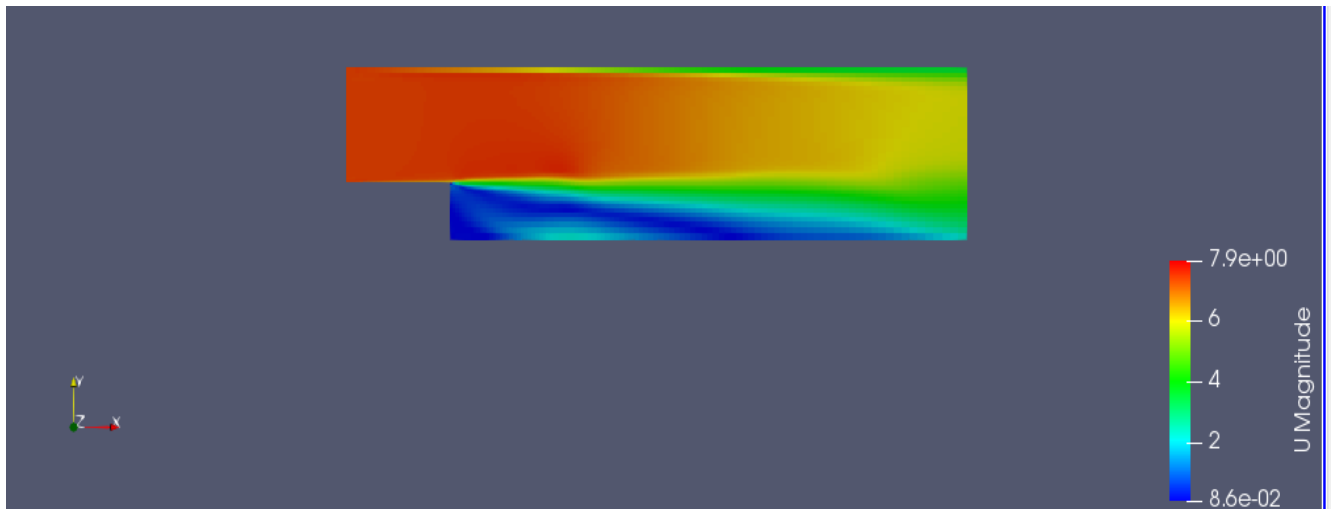


Presión k-ω model 4.43

Vemos diferencias significativas tanto a la entrada como a la salida usando diferentes modelos.

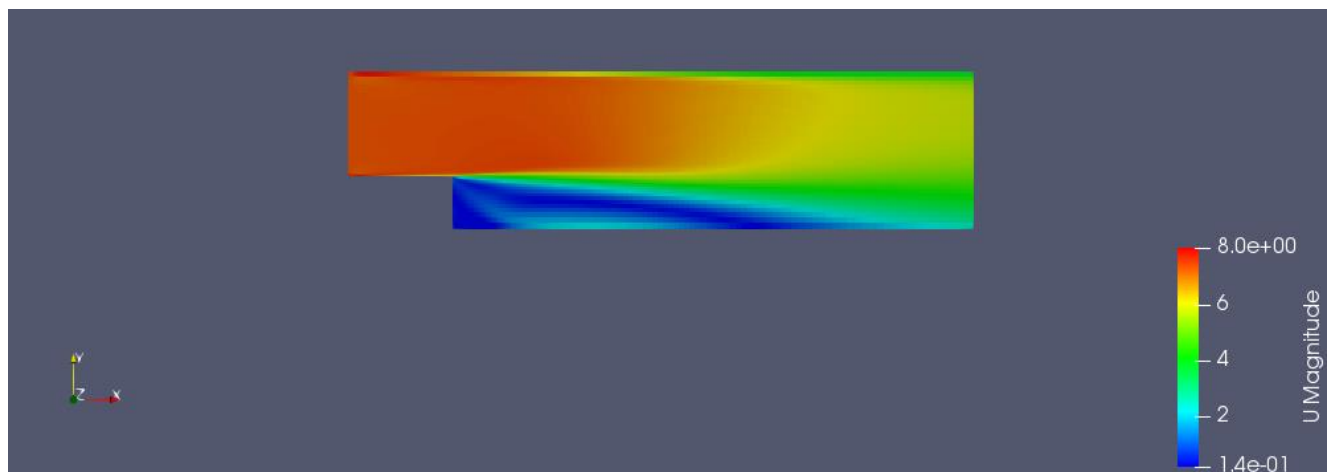
2. Velocidad.

- k- epsilon model.



Velocidad k- ϵ model 4.44

- k- omega model.

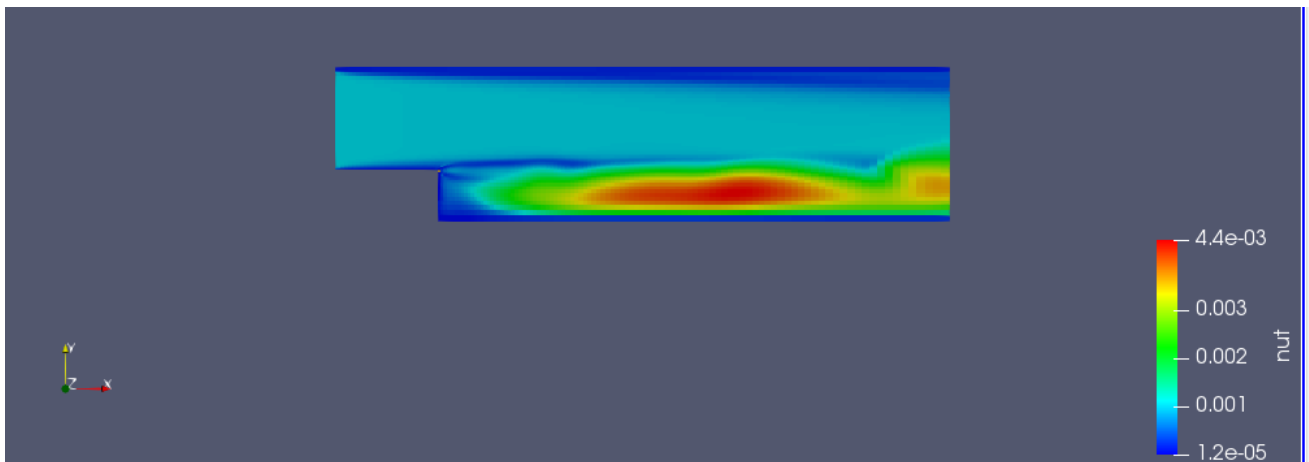


Velocidad k- ω model 4.45

Para la velocidad, tenemos resultados muy parecidos, ya que hemos impuesto nosotros dicha propiedad a partir del número de Reynolds.

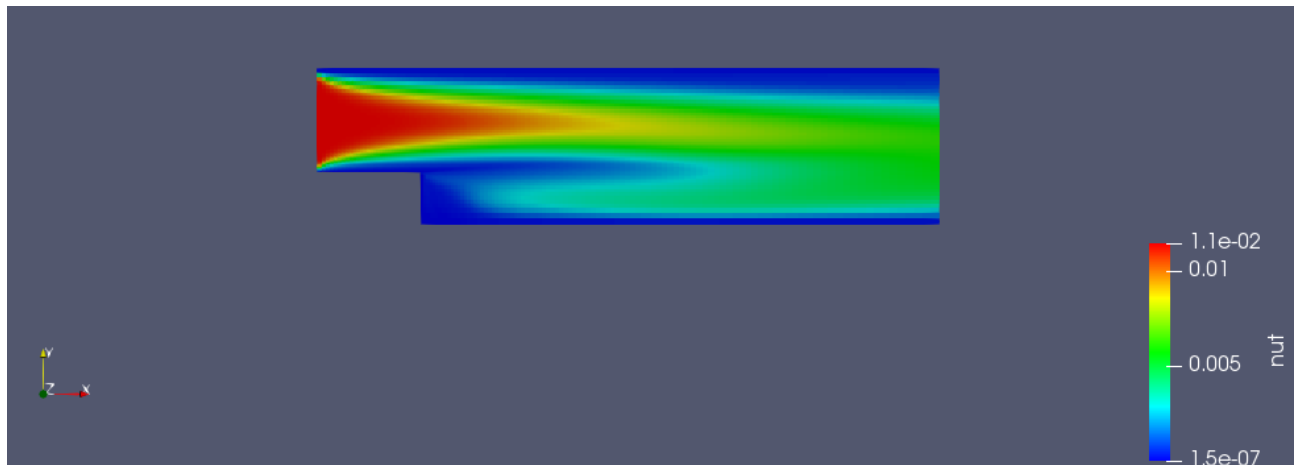
3. Viscosidad.

- k-epsilon model.



nut k- ϵ model 4.46

- k-omega model.

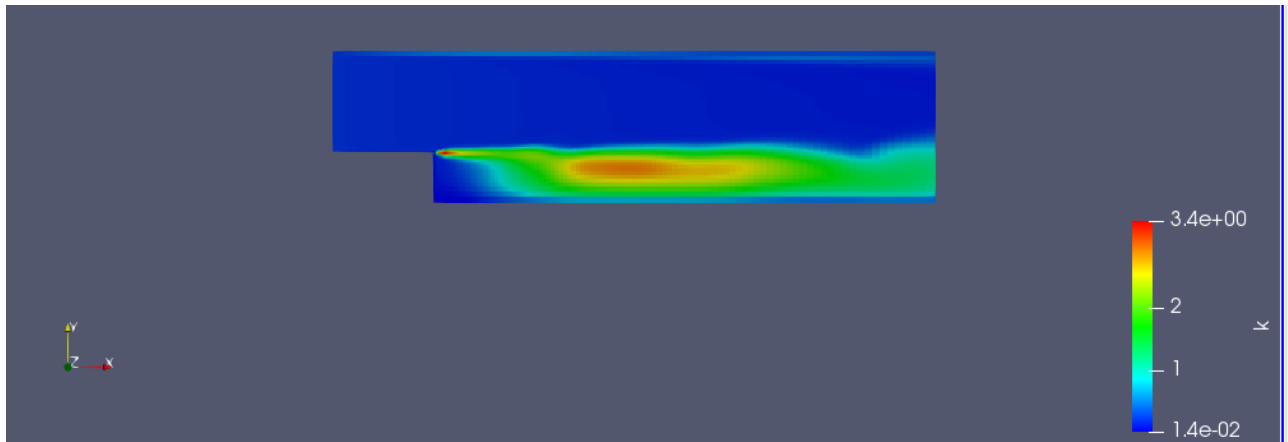


nut k- ω model 4.47

En este caso podemos apreciar diferencias muy grandes, en parte debido por el modelo usado, y por otra parte por cómo hemos definido la viscosidad turbulenta en cada uno de los modelos.

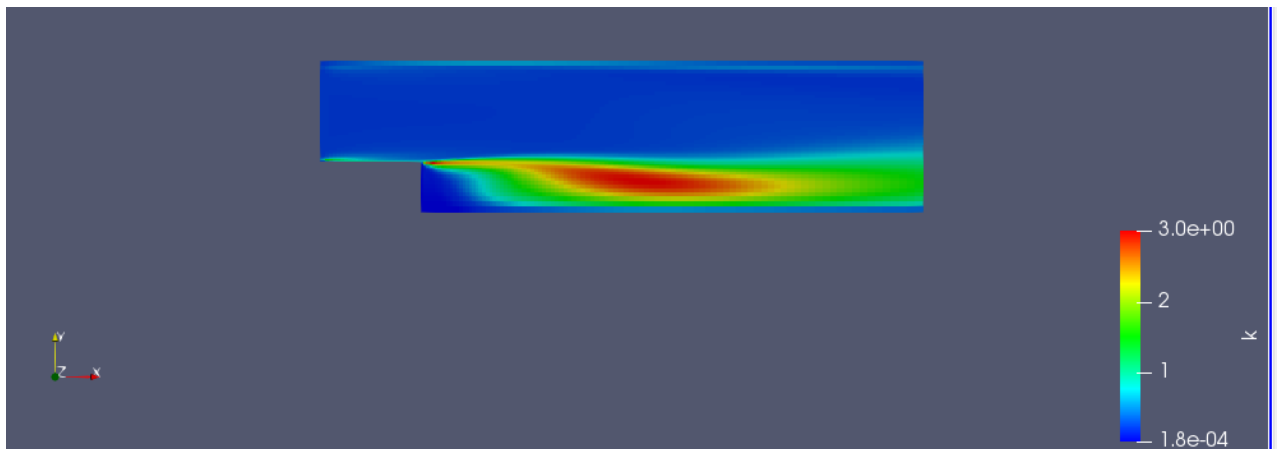
4. Energía cinética turbulenta k .

- k -epsilon model.



k k-ε model 4.48

- k -omega model.

