



Universidad
Politécnica
de Cartagena



industriales
etsii UPCT

Descripción lagrangiana del mezclado en reactores de flujo oscilatorio

Titulación: Ingeniero Industrial
Intensificación: -
Alumno/a: Antonio Alcaraz Melgarejo
Director/a/s: Juan Pedro Solano Fdez

Cartagena, 4 de Septiembre de 2013

INDICE

CAPITULO 1	Tecnología y aplicaciones de los Reactores con deflectores de flujo oscilatorio. Problema del mezclado
CAPTITULO 2	Modelo Mackley
CAPITULO 3	Preparación del caso en Fluent. Obtención del repositorio de campo de derivadas, etc.
CAPITULO 4	Algoritmo de MATLAB para el seguimiento Lagrangiano y resolución de las ec. del Stretching & Folding
CAPITULO 5	Resultados. Valores Alfa medio para distintos Reynolds y Strouhal oscilatorios ensayados.
CAPITULO 6	Conclusiones
CAPITULO 7	Anexos

Capitulo 1 Tecnología y aplicaciones de los Reactores con deflectores de flujo oscilatorio. Problema del mezclado

1. REACTORES DE FLUJO OSCILATORIO.

1.1.1. TERMINOLOGÍA

Considérese inicialmente un flujo en un conducto liso de sección transversal circular, caracterizado por la velocidad media del flujo u , el diámetro interior del conducto D , y la viscosidad cinemática del fluido ν . Se denomina **flujo oscilatorio puro** (*pure oscillatory flow, POF*) a la superposición de un movimiento oscilatorio sobre dicho flujo. Para caracterizar un *POF* se requiere, además de los parámetros anteriores, un parámetro adicional: la velocidad de oscilación $u_0 = x_0\omega$, siendo x_0 la amplitud de oscilación y ω la frecuencia angular de oscilación. Insertando en el conducto una serie de elementos equiespaciados se obtiene un **flujo oscilatorio con orificios deflectores** (*oscillatory baffled flow, OBF*). En la caracterización de un *OBF* intervienen, además de los parámetros anteriores, dos parámetros geométricos: el diámetro interior del elemento deflector D_0 y la distancia entre elementos deflectores L .

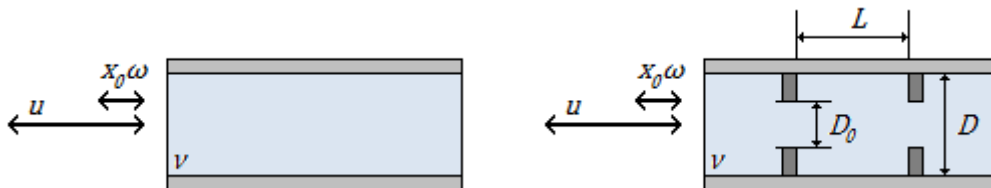


Figura 1.1. Parámetros de caracterización del flujo oscilatorio. Izquierda, flujo oscilatorio puro (POF). Derecha, flujo oscilatorio con orificios deflectores (OBF).

Los dispositivos de mezcla que generan en su interior un *OBF* se conocen como **columnas de flujo oscilatorio con orificios deflectores** (*oscillatory baffled colum, OBC*). Por lo tanto, son dos los aspectos que están siempre presentes en estos equipos de mezcla tubulares:

- Presencia de elementos insertados espaciados uniformemente.
- Existencia de un flujo oscilatorio sobre un flujo neto.

En la figura anterior se muestra la forma típica de un *OBC*, en el que los **elementos insertados** son discos con un único orificio central. Otros elementos que pueden ser también utilizados son discos con múltiples orificios, muelles helicoidales, láminas onduladas, superficies aleteadas, mallas, etc.

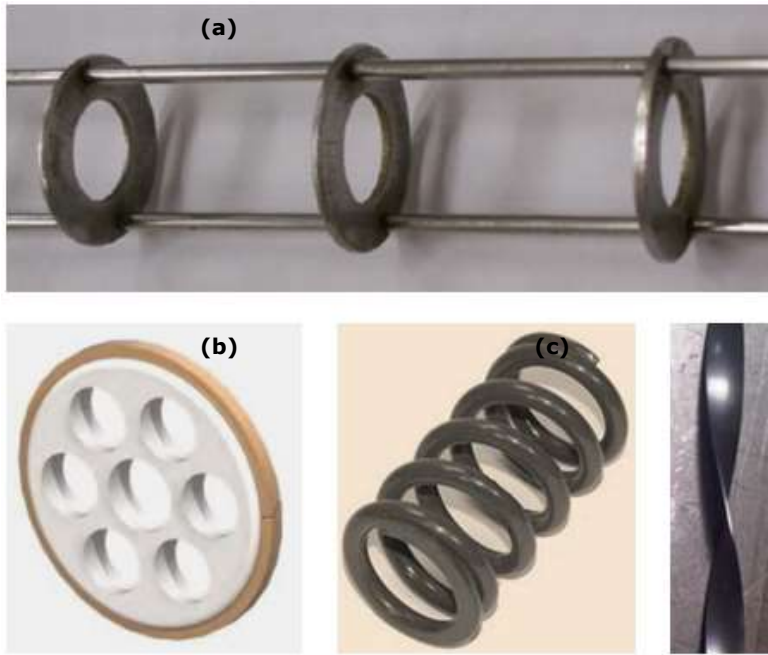


Figura 1.3. Otros elementos insertados en columnas de flujo oscilatorio. Disco con un único orificio (a), disco con múltiples orificios (b), muelle helicoidal (c), lámina ondulada (d).

La importancia de tener elementos constrictores del flujo se ve reflejada en la propia definición de Reynolds,

El Re oscilatorio depende de la f (frecuencia de oscilación) que multiplicado por 2π daría una idea de la rapidez con que se trata de hacer oscilar el fluido, esto se traduce en mayor o menor intensidad de mezclado. La x_0 (amplitud de oscilación) es un parámetro para el que hay que buscar un solución de compromiso ya que la intensidad de mezclado (Re_{osc}) aumenta con $X0$ pero disminuye el St_{osc} . Esto es porque el número de St oscilatorio habla sobre la capacidad del flujo pistón para propagar estructuras de 'Folding', (estas son parecidas a los vórtices). En cualquier caso intensidad de mezclado y propagación de vórtices se ven favorecidas por D que ocupen la mayor parte del tanque.

De ahí que otras geometrías que se prueban son mezcladores, láminas, o discos con 3 o mas agujeros.