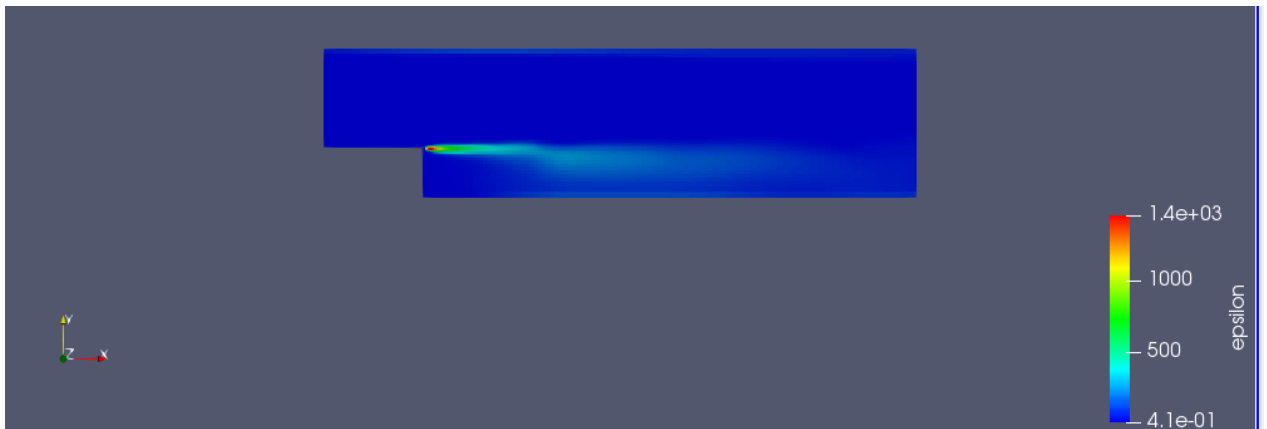


k k-ω model 4.49

Podemos apreciar que son muy parecidos, aunque el modelo k -omega parece que promedia más uniformemente el campo espacial para el lapso de tiempo simulado.

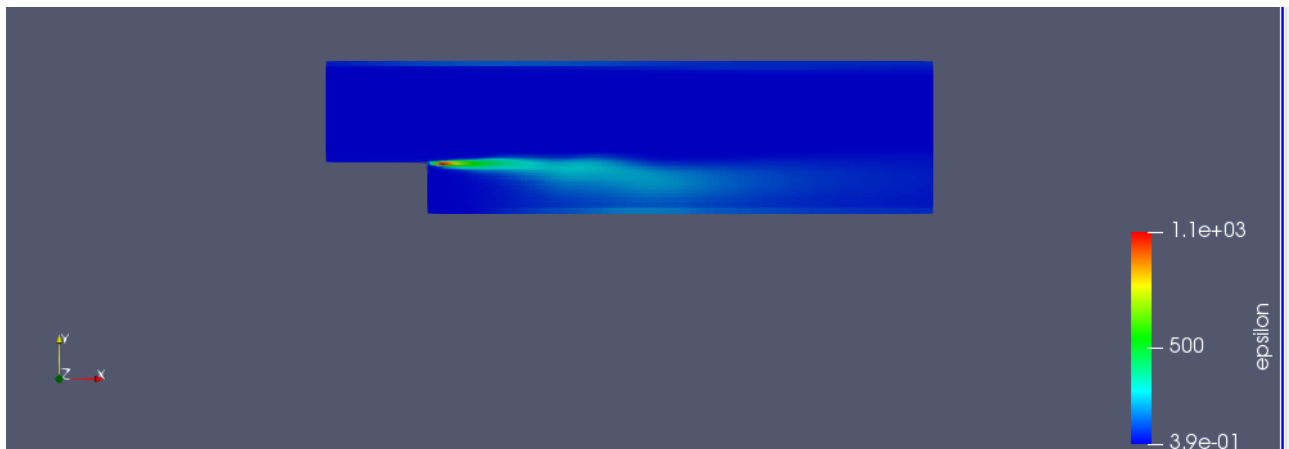
5. Epsilon.

- k- epsilon model.



Épsilon k- ϵ model 4.50

- k-omega model.



Épsilon k- ω model 4.51

6. Omega.

Por último, tendremos omega que sólo podremos ver en el modelo k-w.



Omega k- w model 4.52

4.6.2- Tutorial “buoyantBoussinesqPimpleFoam” para introducir condiciones de contorno.

Utilizaremos el “solver” de OpenFoam llamado “buoyantBoussinesqPimpleFoam”. Nos ayudaremos de un tutorial que presenta un caso similar al nuestro [6.12 y 6.13 en la bibliografía].

Con este tutorial desarrollado en este TFG cambiaremos las condiciones especificadas anteriormente para nuestra malla.

1. Debemos crear una carpeta nueva en la dirección de “run” para este caso a parte de la que teníamos para la creación de la malla.

La podemos llamar “cilindroconcondicionescontorno”.

A partir de ahora trabajaremos con ésta.

2. Nos vamos a la dirección dentro de las carpetas del programa a OpenFOAM-5.x/tutorials/heatTransfer/buoyantBoussinesqPimpleFoam/hotRoom. Copiamos el contenido que hay en su interior y lo pegamos en nuestra carpeta de trabajo.
3. Sustituimos “blockMeshDict” de nuestra malla por el del tutorial.
4. En el terminal introducimos los siguientes comandos para comprobar que se crea la malla:

- cd cilindroconcondicionescontorno
- blockMesh
- paraFoam &

Esto creará un archivo llamado “PolyMesh” en “constant”. Si entramos en éste podremos ver el archivo “boundary” donde veremos los nombres de los límites de nuestra malla y que serán con los que trabajemos.

- cd ..
- cd constant

- cd PolyMesh
- nano boundary

```
21 3
22 (
23     walls
24     {
25         type            wall;
26         inGroups       1(wall);
27         nFaces         24000;
28         startFace     2143200;
29     }
30     inlet
31     {
32         type            patch;
33         nFaces         4800;
34         startFace     2167200;
35     }
36     outlet
37     {
38         type            patch;
39         nFaces         4800;
40         startFace     2172000;
41     }
42 )
```

Archivo boundary (2) 4.53

Usaremos estos límites y los iremos cambiando en las distintas propiedades del tutorial.

5. Empezamos a cambiar propiedades, cambiamos “aphat”. Para nuestro caso utilizaremos el valor que propone en el tutorial.

- cd ..
- cd ..
- nano 0/alphat

```
17
18 dimensions      [0 2 -1 0 0 0 0];
19
20 internalField   uniform 2.3039e-5;
21
22 boundaryField
23 {
24     walls
25     {
26         type      alphasJayatilekewallFunction;
27         Prt       0.71;
28         value     $internalField;
29     }
30     inlet
31     {
32         type      fixedValue;
33         value     $internalField;
34     }
35     outlet
36     {
37         type      zeroGradient;
38     }
39 }
40
```

Archivo U alphas 4.54

Hemos establecido un valor con el que inicializar y las funciones que nos proporciona el tutorial.

6. Cambiamos “épsilon”. Utilizando los valores del tutorial.

- nano 0/epsilon

```
17
18 dimensions      [0 2 -3 0 0 0 0];
19
20 internalField    uniform 5e-5;
21
22 boundaryField
23 {
24     walls
25     {
26         type      epsilonWallFunction;
27         value      $internalField;
28     }
29     inlet
30     {
31         type      fixedValue;
32         value      $internalField;
33     }
34     outlet
35     {
36         type      zeroGradient;
37     }
38 }
```

Archivo epsilon 4.55

7. Procedemos igualmente con “k”.

```
18 dimensions      [0 2 -2 0 0 0 0];
19
20 internalField    uniform 1e-4;
21
22 boundaryField
23 {
24     walls
25     {
26         type      kqRWallFunction;
27         value      $internalField;
28     }
29     inlet
30     {
31         type      fixedValue;
32         value      $internalField;
33     }
34     outlet
35     {
36         type      zeroGradient;
37     }
38 }
```

Archivo k 4.56

8. Pasamos a la viscosidad “nut”.

- nano 0/nut

```

18 dimensions      [0 2 -1 0 0 0 0];
19
20 internalField    uniform 0.1;
21
22 boundaryField
23 {
24     walls
25     {
26         type      nutkwallFunction;
27         value     uniform 0;
28     }
29     inlet
30     {
31         type      fixedValue;
32         value     $internalField;
33     }
34     outlet
35     {
36         type      zeroGradient;
37     }
38 }

```

Archivo nut 4.57

9. Presión a la salida.

- nano 0/p

```

17 dimensions      [0 2 -2 0 0 0 0];
18
19 internalField    uniform 100000;
20
21 boundaryField
22 {
23     walls
24     {
25         type      zeroGradient;
26     }
27     inlet
28     {
29         type      zeroGradient;
30     }
31     outlet
32     {
33         type      fixedValue;
34         value     $internalField;
35     }
36 }

```

Archivo p 4.58

10. Ahora cambiamos la presión característica del “solver”.

- nano 0/p_rgh

```

17 dimensions      [0 2 -2 0 0 0 0];
18
19 internalField    uniform 100000;
20
21 boundaryField
22 {
23     walls
24     {
25         type      zeroGradient;
26     }
27     inlet
28     {
29         type      zeroGradient;
30     }
31     outlet
32     {
33         type      fixedValue;
34         value     $internalField;
35     }
36 }

```

Archivo p_rgh 4.59

Establecemos los mismos valores.

11. Para la temperatura cambiamos en la carpeta de archivos el nombre del archivo de la temperatura por “T”. Después de este cambio realizamos el cambio de temperatura a la entrada y fijamos el gradiente de temperatura en la pared.

- nano 0/T

```

16
17 dimensions      [0 0 0 1 0 0 0];
18
19 internalField    uniform 300;
20
21 boundaryField
22 {
23     walls
24     {
25         type      fixedGradient;
26         gradient  uniform 118.68;
27     }
28     inlet
29     {
30         type      fixedValue;
31         value     uniform 298;
32     }
33     outlet
34     {
35         type      zeroGradient;
36     }

```

Archivo T 4.60

12. Para la velocidad usaremos primero la velocidad de 2m/s que establecemos en la dirección “z” de la malla.

- nano 0/U

```
17 dimensions      [0 1 -1 0 0 0 0];
18
19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     walls
24     {
25         type      noslip;
26     }
27     inlet
28     {
29         type      fixedValue;
30         value     uniform (0 0 2);
31     }
32     outlet
33     {
34         type      inletOutlet;
35         inletValue uniform (0 0 0);
36         value     uniform (0 0 0);
37     }
38 }
```

Archivo U 4.61

13. Entramos en “controlDict” para cambiar el tiempo que queremos simular nuestro caso y para decidir en qué intervalos debe hacerlo.

- cd system
- nano controlDict

Realizamos los siguientes cambios:

```
18 application      buoyantBoussinesqPimpleFoam;
19
20 startFrom        startTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          7;
27
28 deltaT           0.004;
29
30 writeControl     timestep;
31
32 writeInterval    20;
33
34 purgeWrite       0;
35
36 writeFormat      ascii;
37
38 writePrecision   6;
39
40 writeCompression off;
41
42 timeFormat       general;
43
44 timePrecision    6;
45
46 runtimeModifiable true;
47
48 adjustTimestep   yes;
49
50 maxCo            1.0;
```

Archivo controlDict 4.62

Hemos establecido como tiempo final 7 segundos en “endTime”, y un “deltaT” de 0.004. En el caso de “writeInterval” que es donde tendremos que establecer cada cuántos intervalos queremos que nos guarde las iteraciones en nuestro equipo, cogeremos uno mayor a 20 para que no tarde un tiempo excesivo en calcularlo, ya que realmente el que nos interesa es el tiempo último ya que estamos trabajando en un caso estacionario. Cogemos 200.

14. Para realizar la simulación, después de crear la malla con “blockMesh”, deberemos introducir el siguiente comando:

- buoyantBoussinesqPimpleFoam

Se calcularán todos los intervalos y ya podremos ver la simulación en “paraView”.

CAPÍTULO 5. RESULTADOS DEL ANÁLISIS DEL PROBLEMA.

En este capítulo se presentan los resultados del análisis del problema en base a las simulaciones numéricas y su comparación con la solución teórica del problema.

Para el análisis de cada caso cogeremos el último tiempo de la simulación, ya que para agilizar el proceso hemos usado el ordenador “Prometeo” de SAIT-UPCT y solo hemos exportado dicho tiempo. Esto no afecta a nuestro estudio ya que se trata de un caso estacionario.

5.1- Preparación del caso para simulación del problema de transmisión de calor por convección en el interior de un tubo.

La parte que nos permitirá estudiar cómo evoluciona el fluido dentro del tubo viene dada por la correcta imposición de las condiciones de contorno.

En nuestro caso impondremos:

1. Una temperatura a la entrada de 298 K. En la salida, impondremos que el flujo está totalmente desarrollado.

En las paredes impondremos un gradiente de temperatura en función del radio del tubo que calculamos a partir de la potencia del autotransformador, que transfiere a la malla encargada de calentar el tubo de acero. De ésta cogeremos un 75% de su valor nominal. Simularemos varios casos de prueba con distintos gradientes de temperatura. El gradiente se expresa como:

$$\frac{dT}{dr} = \frac{\text{Potencia eléctrica}}{k \cdot \pi \cdot D \cdot L} = 118.68^{\circ}\text{C}/\text{m} \quad (5.1)$$

En un primer caso testaremos un gradiente “pequeño”, luego otro “medio” y otro “grande”. También calcularemos con la velocidad teórica y un gradiente térmico donde se aproveche el 95% de la potencia del transformador:

$$\frac{dT}{dr} = \frac{\text{Potencia eléctrica}}{k \cdot \pi \cdot D \cdot L} = 150.32^{\circ}\text{C}/\text{m} \quad (5.2)$$

Donde k es la conductividad térmica. “D” es el diámetro del tubo en metros y L la longitud también en metros.