Por su parte, el flujo que circula por el interior está compuesto por un **flujo neto**, caracterizado por una velocidad u_{net} constante a lo largo del tiempo, sobre el que se superpone un **flujo oscilatorio**, cuya velocidad u_{osc} sigue una ley sinusoidal con el tiempo. En la práctica, la magnitud de la componente oscilatoria es muy superior a la magnitud de la componente neta.

$$u_{osc} = 2\pi f x_0 \cos(2\pi f t) \quad (\text{ec.1.3})$$

$$u(t) = u_{net} + u_{osc} \quad (\text{ec.1.4})$$

La entrada de flujo se efectúa perpendicularmente a la columna, tal y como se observa en la *Figura 1.2*. La componente oscilatoria se consigue por medio de pistones, diafragmas o fuelles que actúan en la dirección axial de la columna, y que se sitúan en uno o en ambos extremos. Existen dos posibles configuraciones para la actuación de los pistones:

- Oscilación del fluido: el pistón actúa directamente sobre el flujo.
- Oscilación de los deflectores: el pistón desplaza a los elementos insertados.

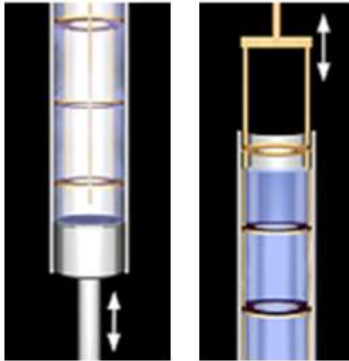


Figura 1.5. Configuración de los pistones. Izquierda, oscilación del fluido. Derecha, oscilación de los deflectores [51].

Aquellos dispositivos *OBC* en los que durante el proceso de mezcla se producen además reacciones químicas son conocidos como **reactores de flujo oscilatorio con orificios deflectores** (*oscillatory baffled reactor, OBR*). El amplio abanico de configuraciones posibles para estos dispositivos de mezcla se engloban bajo la denominación genérica de **reactores de flujo oscilatorio** (*oscillatory flow reactor, OFR*). Se distinguen dos modos de funcionamiento:

- Modo por lotes (*batch oscillatory baffled reactor, BOBR*).
- Modo continuo (*continuous oscillatory baffled reactor, COBR*).

1.1.2. Ventajas y aplicaciones.

Los reactores de flujo oscilatorio *OFR* ofrecen interesantes ventajas respecto a los procesos tradicionales de mezclado en tanques de agitación, como son una distribución más uniforme de la fase dispersa en la fase acuosa, una mejora de la transferencia de calor y masa, una estrecha distribución del tiempo de residencia de las partículas y un preciso control de las condiciones de mezcla. A diferencia del reactor tubular convencional, en el que se requiere una elevada velocidad superficial para obtener una buena mezcla, el *OFR* puede proporcionar un comportamiento de flujo pulsante en régimen de flujo laminar. Además, un *OFR* es ideal para la realización de reacciones largas, ya que la mezcla conseguida es independiente del gasto másico. Esto permite convertir en continuos a aquellos procesos en modo por lotes que requieren un elevado tiempo de residencia, y que obligan a diseños con alto ratio longitud-diámetro si son ejecutados con reactores convencionales.

Capítulo 1 Tecnología y aplicaciones de los Reactores con deflectores de flujo oscilatorio. Problema del mezclado

El empleo de estos reactores de flujo oscilatorio se da, principalmente, en las industrias químicas, bioquímicas y farmacéuticas. Algunas de las aplicaciones donde actualmente se emplean *OFR* son la polimerización, la fermentación, la síntesis orgánica, el procesamiento o la dispersión. Quizás la aplicación más relevante sea la fabricación de polímeros mediante **polimerización en suspensión**. En este proceso se mezcla una fase acuosa, generalmente agua, con una cierta cantidad de monómero insoluble, dando lugar a la formación de gotas de monómero en la fase acuosa. El *OFR* permite obtener una distribución adecuada y uniforme del tamaño de las partículas de polímero a través de control óptimo de las condiciones de mezcla, especialmente de la temperatura, así como evitar la fusión de las gotas de monómero durante la reacción. Con la polimerización en suspensión se producen polímeros como el cloruro de polivinilo, el poliestireno o las resinas de intercambio de cationes y aniones, y más recientemente el metacrilato de metilo, la acrilamida y el estireno.

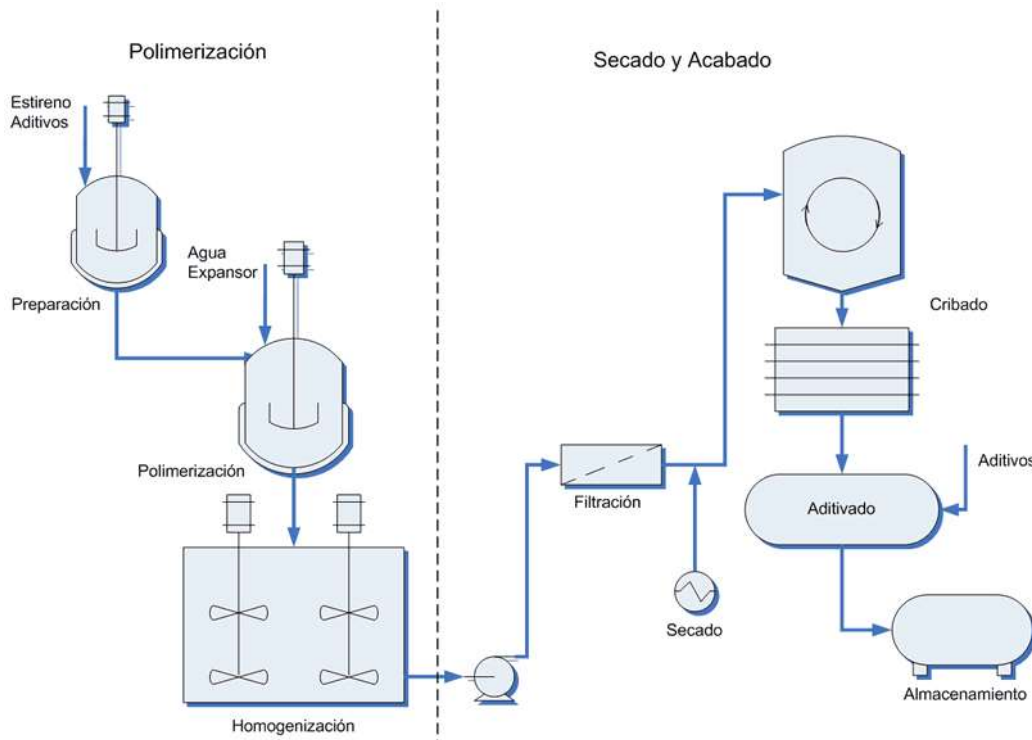


Figura 1.6. Esquema del proceso de polimerización en suspensión. Fabricación de estireno.