Lo que queremos realmente es obtener un perfil de temperaturas en el aire, por tanto, cogeremos como conductividad la del aire: 0.0262 W/m·K y optaremos por el 75% del valor nominal de la potencia. Con estas condiciones, se obtiene:

$$\frac{dT}{dr} = \frac{\text{Potencia eléctrica}}{k \cdot \pi \cdot D \cdot L} = \frac{900}{0.0262 \cdot \pi \cdot 0.07315 \cdot 1.5} = 99651.94^{\circ}\text{K}/\text{m} \quad (5.3)$$

Los casos anteriores los usamos para comprobar que el programa funciona correctamente y que hemos metido las condiciones de contorno correctamente, es decir, que el programa nos da resultados parecidos a los que nos aportan los cálculos teóricos, realizaremos dichas comprobaciones a la hora de valorar los resultados.

2. Para la presión estableceremos la presión atmosférica en la salida para nuestro caso de interés que posee el mayor gradiente térmico, es decir, 101325 Pa. Para los tres casos anteriores dejamos la presión establecida en el tutorial.

3. Impondremos a su vez una velocidad a la entrada del tubo, valoraremos el caso variando esta propiedad. Primero utilizaremos una velocidad de 2 m/s para la primera simulación (que corresponde a un caudal de $0.0084 \text{ m}^3/\text{s}$), en segundo lugar, calcularemos teóricamente la velocidad del tubo en función de la sección y compararemos los casos.

Anteriormente especificamos que la velocidad la calcularíamos experimentalmente con la diferencia de presiones a la entrada para compararla con la teórica. Ahora tomando un valor mínimo de funcionamiento del ventilador tomando su caudal y calcularemos esta velocidad teórica.

El caudal en una sección circular viene definido por $Q = V_m \cdot A = V_m \cdot \pi \cdot r^2 \left[\frac{\text{m}^3}{\text{s}} \right]$

Donde V_m es la velocidad media del conducto, y la que nos interesa hallar.

Despejando obtendremos la siguiente relación:

$$V_m = \frac{Q}{\pi \cdot r^2} \text{ [m/s]} \quad (5.4)$$

Observando las características técnicas del ventilador observamos que el caudal máximo que puede producir es de $475 \text{ m}^3/\text{h}$ si tomamos el valor mínimo de éste observando las curvas características tendremos aproximadamente $100 \text{ m}^3/\text{h}$ que pasando a unidades del sistema internacional se nos quedará en $Q = 0.028 \text{ m}^3/\text{s}$.

Introduciendo valores tendremos un valor máximo y mínimo de:

$$V_{m_máximo} = \frac{0.1319}{\pi \cdot 0.036575^2} = 23.56 \text{ m/s} \quad (5.5)$$

$$V_{m_mínimo} = \frac{0.028}{\pi \cdot 0.036575^2} = 6.66 \text{ m/s} \quad (5.6)$$

Nos quedaremos con el valor mínimo, de manera que podamos obtener el mayor salto de temperaturas del aire entre la entrada y la salida del tubo para una potencia dada.

5.2- Análisis de resultados del caso velocidad de entrada 2 m/s con 75% de la potencia y $dT/dr=118.68\text{ }^{\circ}\text{C/m}$.

Realizaremos un análisis visual de las distintas propiedades en la malla, haciendo un corte del cilindro en dirección normal al eje y. Nos permitirá ver la variación de las propiedades dentro del tubo.

1. Comenzamos con la presión:



Presión caso 2ms 5.1

Podemos ver que mantiene una presión constante a lo largo del tubo excepto al principio que podemos observar unos errores residuales en la primera columna, pero que para nuestro caso no es relevante, ya que no afecta a los resultados generales de nuestro caso.

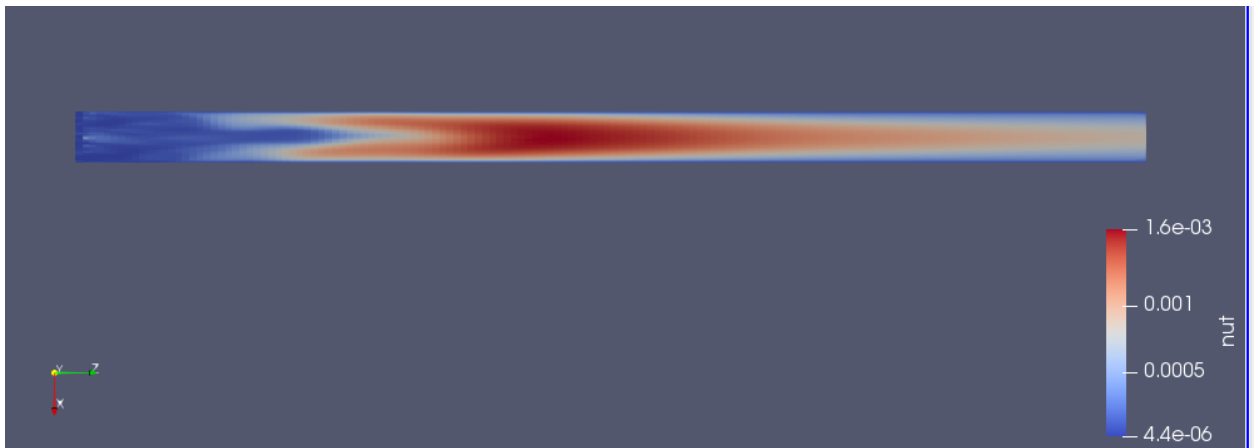
2. Con la velocidad pasa algo parecido:



Velocidad caso 2ms 5.2

También presenta pequeñas fluctuaciones al principio debido a las condiciones de contorno impuestas.

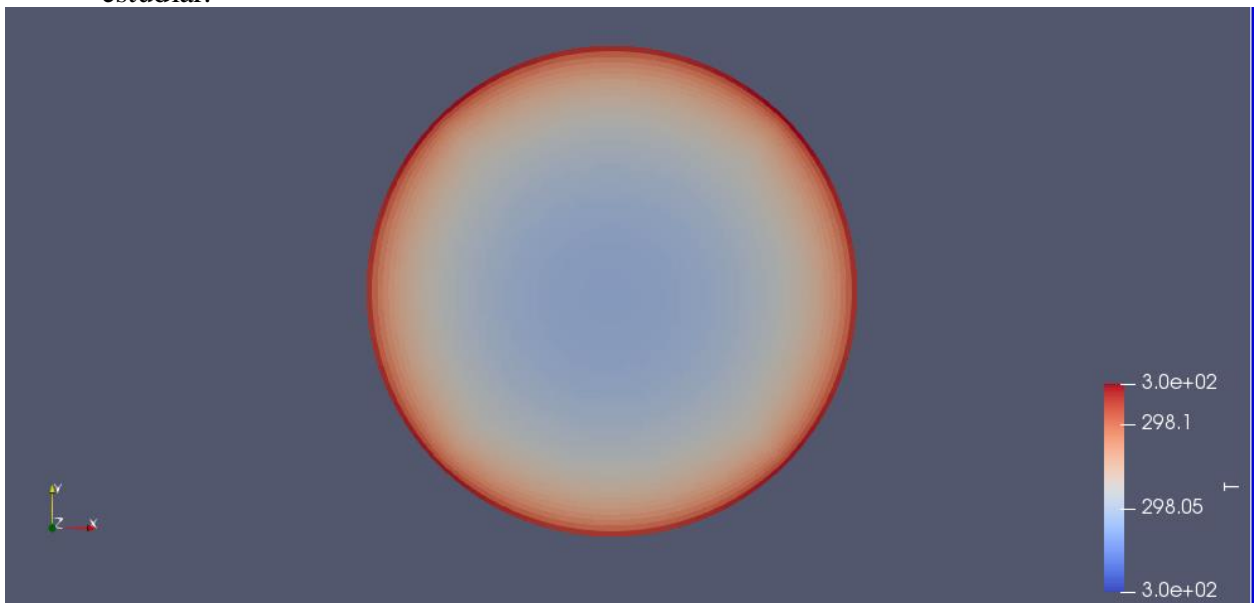
3. Pasamos a la viscosidad del fluido:



Nut caso 2ms 5.3

Se puede apreciar un aumento de la viscosidad turbulenta en el centro del tubo y que después se va disolviendo de nuevo hasta la salida, esto se puede tratar de que el régimen turbulento se estabiliza.

4. Como última propiedad valoraremos la temperatura, la principal variable que nos interesa estudiar:



T sección salida caso 2ms 5.4

Se puede ver el gradiente de temperaturas que va calentando desde el exterior en la sección de salida del tubo.

En el corte transversal podemos ver:



T caso 2ms 5.5

Vemos como el fluido crea una capa límite térmica al final del tubo, el espesor de ésta lo calcularemos posteriormente extrayendo los datos en esa última sección final. Con este proceso también calcularemos la temperatura media del fluido a la salida.

5.3- Análisis de resultados del caso velocidad de entrada 6.66 m/s con 75% de la potencia y $dT/dr=118.68 \text{ }^\circ\text{C/m}$.

Procedemos a ver las mismas propiedades que en el caso anterior:

1. Presión:



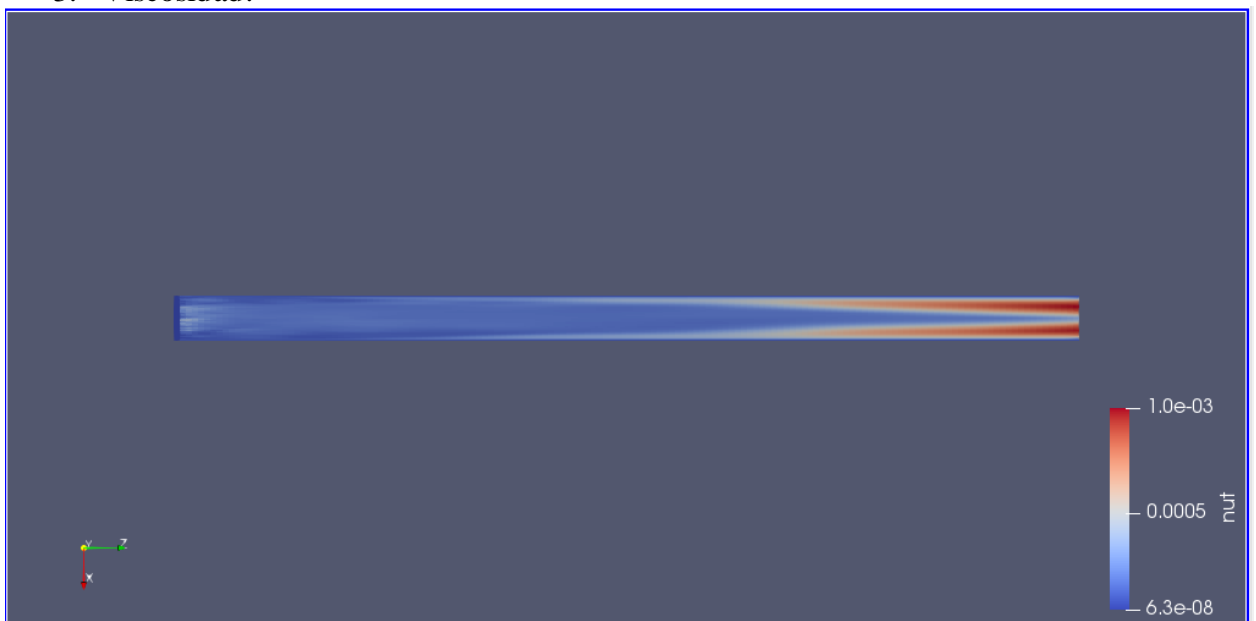
p caso 6.66ms 5.6

2. Velocidad:



U caso 6.66ms 5.7

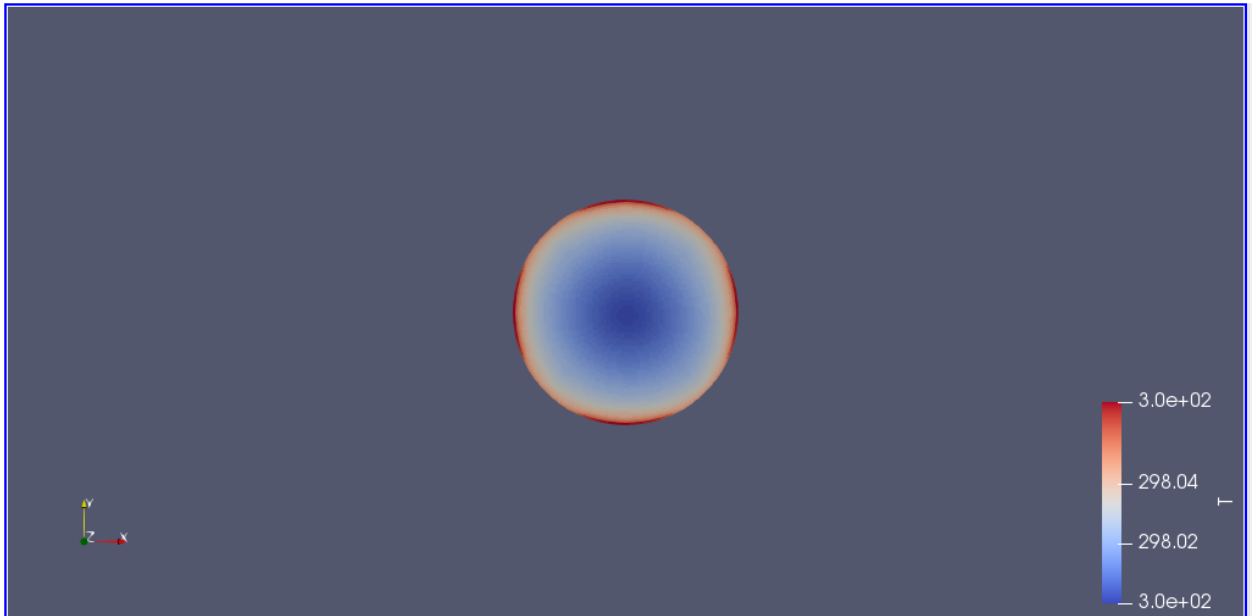
3. Viscosidad:



Nut caso 6.66ms 5.8

Apreciamos un cambio en esta propiedad respecto al caso con 2 m/s, ahora la viscosidad turbulenta aumenta en el final del tubo, al haber aumentado la velocidad del flujo tiene sentido que la turbulencia apareciera después.