Una de las aplicaciones potenciales más atractivas para los reactores de flujo oscilatorio es la **producción de biocarburantes**. El empleo de reactores tipo tanque de agitación es una tecnología suficientemente conocida, y cuya eficiencia es ya difícilmente mejorable. Por ello se trabaja actualmente en el desarrollo de reactores intensificados: reactores estáticos, reactores con micro-canales, reactores con cavitación, reactores rotativos y reactores de flujo oscilatorio. Estos dispositivos están caracterizados por la miniaturización de los equipos tubulares, un mayor grado de conversión y una menor cantidad de catalizador, el uso de condiciones de proceso más suaves, la reducción de los costes energéticos asociados y el aumento de la capacidad de producción [16]. En última instancia, estas nuevas tecnologías permitirán aumentar la competitividad de las instalaciones de producción de biodiesel.

El nuevo **Plan de Energías Renovables 2011-2020** estima un aumento en la producción nacional de biocarburantes del 80 % durante la presente década, a través de una estrategia doble: desarrollar una normativa inequívoca para la comercialización de mezclas etiquetadas de biocarburantes (biodiesel y bioetanol) en estaciones de servicio, y adoptar las propuestas de la Iniciativa Industrial Europea sobre Bioenergía (EIBI). Estas propuestas se basan en el desarrollo de nuevas tecnologías termoquímicas para la transformación de materia prima vegetal en biocarburantes, y está apoyado por la creación de un Programa Nacional de Desarrollo Tecnológico en Biocarburantes y un Plan Nacional de Desarrollo Agroenergético.

1.1.3. Mecanismo de mezcla.

El patrón de movimiento del flujo en el interior de un *OFR* es generado por el efecto combinado de la presencia de elementos insertados y la existencia de flujo oscilatorio superpuesto. El flujo se acelera y se desacelera según una ley sinusoidal para la velocidad en función del tiempo. Con cada aceleración del flujo se forman anillos de vorticidad aguas abajo de los deflectores. Cuando el flujo se desacelera se forman nuevos anillos de vorticidad en el lado opuesto de los deflectores que, a su vez, desplazan a los anillos formados durante la aceleración hacia la región entre los deflectores.

La interacción entre los anillos de vorticidad creados durante la aceleración y desaceleración del flujo da lugar a complejas estructuras vorticales que intensifican el proceso de mezcla. En la *Figura 1.8* se muestra esquemáticamente este mecanismo:

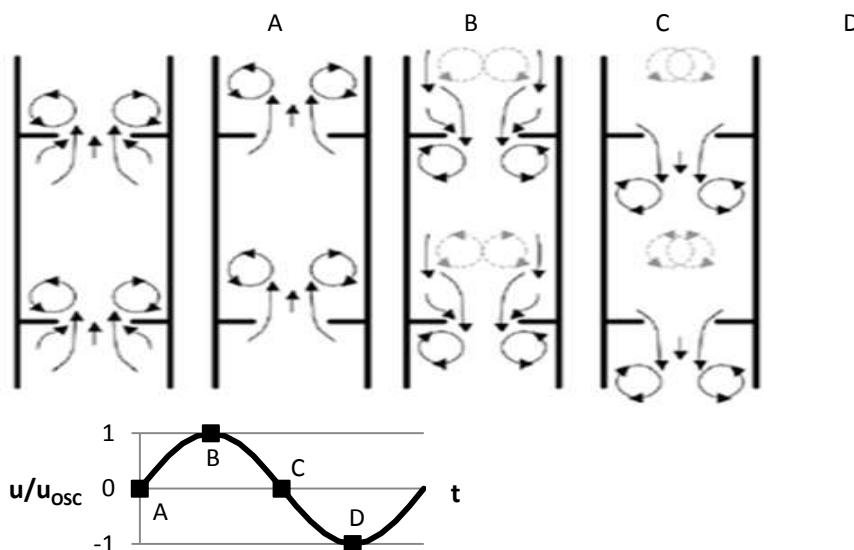


Figura 1.8. Esquema del mecanismo de mezcla en reactores de flujo oscilatorio.

Con la repetición sucesiva de ciclos de formación de anillos de vorticidad, la elevada componente de velocidad radial que se alcanza genera una mezcla uniforme en cada región entre deflectores, que se acumula a lo largo de la longitud del reactor [9, 12, 13]. Este mecanismo de mezcla posee, por tanto, **periodicidad espacial** y **periodicidad temporal**, ya

Capítulo 1 Tecnología y aplicaciones de los Reactores con deflectores de flujo oscilatorio. Problema del mezclado

que las estructuras vorticales son las mismas en cada una de las regiones entre deflectores, y se repiten con cada ciclo velocidad-tiempo. En la *Figura 1.9* se muestra una visualización de este mecanismo de mezcla:

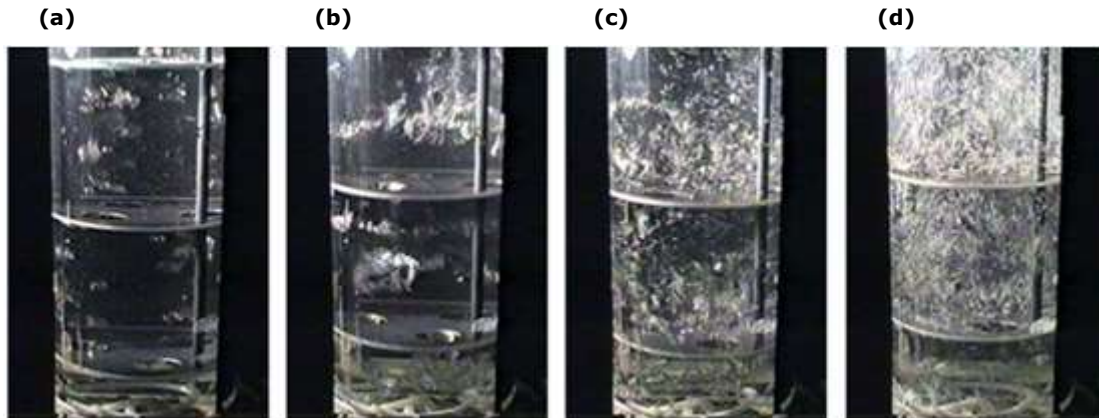


Figura 1.9. Visualización del mecanismo de mezcla. Sin oscilación (a). Comienzo de la oscilación (b). Tras un ciclo completo de oscilación (c). Tras varios ciclos completos de oscilación (d).

1.2. DEFINICIÓN DEL PROBLEMA.

1.2.1. Condiciones de operación.

El problema fluido que se plantea en este Proyecto Fin de Carrera es la resolución del campo de velocidades en un reactor de flujo oscilatorio (*oscillatory baffled reactor, OBR*). El estudio se centra en el problema mecánico del flujo oscilatorio, sin abordar el problema térmico.

El Proyecto se articula en dos bloques que cubren las dos configuraciones posibles para la actuación de los pistones en un *OBR*:

- En el primer bloque, **Oscilación del fluido**, se aborda la configuración en la que el fluido es oscilado axialmente por acción directa del pistón, permaneciendo fijos los orificios deflectores.
- En el segundo bloque, **Oscilación de los discos**, se estudia la configuración en la que la oscilación es ejercida sobre los orificios deflectores, que son desplazados axialmente por la actuación del pistón.

Para ambos bloques la geometría del reactor está formada por un conducto liso de sección transversal circular de diámetro $D = 25.0 \text{ mm}$, en el que se insertan, separados una distancia $L = 18.0 \text{ mm}$, una serie de discos con orificio, de diámetro interior $D_0 = 11.5 \text{ mm}$ y espesor $e = 1.5 \text{ mm}$.

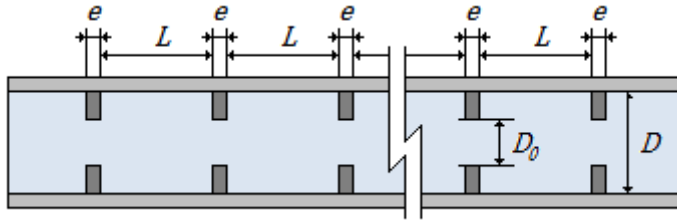


Figura 1.14. Geometría del reactor de flujo oscilatorio.

El fluido de trabajo es agua, de densidad $\rho = 998.2 \text{ kg/m}^3$, viscosidad dinámica $\mu = 0.001003 \text{ kg/ms}$ y viscosidad cinemática $\nu = 1.004809 \cdot 10^{-6} \text{ m}^2/\text{s}$. Las condiciones de operación simuladas vienen definidas los **parámetros de funcionamiento del pistón**: amplitud de oscilación x_0 y frecuencia de oscilación f , siendo la velocidad de oscilación $u_0 = 2\pi f x_0$. Nótese que la frecuencia de oscilación f se relaciona con la frecuencia angular de oscilación ω a través de $\omega = 2\pi f$.

1.3.2. Grupos adimensionales.

El problema fluidomecánico del flujo oscilatorio en un *OBR* se caracteriza tradicionalmente por dos **grupos adimensionales**: el número de Reynolds oscilatorio Re_o y el número de Strouhal St . El número de Reynolds oscilatorio describe la intensidad de la mezcla, mientras que el número de Strouhal mide la propagación efectiva de vórtices. La caracterización de las condiciones de operación según estos grupos adimensionales se recoge en la *Tabla 1.2*.

$$Re_o = \frac{D x_0 \omega}{\nu} \quad (\text{ec.1.5})$$

$$St = \frac{D}{4\pi x_0} \quad (\text{ec.1.6})$$

1.4.1 Oscilación del fluido.

En este apartado se aborda la configuración en la que el fluido es oscilado axialmente por acción directa del pistón, permaneciendo fijos los orificios deflectores. Para este estudio numérico, la componente oscilatoria del flujo se superpone sobre un **flujo neto nulo**: en la práctica, la magnitud de la componente oscilatoria es muy superior a la magnitud de la componente neta. Esta hipótesis es aplicada por los investigadores cuando el estudio numérico se enfoca a la resolución del problema fluidomecánico, como es el caso de este trabajo, ya que permite evaluar el patrón de

movimiento haciendo la solución independiente del flujo neto. Cuando el objetivo del estudio es la resolución del problema térmico o la evaluación de los tiempos de residencia, se emplea flujo neto no nulo.

1.4.2 Oscilación de los deflectores.

En este segundo bloque se estudia la configuración en la que la oscilación es ejercida sobre los orificios deflectores, que son desplazados axialmente por la actuación del pistón. Al igual que para la oscilación del fluido, se aplica la hipótesis de **flujo neto nulo**; en este caso la componente oscilatoria del flujo no existe y son las leyes de desplazamiento de los deflectores (*baffles*) las que gobiernan el movimiento:

$$x_b(t) = L_0 - L_0 \cos(2\pi ft) \quad (\text{ec.1.14})$$

$$u_b(t) = \frac{dx_b}{dt} = 2\pi f L_0 \sin(2\pi ft) \quad (\text{ec.1.15})$$

Nótese que, a priori, la amplitud de oscilación de los deflectores L_0 y la amplitud de oscilación del fluido x_0 son parámetros de funcionamiento diferentes. A efectos prácticos, en este estudio se consideran iguales con el fin de reproducir las mismas condiciones de operación, tanto con la oscilación del fluido como con la oscilación de los deflectores, para la posterior comparación de resultados.

$$L_0 = x_0 \quad (\text{ec.1.16})$$

**Capitulo 2 Condiciones e hipótesis
de Mackley.
Justificación de la elección de una
de las hipótesis.**

2.3 HIPOTESIS FORMULADAS POR EL AUTOR

2.3.1 HIPOTESIS DEL ELEMENTO LÍNEA

El movimiento de un elemento infinitesimal, línea y su velocidad de alargamiento se estudian a partir de las soluciones locales del campo de velocidad y derivadas. Así no cometemos error en la discretización a partir del campo de velocidades únicamente; vamos a usar las derivadas que obtenemos de resolver el caso con Fluent.

La velocidad de separación local es la que tiene una línea infinitesimal tal que es transportada mediante advección en el flujo.

Esta velocidad de separación o “stretch rate” puede ser estimada usando seguimiento Lagrangiano. Para ello partimos de líneas formadas por partículas que eran al principio adyacentes.

Si “local stretch rate” la velocidad de estiramiento de esa línea tiende a una exponencial, se puede afirmar que tenemos un flujo eficiente en términos de mezclado.

Los resultados obtenidos de seguir el desarrollo de esa línea pueden utilizarse para medir, “kinematic mixing rate” la velocidad de mezclado cinemática.

Resolviendo para Strouhal oscilatorio $St=1$, las ecuaciones del movimiento en 2D voy a conseguir las posiciones de las partículas en la celda del reactor.