4. Temperatura:



T sección salida caso 6.66 5.9

Se ve un gradiente menos uniforme, esto puede ser debido al haber aumentado la velocidad del fluido y que no de tiempo suficiente a calentarse igual que en el caso anterior, o quizás a la no uniformidad de la malla que genera un gradiente circunferencial de temperatura para el caso en que existan gradientes radiales de temperatura altos. Si fuera esto último, se podría corregir con un refinado de malla.



T corte transversal caso 6.66ms 5.10

5.4- Análisis de resultados del caso velocidad de entrada 6.66 m/s con 95% de la potencia y $dT/dr=118.68\text{ }^{\circ}\text{C/m}$.

Al tener una potencia mínima del ventilador, hemos aumentado la velocidad del fluido, algo que no nos interesa del todo en la instalación experimental. Cuanto menor sea la velocidad del fluido más se podrá calentar y podremos valorar como cambia dicha temperatura, pero por las limitaciones que tenemos en cuanto a la velocidad mínima del ventilador podemos probar a aumentar el calor de las paredes, así estudiamos el mismo caso, pero con un gradiente de temperatura mayor.

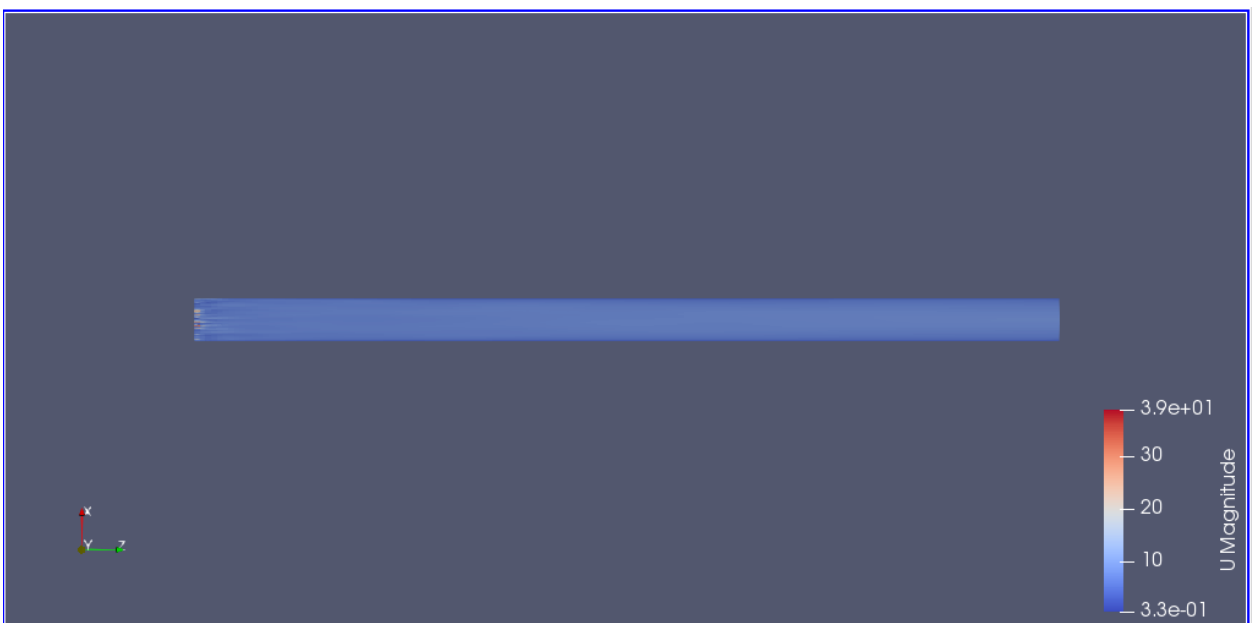
Procedemos igualmente a visualizar las distintas propiedades de este caso:

1. Presión:



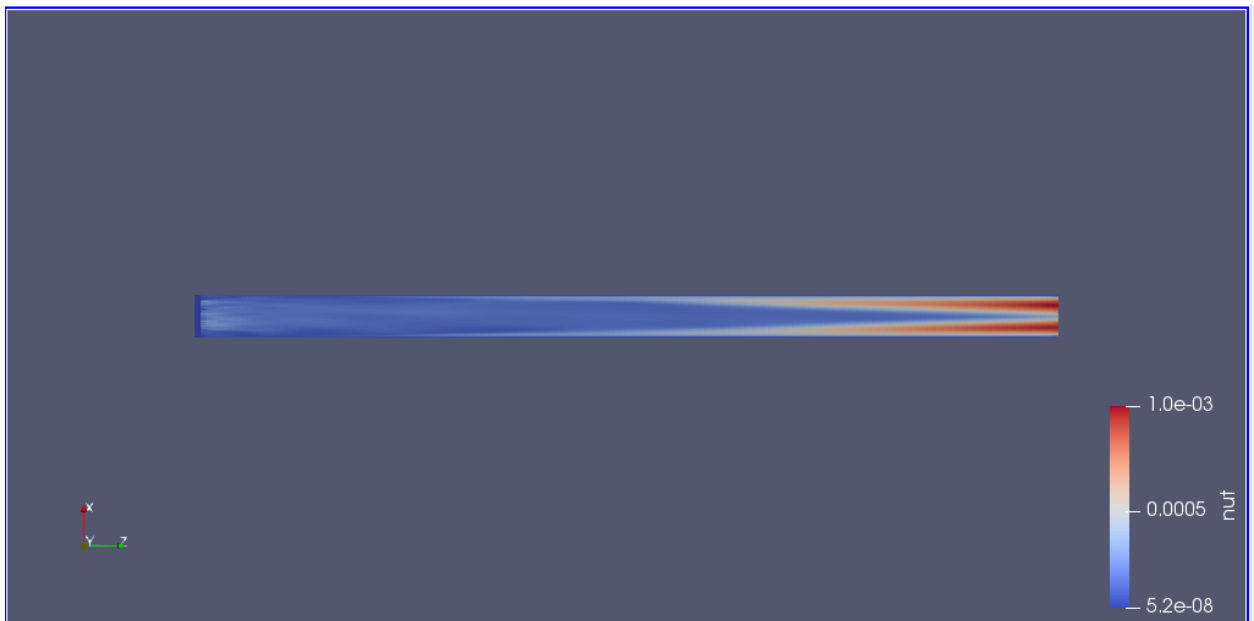
P caso con 95% 5.11

2. Velocidad:



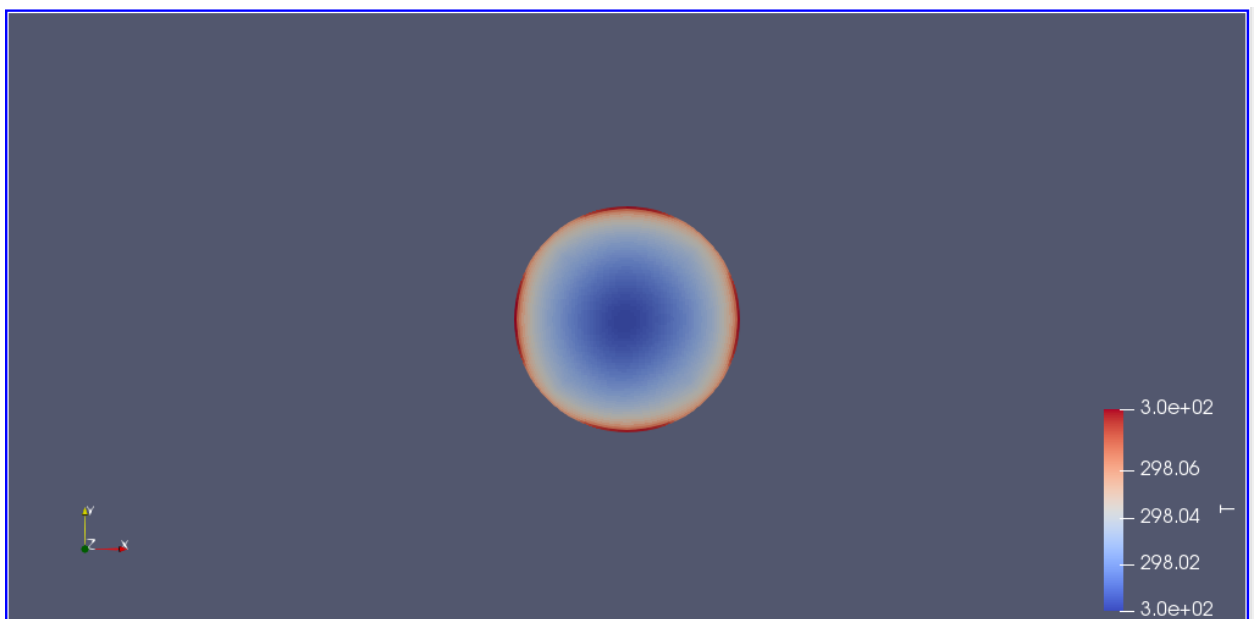
U caso con 95% 5.12

3. Viscosidad turbulenta.



Nut caso con 95% 5.13

4. Temperatura:



T sección salida caso 95% 5.14

Y el corte transversal:



T corte transversal caso 95% 5.15

A priori el caso con 95% de la potencia del transformador y el caso con la potencia a 75% de la potencia parecen iguales, pero después comprobaremos si numéricamente hay una diferencia apreciable o no.

5.5- Análisis de resultados del caso con velocidad teórica de 6.66 m/s con 75% de la potencia y $dT/dr=99651.94 \text{ }^\circ\text{C/m}$.

Al salir un gradiente de temperatura mucho más alto en nuestro caso por haber usado el coeficiente de conductividad térmica del aire como hemos explicado al poner las condiciones de contorno de cada caso, lo que esperamos es que el fluido se caliente de forma más significativa que los anteriores.

1. Presión:



P caso teórico 5.16

2. Velocidad:



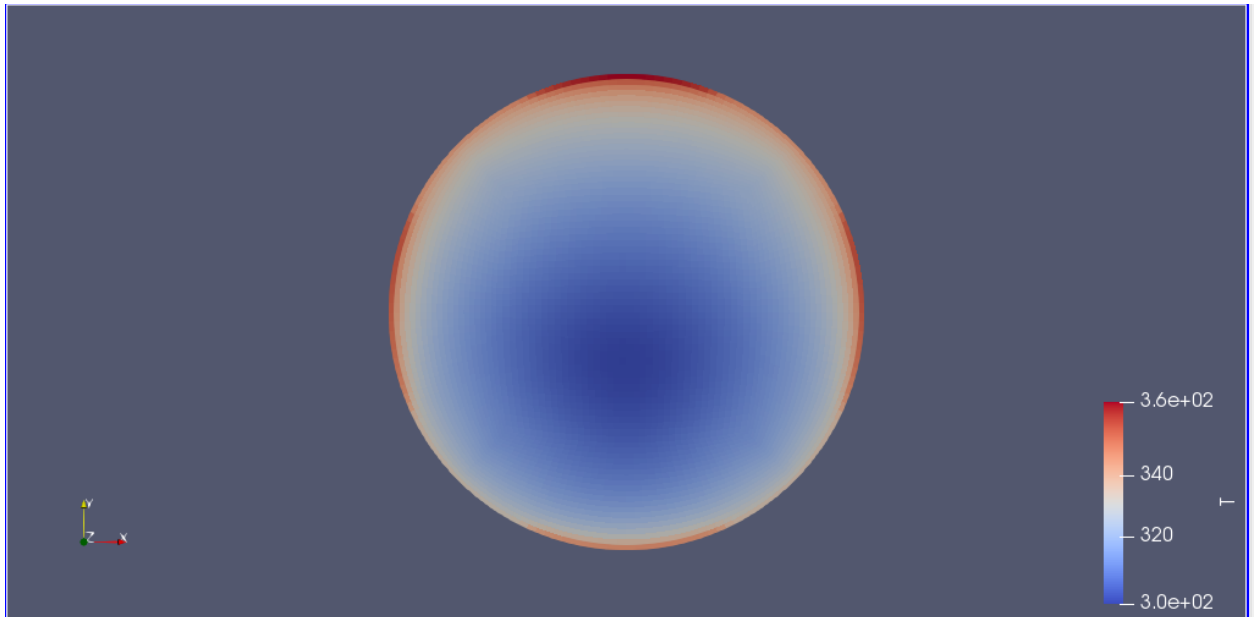
U caso teórico 5.17

3. Viscosidad:



nut caso teórico 5.18

4. Temperatura:



T caso teórico (sección salida) 5.19

En este caso nos volvemos a encontrar con un gradiente que no resulta uniforme, esto nos indica que deberíamos utilizar una malla más refinada y con un mayor número de elementos para que luego se pueda parecer a los datos teóricos.