Lo que sucede es una advección, un transporte del elemento línea por parte del flujo. La velocidad de alargamiento se calcula usando esta ecuación figura 2.3

$$\frac{d(\ln l)}{dt} = \sin^2 \theta * \frac{\partial v}{\partial y} + \sin \theta * \cos \theta * \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) + \cos^2 \theta * \frac{\partial u}{\partial x}$$

Fig.2.3

Calculamos también la velocidad de rotación, para obtener el ángulo que necesitamos introducir en la ecuación de velocidad de alargamiento. El ángulo se medirá en sentido contrario a las agujas del reloj respecto del eje axial. La velocidad de rotación viene dada por la ecuación fig.2.4

$$\frac{d(\varphi)}{dt} = \cos^2\theta * \frac{\partial v}{\partial x} + \sin\theta * \cos\theta * \left(\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) - \sin^2\theta * \frac{\partial u}{\partial y}$$

Fig 2.4

Tras esto Mackley propone un parámetro que cuantifica el mezclado, llamado 's' o 'alfa' en otras ocasiones dentro de "The simulation of stretch rates for the quantitative prediction and mapping of mixing within a cannell flow" Cambridge University, 1995). Consiste en hacer la media de la velocidad de alargamiento; esta proviene de resolver la ecuación de la figura 2.3 para cada uno de los elementos línea en estudio. Por lo tanto en lo que consiste este parámetro de mezclado es en sumar los valores discretos que en cada paso de tiempo adquiere la velocidad de estiramiento instantánea, fig.2.5

$$s = \frac{1}{t} \int_0^{t'} \frac{d(\ln l)}{dt'} dt'$$

Fig.2.5

2.3.4 LA HIPÓTESIS DE LA SEPARACION DE PARTICULAS

Partiendo de una distribución inicial de partículas como la de la fig.2.6

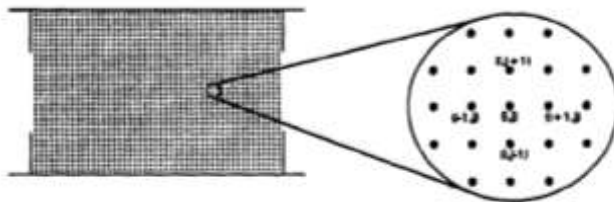


Fig.2.6

Lo que se pretende analizar es como cambia la línea a lo largo del tiempo fig.2.7



Fig.2.7

En este caso el coeficiente que usa M. Mackley para medir el mezclado es fig.2.8

$$s = \frac{lt}{l_0}$$

Fig.2.8

Y la velocidad de alargamiento promediada en el tiempo, fig.2.9

$$s = \frac{1}{t} \ln\left(\frac{lt}{l_0}\right)$$

Fig.2.9

2.4 METODO USADO

Uno de los objetivos de este proyecto es demostrar la hipótesis de elemento línea, y obtener las curvas para diferentes Reynolds. Observando si realmente se obtiene la asíntota para el parámetro de mezclado cuando el flujo lleva cierto tiempo desarrollándose.

La figura 2.10 muestra una comparación de ambos métodos.

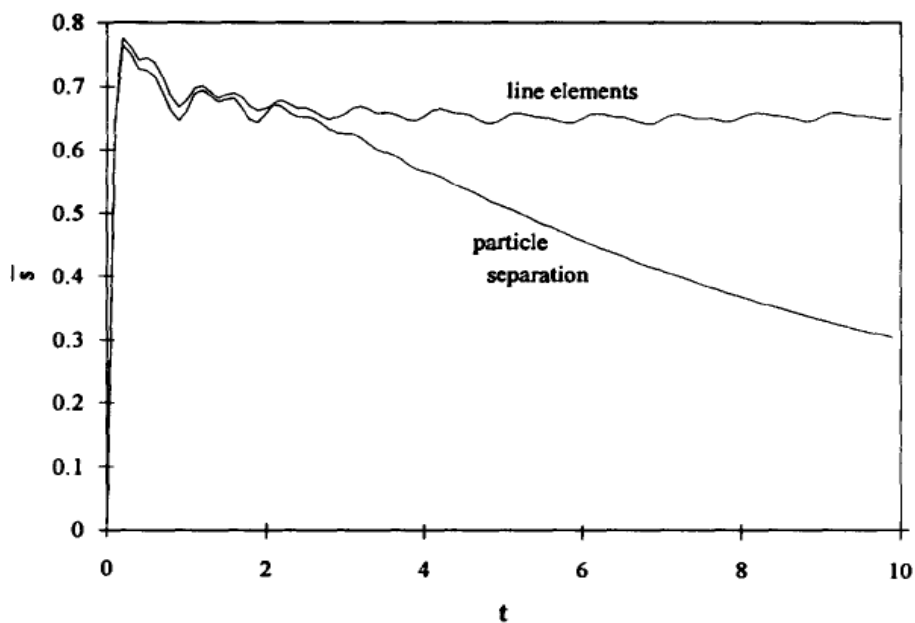


Fig.2.10

Vemos que la hipótesis de elemento línea ofrece una tendencia asíntota con pequeñas oscilaciones debido a lo complejo de las Ecuaciones Diferenciales que se utilizan; y a que el paso de tiempo no se ha disminuido en exceso con el fin de agilizar simulaciones y cálculos.

Con la hipótesis de separación de partículas el parámetro alfa empieza a decrecer a partir de cierto tiempo.

Pues que la hipótesis de elemento línea es la que más sentido físico tiene es la que seguiremos. Pues el mezclado a lo largo del tiempo no debe caer, en todo caso lo que sucedería es que llega un momento que las partículas se han dispersado tanto y han sufrido tal advección que el espacio donde están confinadas no es suficiente para que se alejen más y de ahí que lleguemos a esa asíntota.

Además de tratar de reproducir esta asíntota, demostraremos que puede obtenerse independientemente de que elemento línea usemos, es decir, tomamos dos partículas adyacentes a 0, 45 o 90 grados. Hecho que ya indica el autor en el artículo de 2002

“The quantitative description of fluid mixing using Lagrangian and concentration-based numerical approaches” de Mackley & Neves Saraiva.

El autor evidencia que el valor del parámetro alfa es independiente del número de partículas utilizado. Fig.2.11