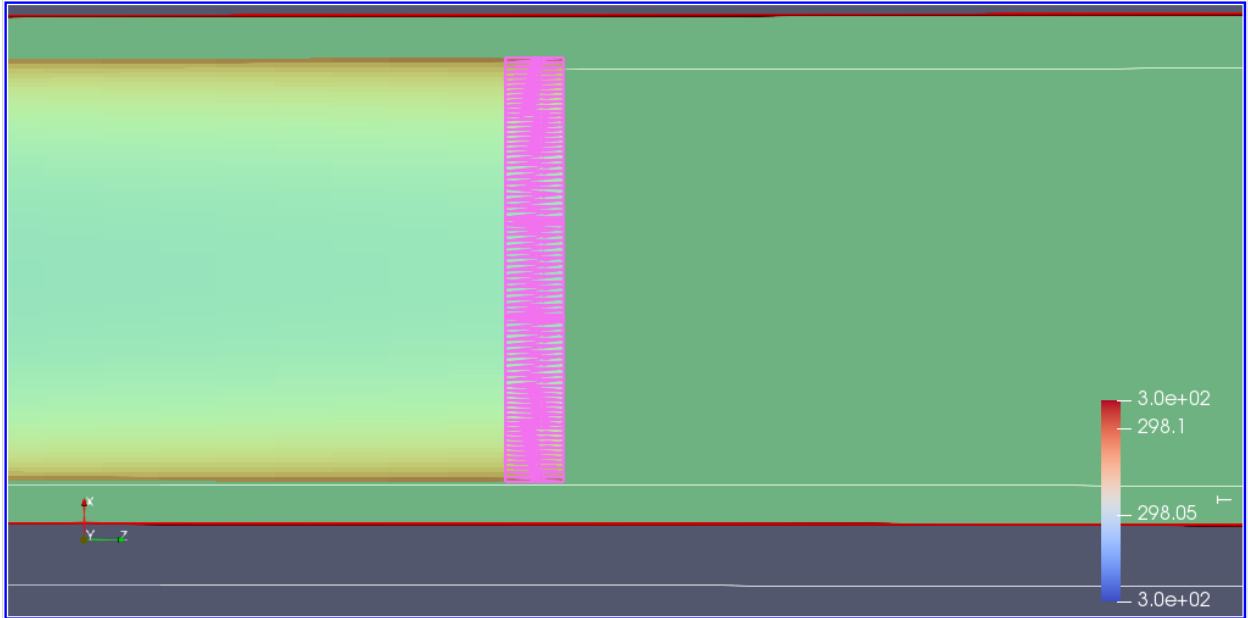
T caso teórico corte transversal 5.20

5.6- Influencia de la velocidad de entrada y del gradiente de temperaturas.

Lo que haremos será estudiar la última columna de celdas del corte transversal que hemos realizado, esto nos permitirá valorar la temperatura media a la salida y calcular de forma aproximada el espesor de la capa límite.

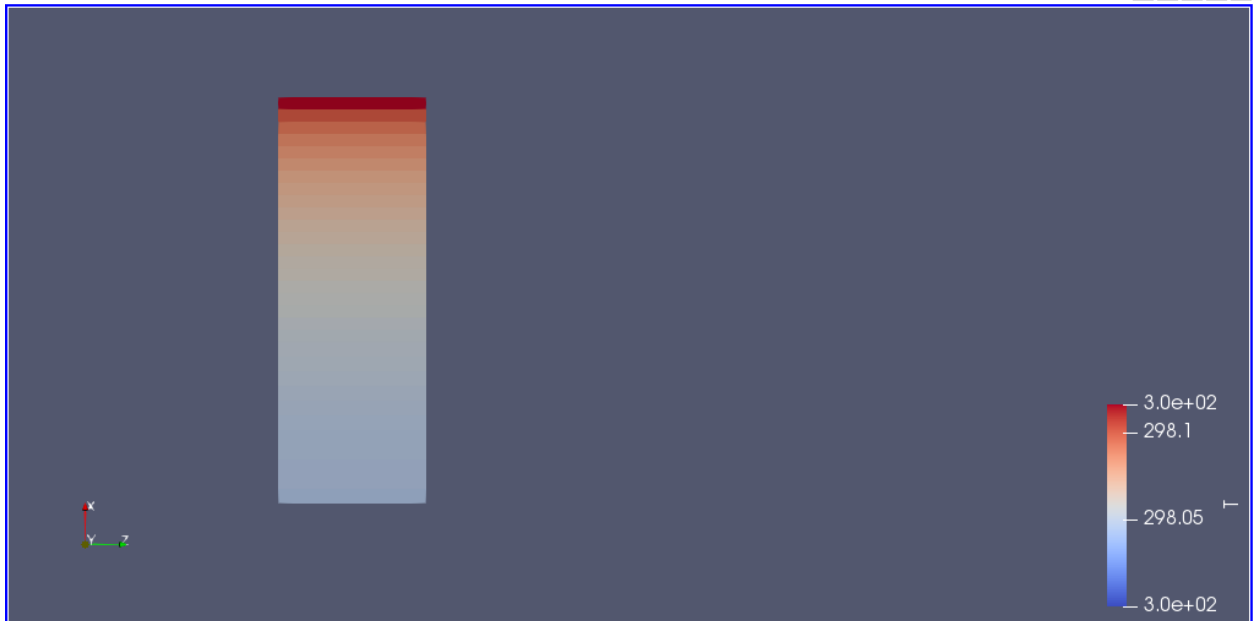
Este proceso lo haremos de la siguiente manera:

1. Seleccionamos, en Paraview, las celdas de la última columna del caso resuelto, con la opción de “SelectCellsOn”:



Selección celdas 5.16

2. Extraemos estos datos como “points” en una carpeta de Excel en la opción de “SaveData” que nos ofrece el propio programa.
3. Para calcular la capa límite haremos una selección menor, desde el borde hasta donde vemos que la temperatura se vuelve de un color estable, esto significará el final de la capa límite térmica. Después estos datos los exportaremos y calcularemos la capa límite de forma en que se acabará cuando se aprecie que se estabilizan los valores de temperatura.



Selección capa límite 5.17

- Estos datos también los extraeremos para calcular ese espesor de la capa límite.

Después de calcular la temperatura media y el espesor de la capa límite térmica en cada caso los compararemos, al igual que haremos con el coeficiente de transmisión de calor de cada caso.

5.6.1- Temperatura media a la salida.

- Caso de $U=2\text{m/s}$ con 75% de la potencia y $dT/dr=118.68^\circ\text{C/m}$.
Presenta una temperatura media a la salida de 298.057 K.
Con un mínimo de 298.04 K y un máximo de 298.11 K.
- Caso de $U=6.66\text{m/s}$ con 75% de la potencia y $dT/dr=118.68^\circ\text{C/m}$.
Presenta una temperatura media a la salida de 298.018 K.
Con un mínimo de 298 K y un máximo de 298.07 K.
- Caso de $U=6.66\text{m/s}$ con 95% de la potencia y $dT/dr=118.68^\circ\text{C/m}$.
Presenta una temperatura media a la salida de 298.024 K.
Con un mínimo de 298 K y un máximo de 298.09 K.
- Caso teórico $U=6.66\text{m/s}$ con 75% de la potencia y $dT/dr=99651.94\text{ K/m}$.
Presenta una temperatura media a la salida de 314.563 K.
Con un mínimo de 301.15 K y un máximo de 359.39 K.

5.6.2-Espesor capa límite térmica.

- Caso de $U=2\text{m/s}$ con 75% de la potencia y $dT/dr=118.68^\circ\text{C/m}$.
Presenta un espesor máximo de capa límite de 14.084 mm. Es un dato alto, pero tiene lógica a ser el caso que más se calienta de los tres primeros.
- Caso de $U=6.66\text{m/s}$ con 75% de la potencia y $dT/dr=118.68^\circ\text{C/m}$.
Presenta un espesor de capa límite de 9.945 mm.

3. Caso de $U=6.66\text{m/s}$ con 95% de la potencia y $dT/dr=118.68^\circ\text{C/m}$.
Presenta un espesor de capa límite de 13.26 mm. Al aumentar la potencia utilizada para el gradiente de temperaturas hemos conseguido aumentar la temperatura a la salida y el espesor de la capa límite térmica, saliendo datos similares al primer caso.
4. Caso teórico $U=6.66\text{m/s}$ con 75% de la potencia y $dT/dr=99651.94\text{ k/m}$.
Presenta un espesor de capa límite de 15.746 mm. Con un gradiente de temperaturas mayor el espesor de dicha capa aumenta.

5.6.3- Temperatura teórica media a la salida.

Si nos fijamos en la relación de ecuaciones (1.4) que encontramos en la introducción teórica, podemos comparar la temperatura media que el programa nos da con la teórica:

$$T_{sal} = T_{ent} + \frac{\dot{q}A}{\dot{m}c_p} \quad (5.7)$$

1. Caso de $U=2\text{m/s}$.

Comprobamos con este caso que el programa nos da un salto de temperaturas similar al teórico.

Si introducimos valores:

$$T_{sal} = 298 + \frac{1.07293446}{0.033159748 * 1000} = 298.03\text{ K} \quad (5.8)$$

El factor de Q lo hemos calculado como:

$$Q = KA \frac{dT}{dr} = 1.07293446W \quad (5.9)$$

Esta es en realidad la potencia que hemos medido al perfil del aire, nos sirve para comprobar que la temperatura media que nos da el programa es del mismo orden que la teórica y que hemos introducido correctamente los valores y las condiciones de contorno necesarias.

2. Caso teórico para una potencia calorífica de 900W.

$$T_{sal} = 298 + \frac{900}{0.033159748 * 1000} = 325.14\text{ K} \quad (5.10)$$

Es decir, presenta un salto de temperaturas de 27.14 K.

El resultado de las simulaciones nos ha dado un salto de temperatura similar aunque un poco menor, pero que sigue siendo del mismo orden de magnitud que el teórico.

Si en vez de hacer un corte transversal en y lo hacemos para la sección final en z nos dará un salto de temperaturas algo mayor, al promediar una temperatura de 318.84 K. Se acerca al valor teórico, sin embargo, sigue habiendo una diferencia notable que se puede deber a la irregularidad del gradiente en la malla.

5.6.4- Coeficiente de transmisión de calor y número de Nusselt.

Para calcular el coeficiente de transmisión de calor en nuestro caso de flujo de calor constante, sabemos que habrá una longitud de entrada térmica que nos dicta a partir de qué coordenada x este coeficiente se vuelve constante. Esta longitud la calcularemos con la ecuación (1.25) establecida en el capítulo introductorio para un régimen turbulento.

$$L_{t,turbulento} = 10D = 10 \times 73.15 = 731.5\text{mm} \quad (5.11)$$

Esto nos indica que la longitud a la que el coeficiente de calor dejará de variar será aproximadamente a la mitad del tubo. El coeficiente lo calcularemos con la siguiente relación:

$$h = \frac{k \frac{dT}{dr}}{(T_s - T_m)} \quad (5.12)$$

Utilizamos la conductividad térmica del aire, el gradiente de temperaturas que hemos impuesto y para la temperatura de la pared tomando la máxima temperatura que nos dé el programa en la sección de celdas que elijamos más el gradiente de temperaturas por la mitad de la altura de la propia celda, y la temperatura media de esas celdas.

Para ver la variación de este término elegiremos 3 secciones distintas antes de la longitud térmica de entrada calculada.

1. Sección 1.

A una distancia de la entrada de 100 mm, extrayendo datos de la simulación, obtenemos como temperaturas: $T_s = 319.68 + 99651.94 * 0.0004145 = 360.99$ K y

$$T_m = 298.89 \text{ K}$$

Esto nos dará:

$$h_1 = \frac{k \frac{dT}{dr}}{(T_s - T_m)} = \frac{0.0262 * 99651.94}{(360.99 - 298.89)} = 42.11 \text{ W/m}^2\text{K} \quad (5.13)$$

Y el número de Nusselt:

$$Nu_1 = \frac{h \cdot x}{k_f} = \frac{h \cdot D}{k} = \frac{42.11 * 0.07315}{0.0262} = 117.57 \quad (5.14)$$

2. Sección 2.

A una distancia de la entrada de 320 mm obtenemos como temperaturas: $T_s = 335.69 + 99651.94 * 0.0004145 = 376.99$ K y $T_m = 300.66$ K.