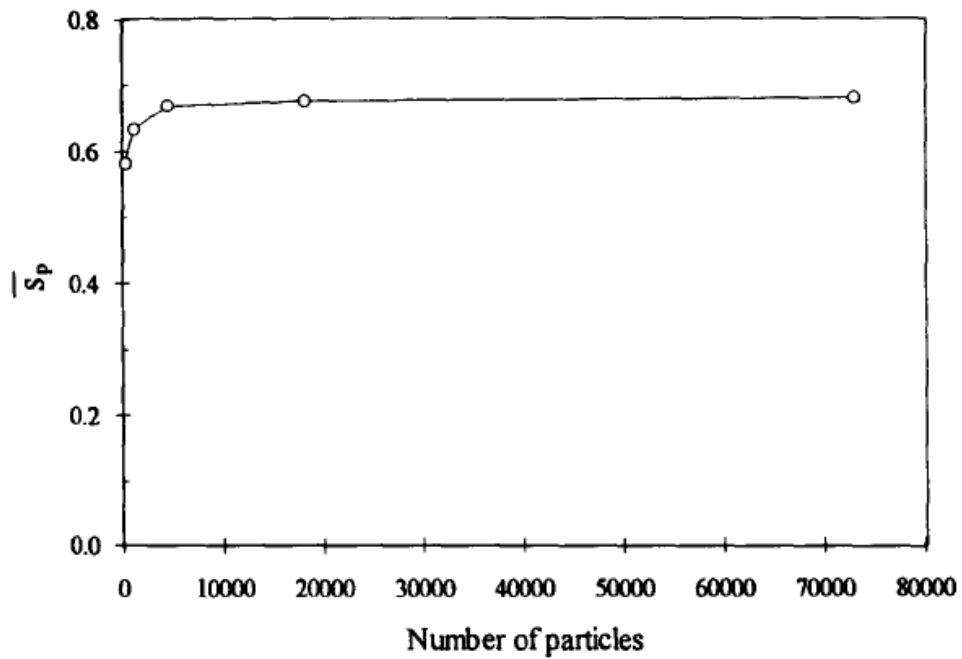


Fig.2.11

Además se exige un mínimo de elementos línea; que se establece entorno a 1000. Fig 2.12 Aunque el autor muestra 3600 elementos como la mejor forma de obtener una solución suave. En este proyecto se ha considerado que con algo más de 1000 elementos línea se puede obtener una solución adecuada al alcance buscado.

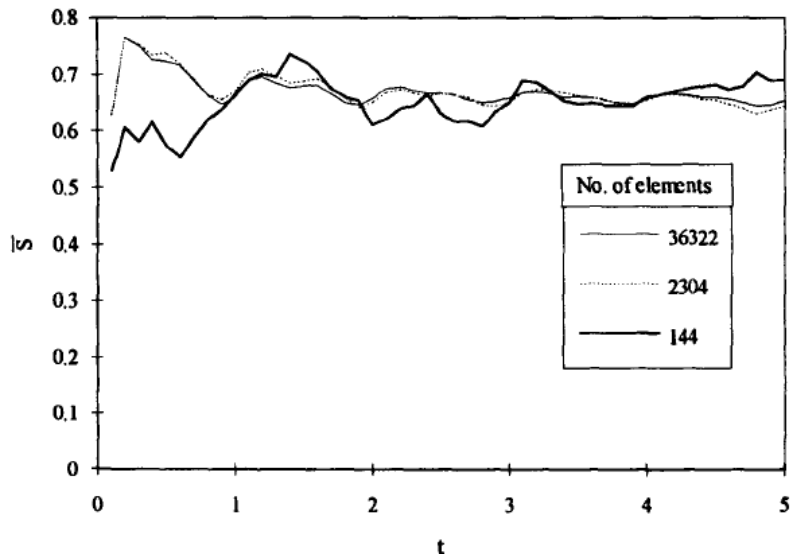


Fig. 2.12

Estos elementos línea habrá que computerizarlos, usando el algoritmo que está descrito en el capítulo 4 que trata sobre el Algoritmo del método. Como se observa un número suficientemente grande de elementos línea favorece soluciones más suavizadas que si usamos pocos elementos línea, como por ejemplo 144 elementos.

2.5 Comentarios sobre Teoría del mezclado

Las imágenes del campo de velocidad, pueden servir para el diseño de dispositivos eficientes de mezclado, fig 2.13 muestra la magnitud; fig. 2.14 los vectores

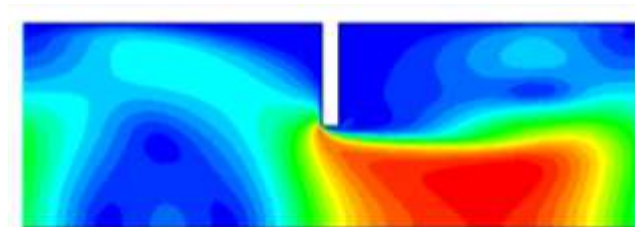


Fig. 2.13

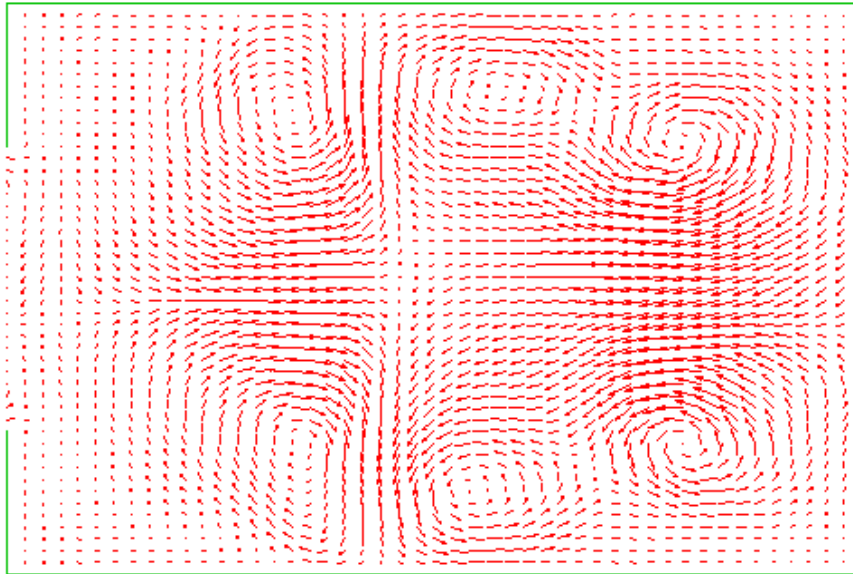


Fig. 2.14

Los vórtices que se observan es lo que se llaman estructuras de Folding. Siguiendo su propagación en el tiempo podemos a primera vista tener una idea sobre la eficiencia del dispositivo a la hora de mezclar o transferir calor al fluido.

El “stretching & folding” de las líneas de fluido (en 2-D) está muy íntimamente ligado al mezclado, y habla sobre la advección caótica de un flujo determinado en el reactor de estudio. En la parte teórica, hay que trabajar con los teoremas de transporte, deformación de elementos línea (caso 2-D) y la ecuación de vorticidad.

La teoría del mezclado se puede describir atendiendo a términos de la descripción cinemática, mecánica del flujo.

2.6 CONCLUSIÓN DEL CAPÍTULO

El objetivo del proyecto es el de hacer un estudio computerizado del “stretching & folding”. Siguiendo el modelo Lagrangiano propuesto por Roberts, E.P.R. & Mackley, M.R. (1995) The simulation of stretch rates for the quantitative prediction and mapping of mixing within a channel flow. Partiendo de un campo discreto de velocidad.