Esto nos dará:

$$h_2 = \frac{k \frac{dT}{dr}}{(T_s - T_m)} = \frac{0.0262 * 99651.94}{(376.99 - 300.66)} = 34.2 \text{ W/m}^2\text{K} \quad (5.15)$$

Y el número de Nusselt:

$$Nu_2 = \frac{h \cdot x}{k_f} = \frac{h \cdot D}{k} = \frac{34.2 * 0.07315}{0.0262} = 95.49 \quad (5.16)$$

3. Sección 3.

A una distancia de la entrada de 480mm obtenemos como temperaturas: $T_s = 340.58 + 99651.94 * 0.0004145 = 381.89$ K y $T_m = 301.9$ K.

Esto nos dará:

$$h_3 = \frac{k \frac{dT}{dr}}{(T_s - T_m)} = \frac{0.0262 * 99651.94}{(381.89 - 301.9)} = 32.64 \text{ W/m}^2\text{K} \quad (5.17)$$

Y el número de Nusselt:

$$Nu_3 = \frac{h \cdot x}{k_f} = \frac{h \cdot D}{k} = \frac{32.64 \cdot 0.07315}{0.0262} = 91.13 \tag{5.18}$$

4. Sección a la salida.

A una distancia de la entrada de 1500mm, dentro de la región donde el valor del coeficiente se estabiliza, obtenemos como temperaturas: $T_s = 354.39 + 99651.94 \cdot 0.0004145 = 395.7$ K y $T_m = 312.7$ K.

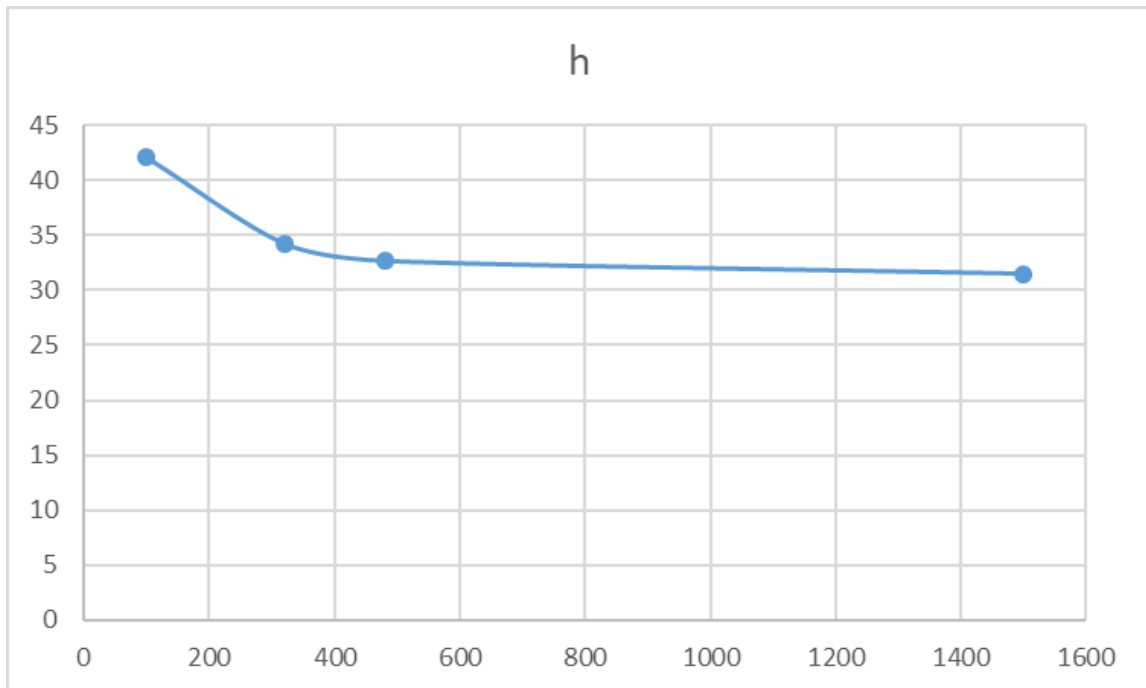
Esto nos dará:

$$h_4 = \frac{k \frac{dT}{dr}}{(T_s - T_m)} = \frac{0.0262 \cdot 99651.94}{(395.7 - 312.7)} = 31.46 \text{ W/m}^2\text{K} \tag{5.19}$$

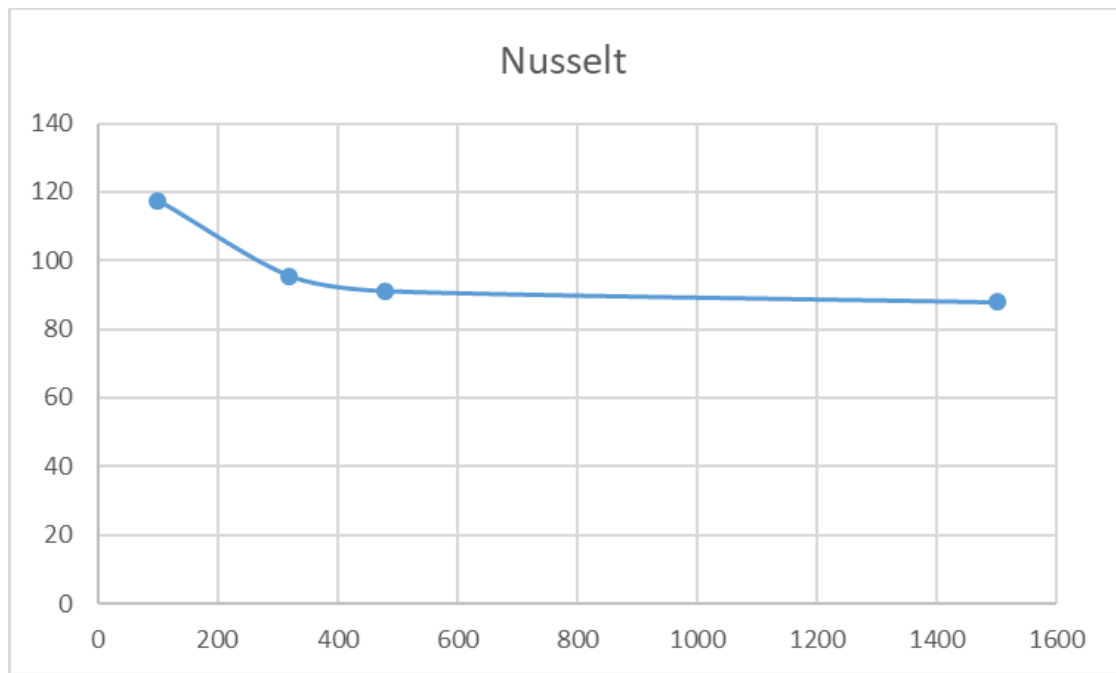
Y el número de Nusselt:

$$Nu_4 = \frac{h \cdot x}{k_f} = \frac{h \cdot D}{k} = \frac{31.46 \cdot 0.07315}{0.0262} = 87.83 \tag{5.20}$$

Podemos representar estos valores frente a las diferentes longitudes a lo largo del tubo para ver cómo se estabilizan.



Variación de h 5.18



Variación de Nu 5.19

Se puede estimar el valor promedio a partir de:

$$h = \frac{1}{L} \int_0^L h(x) dx \tag{5.21}$$

Pero podemos aproximarlo con un conjunto discreto de valores, a partir de las secciones que hemos seleccionado. Así tendremos $h=35.1 \text{ W/m}^2\text{K}$ y $Nu=94.61$ como valores promedio.

Comparamos estos resultados con las correlaciones empíricas:

La ecuación de Petukhov y Polyakov para flujo hidrodinámicamente desarrollado en condiciones de $q''=cte$ es:

$$Nu_x = 4,36 + 1,31Gz^{1/3} \exp\left(-\frac{13}{\sqrt{Gz}}\right) \tag{5.22}$$

Obteniendo las propiedades del aire a 298°k tenemos:

$$Re = \frac{\rho U D}{\mu} = \frac{1.2 \cdot 6.6 \cdot 0.07315}{18.41 \cdot 10^{-6}} = 31469.2 ; Pr = 0.7125 ; Gz = \frac{D}{L} Re Pr = 1093.44 \tag{5.23}$$

De esta forma nos dará un valor de Nusselt de 13.47. Nos da un valor significativamente distinto. Con el cálculo a partir de h estamos teniendo en cuenta el gradiente de temperaturas introducido y en esta correlación no.

Flujo turbulento desarrollado en tubos suaves.

- Relación de Colburn

$$f = 0.184 Re^{-0.2} \quad Nu = 0.023 \cdot Re^{0.8} \cdot Pr^{1/3} \quad \begin{pmatrix} 0.7 \leq Pr \leq 160 \\ Re > 10000 \end{pmatrix} \tag{5.24}$$

Propiedades determinadas a $T_m=0.5(T_{ent}+T_{sal})$

En nuestro caso $T_m=308.42^\circ\text{K}$.

$$Re = 29407.23 ; f = 0.024 ; Pr = 0.711 \quad (5.25)$$

Nos da un número de Nusselt de 77.11.

- Relación de Dittus-Boelter.

$$Nu = 0.023 \cdot Re^{0.8} \cdot Pr^n \quad \begin{pmatrix} 0.7 \leq Pr \leq 160 \\ Re > 10000 \end{pmatrix} \quad (5.26)$$

$n=0.4$ cuando se calienta, $n=0.3$ cuando se enfría. Propiedades se determinadas a $T_m=0.5(T_{ent}+T_{sal})$

En este caso el Nusselt resultante es 75.38.

Flujo turbulento desarrollado en tubos suaves.

- Relación de Petukov.

$$Nu = \frac{(f/8)Re Pr}{1.07 + 12.7(f/8)^{0.5}(Pr^{2/3}-1)} \quad \begin{pmatrix} 0.5 \leq Pr \leq 2000 \\ 10^4 < Re < 5 \cdot 10^6 \end{pmatrix} \quad (5.27)$$

Propiedades se determinadas a $T_m=0.5(T_{ent}+T_{sal})$

En este caso el Nusselt resultante es 67.55.

- Relación de Gnielinski.

$$Nu = \frac{(f/8)(Re-1000)Pr}{1 + 12.7(f/8)^{0.5}(Pr^{2/3}-1)} \quad \begin{pmatrix} 0.5 \leq Pr \leq 2000 \\ 3 \cdot 10^3 < Re < 5 \cdot 10^6 \end{pmatrix} \quad (5.28)$$

Propiedades se determinadas a $T_m=0.5(T_{ent}+T_{sal})$

En este caso el Nusselt resultante es 70.58.

Nos dan valores menores que Nusselt promedio en el tubo y Nusselt en la sección de la región estable. Pero nos encontramos con valores de un mismo orden de magnitud.

CAPÍTULO 6. CONCLUSIÓN DEL TRABAJO.

En este trabajo se ha realizado un prediseño de una instalación experimental, basada en el calentamiento del aire impulsado por un ventilador mediante un tubo de acero, que a su vez es calentado desde el exterior. La instalación diseñada permitirá el estudio de problemas de transferencia de calor por convección en tubos.

Además, se ha simulado numéricamente con OpenFoam la instalación diseñada para evaluar las condiciones que permitirá analizar en el laboratorio. En cuanto a resultados, las simulaciones muestran que cuando el ventilador opera a $0.028 \text{ m}^3/\text{s}$ y el transformador opera a 900 W , se obtiene una velocidad de 6.66 m/s , un salto de temperatura media entre la entrada y la salida del tubo de 16.56 K y un espesor de capa límite térmica de 15.746 mm y un coeficiente de transmisión de calor por convección h de $35.1 \text{ W/m}^2 \text{ K}$ como valor promedio y de $31.46 \text{ W/m}^2 \text{ K}$ en la región dónde se estabiliza esta variable.

Desde el punto de vista personal, el trabajo me ha permitido profundizar en el conocimiento tanto de transmisión del calor como del manejo de un programa de simulaciones de fluidos como OpenFoam que ha sido totalmente nuevo para mí.

Sería bastante interesante comparar los datos teóricos simulados con los datos de la instalación experimental una vez montada y ver las diferencias en los datos que puedan surgir.

CAPÍTULO 7. BIBLIOGRAFÍA

- 6.1- MADRID GARCÍA, Carmelo Nicolás; NAVARRO ANDREU, José Ramón. *Transmisión del calor*. 1ª edición: Universidad Politécnica de Cartagena, 2012.
- 6.2- 2005 Universidad Nacional Experimental del Táchira. *Fenómenos de transporte*. [http://www.unet.edu.ve/~fenomeno/F_DE_T-153.htm]
- 6.3- Adrián Pedrozo-Acuña y Alec Torres-Freyermuth. *Sobre el uso de las ecuaciones de Navier-Stokes con el promedio de Reynolds en el campo de la ingeniería de costas*. Universidad Nacional Autónoma de México, 2011.
[http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-24222011000200004#:~:text=Las%20ecuaciones%20tipo%20RANS%20se,detalles%20]
- 6.4- Ecuaciones promediadas de Reynolds. [https://es.qwe.wiki/wiki/Reynolds-averaged_Navier%E2%80%93Stokes_equations]
- 6.5- Ecuaciones de Navier-Stokes. Fenómenos turbulentos. [<http://bibing.us.es/proyectos/abreproy/3718/fichero/Parte+I%252FCapitulo+2.pdf>]
- 6.6- Método de volúmenes finitos.
[https://www.u-cursos.cl/ingenieria/2008/2/C171D/1/material_docente/bajar?id_material=200975#:~:text=El%20m%C3%A9todo%20de%20los%20vol%C3%BAmenes,diferencias%20finitas%20y%20elementos%20finitos.&text=De%20esta%20forma%20el%20volumen,los%20vol%C3%BAmenes%20de%20control%20considerados]
- 6.7- Fichero malla cilíndrica. [<https://www.ehsanmadadi.com/cylinder-mesh/>]
- 6.8- Tutorial iniciación con “blueCFD” de la página oficial. [https://www.youtube.com/watch?time_continue=940&v=1Ti0yU1JdTQ&feature=emb_logo]
- 6.9- Tutorial József Nagy. *Introducción al modelo de turbulencia estacionaria. Parte I*. [<https://www.youtube.com/watch?v=IPExwi2Ar-g>]
- 6.10- Tutorial József Nagy. *Introducción al modelo de turbulencia estacionaria. Parte II*. [<https://www.youtube.com/watch?v=CMchl-1d-zI>]
- 6.11- Tutorial József Nagy. *Introducción al modelo de turbulencia estacionaria. Parte III*. [<https://www.youtube.com/watch?v=-46pgYQYER8>]
- 6.12- Tutorial de transferencia de calor. CFD Expert. Parte I. [https://www.youtube.com/watch?v=0Wj_89OXM-c]
- 6.13- Tutorial de transferencia de calor. CFD Expert. Parte II. [<https://www.youtube.com/watch?v=5VMY9MAvzOc>]
- 6.14- Página oficial blueCFD para descargar paquete del programa [<https://bluecfd.github.io/Core/Downloads/>]
- 6.15- Referencia métodos numéricos [repository.uamerica.edu.co]

8. ANEXOS

8.1- Hoja técnica de la guía de posicionamiento (1 unidad).

Marca: Rose Krieger.

Modelo: FNA 1246TA0304.

Precio por unidad: 402,13 euros.

Funktionsprinzip:

Eine Rotationsbewegung der Gewindespindel wird in eine lineare Ausgangsbewegung des Führungsschlittens umgewandelt.

Function:

a rotating movement of the spindle is converted into a linear movement of the guide table.



Especificaciones

Atributo	Valor
Longitud de la Mesa Deslizante	180mm
Anchura de la Mesa Deslizante	120mm
Longitud Global	339.5mm
Longitud del Recorrido	100mm
Valor Nominal de Carga Estática	215No

8.2- Hoja técnica del ventilador (2 unidades).

Marca: SODECA.

Modelo: 512-2M Ventilador centrífugo de media presión.

Precio por unidad: 115 euros.



Ventiladores centrífugos de media presión y simple aspiración con envolvente y turbina en chapa de acero.

- Ventilador:
- Envolvente en chapa de acero
 - Turbina con álabes hacia delante, en chapa de acero galvanizado
 - Modelo CMP 38-2M envolvente en fundición de aluminio



Rodetes equilibrados dinámicamente, con núcleos de gran robustez

- Motor:
- Motores clase F, con rodamientos a bolas, protección IP55, excepto modelos monofásicos protección IP54. Modelo CMP-38 protección IP21
 - Monofásicos 230V.-50Hz., y trifásicos 230/400V.-50Hz. (hasta 5,5CV.) y 400/690V.-50Hz. (potencias superiores a 5,5CV.)
 - Temperatura máxima del aire a transportar: -20°C + 120°C, máximo +100°C modelo CMP-38

Acabado: Anticorrosivo en resina de poliéster, polimerizada a 190°C., previo desengrase alcalino y pretratamiento libre de fosfatos

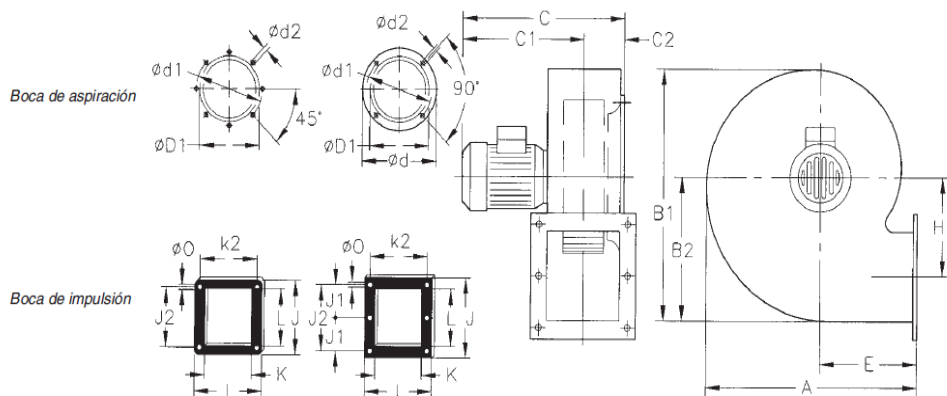
- Bajo demanda:
- Bobinados especiales para diferentes tensiones
 - Ventilador preparado para transportar aire hasta 250°C
 - Ventilador en acero inoxidable.
 - Certificación ATEX Categoría 2 (ver serie CMP/ATEX)

Características técnicas

Modelo	Velocidad (r/min)	Intensidad máxima admisible (A)			Potencia instalada (kW)	Caudal máximo (m ³ /h)	Nivel de presión sonora dB(A)	Peso Aprox. Kg.
		230V	400V	690V				
CMP-512-2T	2850	0,55	0,32	-	0,08	475	62	4,0

Características acústicas

Los valores indicados, se determinan mediante medidas de nivel de presión y potencia sonora en dB(A) obtenidas en campo libre a una distancia

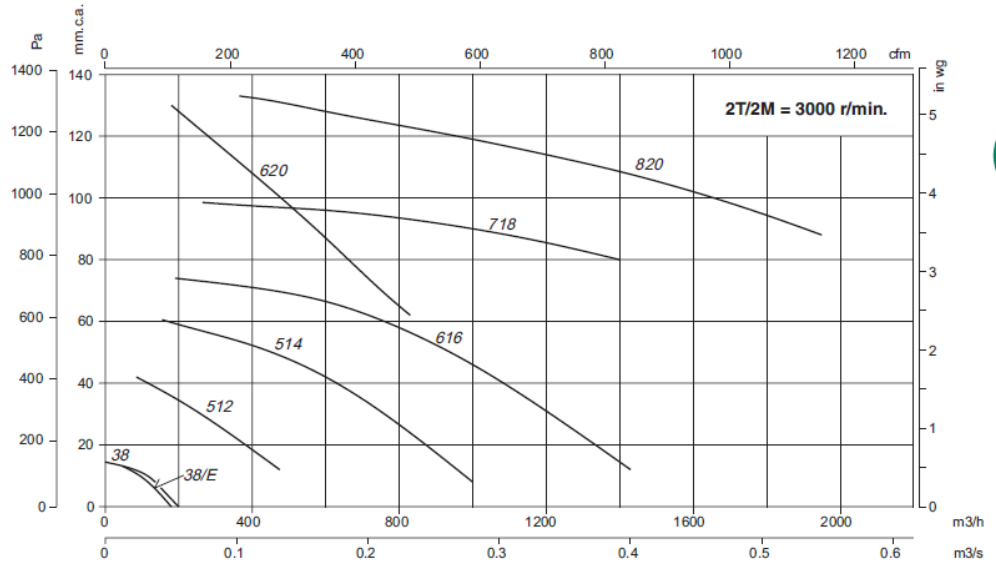


Modelo	A	B1	B2	C	C1	C2	ØD1*	Ød	Ød1	Ød2	E	H1	I	J	J1	J2	K	k2	L	Ø0
CMP-512	182	207	118	250.5	211.5	39	112	140	132	M.4	81	69	106	118	-	105	75	93	86	5.5

Dimensiones (mm)

Curvas Características

Q = Caudal en m³/h y m³/s.
 Pe = Presión estática en mm.c.a. y Pa.

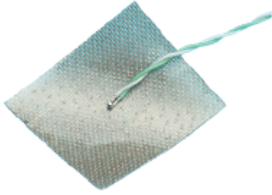


8.3- Hoja técnica termopares (20 unidades).

Marca: TC Direct.

Modelo: Tipo T con conector macho.

Precio por unidad: 10.80 euros.



Adecuado para fijarlo a superficies planas o curvas. Su rango de temperatura es de -50°C a +250°C. Se fabrican con cable termopar aislado con PFA unido a una lámina adhesiva revestida de fibra de vidrio impregnada con PTFE de alta temperatura.

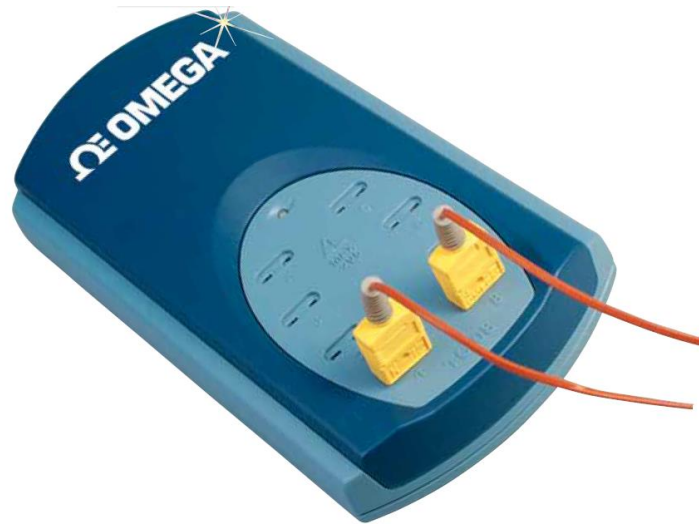
- Disponible en tipo K, T, J ó N
- Tamaño de la lámina: 25mm x 20mm
- Cable de hilo sólido 2 x 0.2mm, par trenzado aislado con PFA
- Longitud del cable: 2 metros
- Rango: -50°C a +250°C

8.4- Hoja técnica registradores de medidas de los termopares (2 unidades).

Marca: Omega.

Modelo: TC 08.

Precio por unidad: 354 euros.



Resolución

Tipo de termopar	Rango total	Resolución de 0,1 °C	Resolución de 0,025 °C
J	-210 a 1200 (-346 a 192 °F)	-210 a 1200 °C	-210 a 1200

Especificaciones

Número de canales: 8
Tiempo de conversión: 100 mseg. (termopar y compensación de unión fría)
Precisión de temperatura: Suma de ±0,2% de lectura y ±0,5 °C
Precisión de voltaje: Suma de ±0,2% de lectura y ±10 µV
Protección contra sobrecarga: ±30V
Máximo común
Voltaje de modo: ±7,5V
Impedancia de entrada: 2 MΩ
Rango de entrada (voltaje): ±70 mV
Resolución: 20-bits
Resolución sin ruido: 16,25-bits
Tipos de termopar: B, E, J, K, N, R, S, T
Conectores de entrada: Termopar miniatura

Conector de salida: Cable USB de 1,8 m (6') incluido
Conexión a PC: USB 1.1
Alimentación: Por puerto USB
Dimensiones: 201 x 104 x 75 mm (7,91 x 4,09 x 1,34")
Software incluido: Registrador y reproductor TC-08
REQUISITOS DEL PC
Procesador: Procesador de clase Pentium o equivalente

Memoria: Mínimo 32 MB
Espacio libre en el disco duro: Mínimo 10 MB
Sistema operativo: Versiones de 32 o 64 bits de WINXP (SP2 o superior)/VISTA/WIN 7
Puertos: Puertos conformes con USB 1.1 o USB 2.0. Debe conectarse directamente al puerto o a un buje USB de alimentación

Para hacer su pedido, visite es.omega.com/tc-08 para consultar precios y detalles

N.º de modelo	Descripción
TC-08	Modulo de adquisición de datos de termopares con 8 canales

*Completo de serie con software para Windows, cable USB y manual del operador en CD.
 Ejemplo de pedido: TC-08 Modulo de adquisición de datos de termopares con 8 canales.*

8.5- Hoja técnica de la manta eléctrica (2 unidades).

Precio por unidad: 228,25 euros.

CALEFACTOR FLEXIBLE DE SILICONA

Dimensiones: 6" (152 mm) x 40" (1016 mm)

Alimentación 120 Vac, Potencia 1.200 W, Carga = 0,77 W/cm²

12" (305 mm) de cable aislado con teflon

Autoadhesivo PSAS de superficie (180°C)

Ref: 060400C1-A001B



8.6- Hoja técnica del sensor de presión diferencial (1 unidad).

Marca: Halstrup-Walcher.

Modelo: PS17.

Precio por unidad: 249,26 euros.