Estudiamos como las líneas formadas por dos partículas cada vez adquieren una orientación diferente y una longitud mayor. Y de esto trata el mezclado eficiente que

esas líneas sean transportadas por la mayor parte del reactor ofreciendo la mayor superficie posible(caso 3-D), la mayor longitud posible (caso 2-D).

CAPITULO 3 PREPARACIÓN DEL CASO EN FLUENT. OBTENCIÓN DEL REPOSITORIO DE CAMPO DE VELOCIDAD, DERIVADAS, ETC.

3. CASO DE FLUENT

3.1 PROBLEMA A RESOLVER

Resolvemos el caso del flujo oscilatorio en un reactor con deflectores. Los deflectores son anillos, cuya geometría se especifica en la tabla 3.1

	D(mm)	d(mm)	A(mm ²)	L=3xd (mm)
One Hole	25	12.5	122.7	37.5

Tabla 3.1

Tras exportar la geometría desde GAMBIT, fig 3.2 procedemos a plantear el caso.

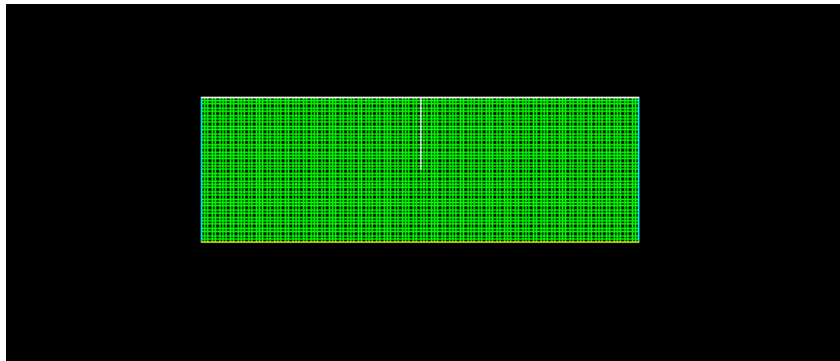


Figura 3.2

Como vemos en la figura 3.2 aparece sólo la mitad del anillo deflector. El hecho de poder resolver el problema usando la mitad de superficie, permite hacer un mallado más fino y así formar mayor cantidad de elementos línea. La importancia de un número alto de elementos línea se explica en el cap.2, apartado 2.4

El reactor de flujo oscilatorio propuesto por Mackley, tiene esta geometría; fig.2.1 Vemos que son discos en el interior que provocarán vorticidad en el flujo y adelantarán la aparición del flujo turbulento.

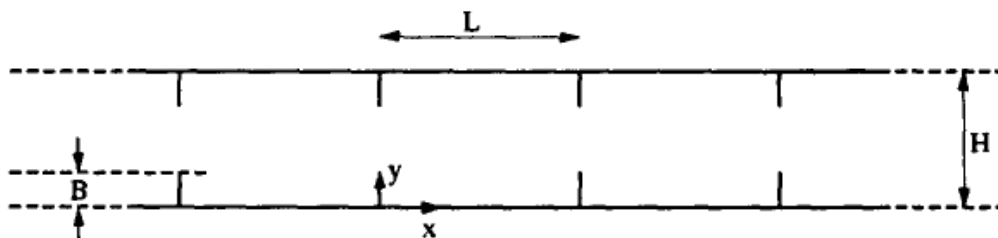


Fig.2.1

Las relaciones impuestas son $L=1.5 \cdot H$, $B=0.25 \cdot H$ la geometría que se ha ensayado en este caso ha sido $H=1.25$ cm, $L=3.75$ cm como vemos en la fig.2.2; como el flujo es axilimétrico, basta con resolver el campo de velocidad de un de las mitades partiendo el reactor respecto al plano axial como vemos en la figura 2.2 que además muestra el campo de velocidad estacionario para un determinado Reynolds, y sin resolver aún el

Capítulo 3 Preparación del caso en Fluent. Obtención del repositorio de campo de derivadas, etc.

flujo periódico. Se aprecia como la reducción de sección supone un aumento de la velocidad, y se crean zonas de vorticidad debido a la geometría de discos insertados en tubo liso que proponen Robert & Mackley (en "The simulation of stretch rates for the quantitative prediction and mapping of mixing within a channel flow" Cambridge University, 1995).

3.1.1 Dominio del reactor

Las diferentes zonas definidas en el dominio se especifican en la tabla 3.3

```
Zone sizes on domain 1:  
 4800 quadrilateral cells, zone 2.  
 120 2D wall faces, zone 3.  
 20 2D wall faces, zone 4.  
 20 2D wall faces, zone 8.  
 120 2D axis faces, zone 5.  
 40 2D periodic faces, zone 7.  
 9420 2D interior faces, zone 9.  
 40 2D shadow faces, zone 6.  
 4980 nodes.
```

Tabla 3.3

Definimos la zona 7 y la 6 como cara periódica y cara sombra. Estas son las zonas de entrada y salida; para las cuales se define el gasto másico como condición de contorno periódica.

3.1.2 Definición del modelo, fluido, condiciones de operación

El fluido incompresible que se usa es agua con las propiedades mostradas en la figura 3.4

Capítulo 3 Preparación del caso en Fluent. Obtención del repositorio de campo de derivadas, etc.

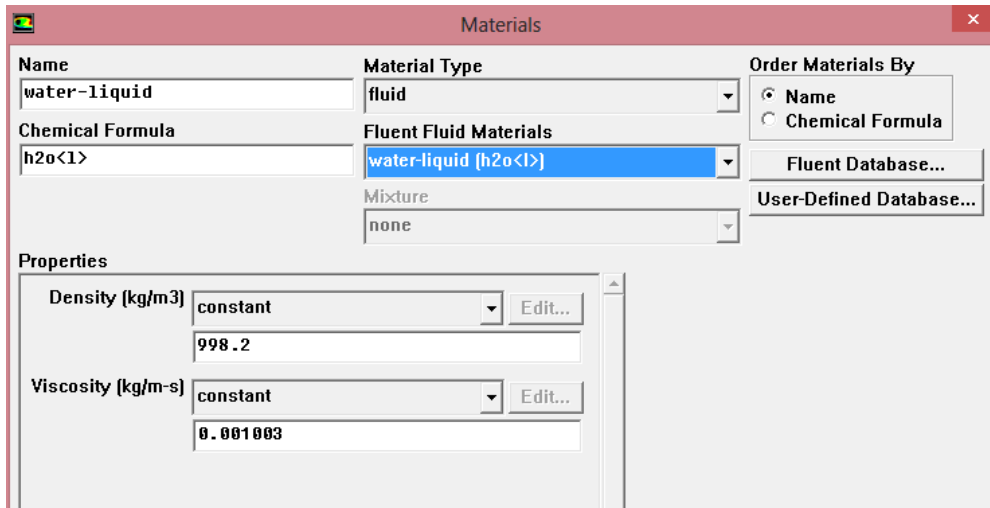
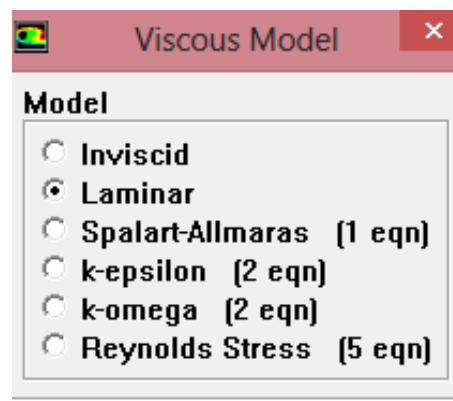