Measurement ranges (also \pm) others available upon request	50/100/200/500 Pa 1/2.5/5/10 kPa
Margin of error (at 22 °C, 0.3 Pa margin of error for the reference)	$\pm 1\%$ of the set value for measurement ranges ≤ 100 Pa plus ± 0.5 Pa
Temperature coefficient span	0.1 % of max. value/K
Temperature coefficient zero point	0.1 % of max. value/K
Max. system pressure / Overload capacity	± 25 kPa: measurement ranges ≤ 250 Pa ± 50 kPa: measurement ranges > 250 Pa
Medium	air, all non-aggressive and non-flammable gases
Sensor response time	25 ms
Ambient temperature	-10 .. 70 °C with display: 0 .. 50 °C
Storage temperature	-10 .. 70 °C with display: -5 .. 55 °C
Calibration temperature	22 °C
Air humidity (medium)	0 .. 80 % RH
Power consumption	< 1 W (optional relay: < 4 W)
Pressure ports	for tubing NW 4 and 6 mm
Protection class	IP67
Weight	approx. 200 g
Certificates	CE

8.7- Hoja técnica del autotransformador (2 unidades).

Marca: Rose Krieger.

Modelo: RS Pro1 Phase 1200VA Variac, 1 Output, 240V.

Precio por unidad: 366,04 euros.

Especificaciones

Atributo	Valor
Tensión nominal primaria	240V
Tensión nominal secundaria	0 → 240 V, 0 → 270 V
Potencia nominal	1.2kVA
Tipo de montaje	Montaje superficial
Rotación del mando	Sentido horario
Número de fases	1
Número de salidas	1
Tipo de terminación	Terminación de cable
Peso	8.8kg
Dimensiones	175 x 175 x 140mm
Corriente nominal primaria	5A
Profundidad	140mm
Longitud:	175mm
Construcción	Bastidor cerrado
Frecuencia de funcionamiento máxima	60Hz
Características de protección	Protección contra sobrecarga
Ancho	175mm
Frecuencia de funcionamiento mínima	50Hz

8.8- Código del problema con 6.66 m/s, potencia al 75% y gradiente de temperaturas de 99651.94 k/m.

Scripts sacados del programa de OpenFoam cuya fuente oficial podemos encontrar en la página web. [<https://www.openfoam.com/>]

1. blockMeshDict.

```

/*-----* C++ *-----*\
|=====|
|\ / Field | OpenFOAM: The Open Source CFD Toolbox |
|\ / Operation | |
|\ / And | Copyright (C) 2016 Ehsan Madadi-Kandjani |
|\ / Manipulation | |
\*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    object blockMeshDict;
}
// ***** //
// General macros to create cylinder mesh
// ***** //

convertToMeters 0.001;

// Inner square side half

// Inner square side curvature

// cylinder radius

// Height of cylinder

// Base z

// Outlet z

// Number of cells at inner square

// Number of cells between inner square and circle
    
```


// x

// y

// * * * * * //

vertices

```
(
  (18.28 -18.28 0) // Vertex s0b = 0
  (-18.28 -18.28 0) // Vertex s1b = 1
  (-18.28 18.28 0) // Vertex s2b = 2
  (18.28 18.28 0) // Vertex s3b = 3

  (25.862430521898 -25.862430521898 0) // Vertex r0b = 4
  (-25.862430521898 -25.862430521898 0) // Vertex r1b = 5
  (-25.862430521898 25.862430521898 0) // Vertex r2b = 6
  (25.862430521898 25.862430521898 0) // Vertex r3b = 7

  (18.28 -18.28 1500) // Vertex s0t = 8
  (-18.28 -18.28 1500) // Vertex s1t = 9
  (-18.28 18.28 1500) // Vertex s2t = 10
  (18.28 18.28 1500) // Vertex s3t = 11

  (25.862430521898 -25.862430521898 1500) // Vertex r0t = 12
  (-25.862430521898 -25.862430521898 1500) // Vertex r1t = 13
  (-25.862430521898 25.862430521898 1500) // Vertex r2t = 14
  (25.862430521898 25.862430521898 1500) // Vertex r3t = 15
);
```

blocks

```
(
  //block0
  hex (1 0 3 2 9 8 11 10)
  square
  (40 40 150)
  simpleGrading (1 1 1)

  //block1
  hex (0 4 7 3 8 12 15 11)
  innerCircle
  (20 40 150)
  simpleGrading (1 1 1)
)
```

```
//block2
hex (3 7 6 2 11 15 14 10)
innerCircle
(20 40 150)
simpleGrading (1 1 1)

//block3
hex (2 6 5 1 10 14 13 9)
innerCircle
(20 40 150)
simpleGrading (1 1 1)

//block4
hex (1 5 4 0 9 13 12 8)
innerCircle
(20 40 150)
simpleGrading (1 1 1)
);

edges
(
  //Circle edges
  arc 7 4 (36.575 0 0)
  arc 4 5 (2.23957283394072e-15 -36.575 0)
  arc 5 6 (-36.575 4.47914566788143e-15 0)
  arc 6 7 (2.23957283394072e-15 36.575 0)

  //Circle edges
  arc 15 12 (36.575 0 1500)
  arc 12 13 (2.23957283394072e-15 -36.575 1500)
  arc 13 14 (-36.575 4.47914566788143e-15 1500)
  arc 14 15 (2.23957283394072e-15 36.575 1500)

  arc 3 0 (20.0 0 0)
  arc 0 1 (0 -20 0)
  arc 1 2 (-20 0 0)
  arc 2 3 (0 20.0 0)

  arc 11 8 (20.0 0 1500)
  arc 8 9 (0 -20 1500)
  arc 9 10 (-20 0 1500)
  arc 10 11 (0 20.0 1500)
);

patches
(
  wall walls
  (
```

```

(4 7 15 12)
(5 4 12 13)
(6 5 13 14)
(7 6 14 15)
)

patch inlet
(
(3 0 1 2)
(3 7 4 0)
(2 6 7 3)
(1 5 6 2)
(0 4 5 1)
)

patch outlet
(
(11 10 9 8)
(11 8 12 15)
(10 11 15 14)
(9 10 14 13)
(8 9 13 12)
)
);

mergePatchPairs
(
);

```

2. fvSchemes.

```

/*-----*- C++ -*-----*\
|=====|
|\ \ / Field | OpenFOAM: The Open Source CFD Toolbox |
|\ \ / Operation | Version: 5 |
|\ \ / And | Web: www.OpenFOAM.org |
|\ \ Manipulation |
\*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object fvSchemes;
}
// *****

ddtSchemes
{

```

```

    default    Euler;
}

gradSchemes
{
    default    Gauss linear;
}

divSchemes
{
    default    none;

    div(phi,U)    Gauss upwind;
    div(phi,T)    Gauss upwind;
    div(phi,k)    Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)    Gauss upwind;
    div(R)        Gauss linear;
    div((nuEff*dev2(T(grad(U)))))) Gauss linear;
}

laplacianSchemes
{
    default    Gauss linear uncorrected;
}

interpolationSchemes
{
    default    linear;
}

snGradSchemes
{
    default    uncorrected;
}

// *****
//
3. fvSolution.
/*-----* C++ *-----*\
|=====| |
| \ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \ / Operation | Version: 5 |
| \ / And | Web: www.OpenFOAM.org |
| \ / Manipulation | |
\*-----*/
FoamFile
{
    version    2.0;

```

```

format    ascii;
class     dictionary;
location  "system";
object    fvSolution;
}
// ***** //

solvers
{
    p_rgh
    {
        solver      PCG;
        preconditioner DIC;
        tolerance    1e-8;
        relTol       0.01;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol       0;
    }

    "(U|T|k|epsilon|R)"
    {
        solver      PBiCGStab;
        preconditioner DILU;
        tolerance    1e-6;
        relTol       0.1;
    }

    "(U|T|k|epsilon|R)Final"
    {
        $U;
        relTol       0;
    }
}

PIMPLE
{
    momentumPredictor no;
    nOuterCorrectors 1;
    nCorrectors      2;
    nNonOrthogonalCorrectors 0;
    pRefCell         0;
    pRefValue        0;
}

relaxationFactors
{

```



```

equations
{
  "(U|T|k|epsilon|R)" 1;
  "(U|T|k|epsilon|R)Final" 1;
}
}

// *****
//
4. alphat.
/*-----* C++ *-----*\
|=====|
|\ / Field | OpenFOAM: The Open Source CFD Toolbox |
|\ / Operation | Version: 5 |
|\ / And | Web: www.OpenFOAM.org |
|\ / Manipulation |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  location "0";
  object alphat;
}
// ***** //

dimensions [0 2 -1 0 0 0 0];

internalField uniform 2.3039e-5;

boundaryField
{
  walls
  {
    type alphasatJayatilekeWallFunction;
    Prt 0.71;
    value $internalField;
  }
  inlet
  {
    type fixedValue;
    value $internalField;
  }
  outlet
  {
    type zeroGradient;
  }
}

```

```
// *****
//
5. Épsilon.
/*----- C++ -----*\
|=====|
|\ / Field | OpenFOAM: The Open Source CFD Toolbox |
|\ / Operation | Version: 5 |
|\ / And | Web: www.OpenFOAM.org |
|\ Manipulation |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  location "0";
  object epsilon;
}
// ***** //
```

dimensions [0 2 -3 0 0 0];

internalField uniform 5e-5;

```
boundaryField
{
  walls
  {
    type epsilonWallFunction;
    value $internalField;
  }
  inlet
  {
    type fixedValue;
    value $internalField;
  }
  outlet
  {
    type zeroGradient;
  }
}
```

```
// *****
//
6. K.
/*----- C++ -----*\
|=====|
|\ / Field | OpenFOAM: The Open Source CFD Toolbox |
```

```

| \ \ / O peration | Version: 5 |
| \ \ / A nd | Web: www.OpenFOAM.org |
| \ \ M anipulation |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  location "0";
  object k;
}
// ***** //

dimensions [0 2 -2 0 0 0];

internalField uniform 1e-4;

boundaryField
{
  walls
  {
    type kqRWallFunction;
    value $internalField;
  }
  inlet
  {
    type fixedValue;
    value $internalField;
  }
  outlet
  {
    type zeroGradient;
  }
}

// *****
//
7. Nut.
/*-----* C++ *-----*\
|=====|
| \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / O peration | Version: 5 |
| \ \ / A nd | Web: www.OpenFOAM.org |
| \ \ M anipulation |
\*-----*/
FoamFile
{
  version 2.0;

```

```

format    ascii;
class    volScalarField;
location "0";
object    nut;
}
// ***** //

dimensions [0 2 -1 0 0 0 0];

internalField uniform 0.1;

boundaryField
{
    walls
    {
        type    nutkWallFunction;
        value    uniform 0;
    }
    inlet
    {
        type    fixedValue;
        value    $internalField;
    }
    outlet
    {
        type    zeroGradient;
    }
}

// *****
//
8. P.
/*----- C++ -----*\
|=====|
| \ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \ / Operation | Version: 5 |
| \ / And | Web: www.OpenFOAM.org |
| \ / Manipulation |
\*-----*/
FoamFile
{
    version    2.0;
    format    ascii;
    class    volScalarField;
    object    p;
}
// ***** //

dimensions [0 2 -2 0 0 0 0];

```

```

internalField uniform 101325;

boundaryField
{
    walls
    {
        type zeroGradient;
    }
    inlet
    {
        type zeroGradient;
    }
    outlet
    {
        type fixedValue;
        value $internalField;
    }
}

// *****
//
9. P_rgh.
/*-----* C++ *-----*\
|=====|
|\ / Field | OpenFOAM: The Open Source CFD Toolbox |
|\ / Operation | Version: 5 |
|\ / And | Web: www.OpenFOAM.org |
|\ / Manipulation |
\*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object p;
}
// *****

dimensions [0 2 -2 0 0 0];

internalField uniform 101325;

boundaryField
{
    walls
    {
        type zeroGradient;
    }
    inlet

```

```

    {
      type      zeroGradient;
    }
    outlet
    {
      type      fixedValue;
      value     $internalField;
    }
  }

// *****
//
10. T.
/*----- C++ -----*\
|=====|
|\  / Field | OpenFOAM: The Open Source CFD Toolbox |
|\  / Operation | Version: 5 |
|\  / And | Web: www.OpenFOAM.org |
|\  / Manipulation |
\*-----*/
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  object T;
}
// ***** //

dimensions [0 0 0 1 0 0 0];

internalField uniform 300;

boundaryField
{
  walls
  {
    type      fixedGradient;
    gradient  uniform 99651.9396;
  }
  inlet
  {
    type      fixedValue;
    value     uniform 298;
  }
  outlet
  {
    type      zeroGradient;
  }
}

```

```

// *****
//
11. U.
/*-----* C++ *-----*\
|=====|
|\ / Field | OpenFOAM: The Open Source CFD Toolbox |
|\ / Operation | Version: 5 |
|\ / And | Web: www.OpenFOAM.org |
|\ / Manipulation |
\*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    object U;
}
// ***** //

dimensions [0 1 -1 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
    walls
    {
        type noSlip;
    }
    inlet
    {
        type fixedValue;
        value uniform (0 0 6.66);
    }
    outlet
    {
        type inletOutlet;
        inletValue uniform (0 0 0);
        value uniform (0 0 0);
    }
}

// ***** //

```