Figura 3.4 Ventana de Fluent

En la interfaz de fluent, siguiendo la ruta define/models/ como el flujo es una sólo fase deshabilitamos el modelo multifase (multiphase). También la ecuación de la energía, ya que no tiene influencia en el problema.

El modelo de viscosidad será el Laminar, como se ve en la fig. 3.5 por simplificación a la hora de obtener y procesar los campos de velocidad. Ya que la herramienta que queríamos desarrollar era para tratar los campos de velocidad y computerizar el método de “Stretching & Folding”.



Las condiciones de operación se muestran en la figura 3.6

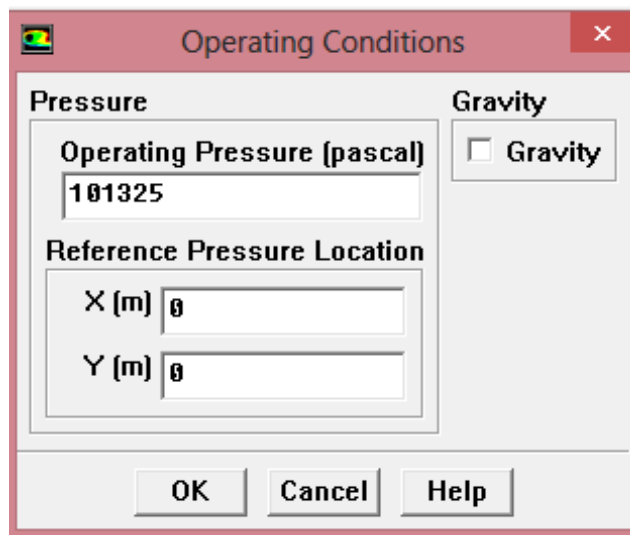


Figura 3.6

Esta ventana me permite controlar la solución utilizando los factores de relajación figura 3.7 son los que han dado mejores resultados a la hora de una convergencia suave. El acoplamiento presión-velocidad permanecerá siempre *SIMPLE*, ya que los casos a estudiar presentan un comportamiento laminar, además de que el esquema SIMPLE presente menos problemas de convergencia y es más rápido.

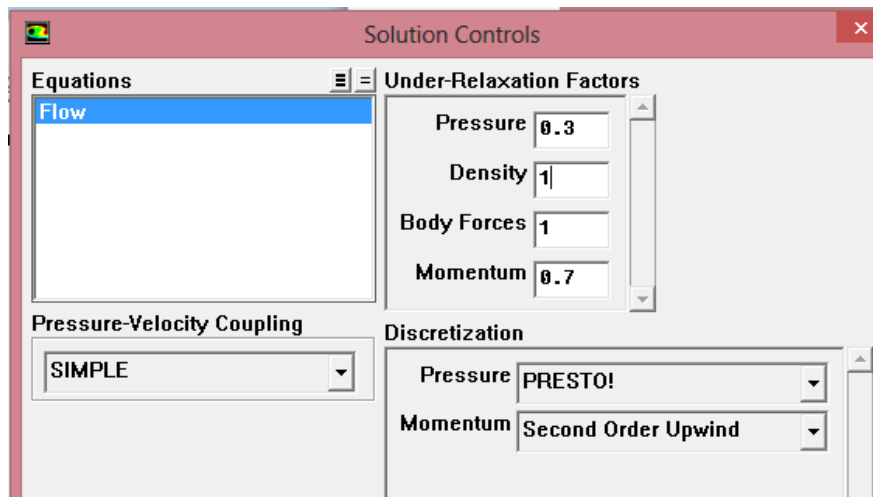


Figura 3.7

3.1.3 CRITERIOS DE CONVERGENCIA

Los criterios de Convergencia utilizados son los más adecuados después de probar $10e-6$ para la continuidad. Las iteraciones que aparecen son las de un primer caso. Pero posteriormente este mismo caso se resuelve con otras 2000 iteraciones. Al final llegamos a hacer 3000 iteraciones hasta que cumplimos de sobra con los criterios de convergencia, figura 3.8

Capítulo 3 Preparación del caso en Fluent. Obtención del repositorio de campo de derivadas, etc.

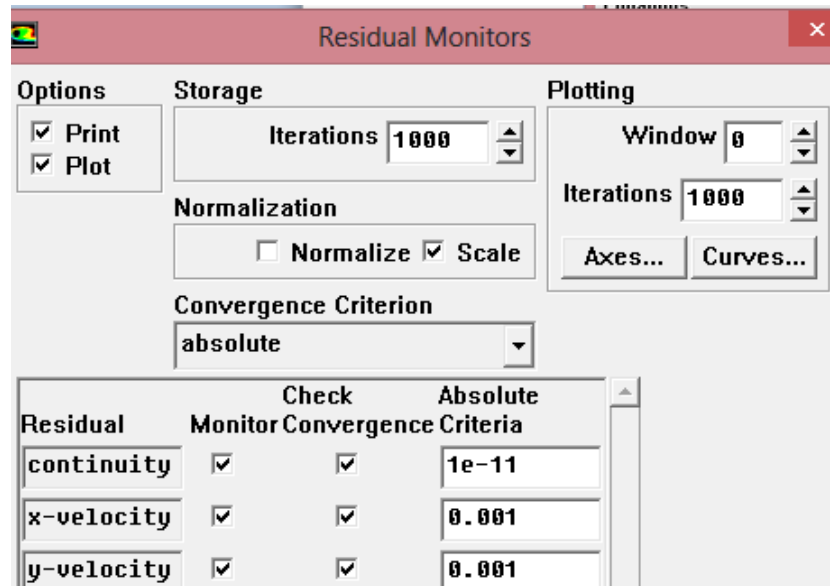


Figura 3.8

Para confirmar que el caso simulado ha convergido solo queda evaluar el balance de masa (recordamos que la ecuación de la energía no está seleccionada y por lo tanto no es posible hacer un balance de energía). Este balance se efectúa mediante el comando *Report Fluxes*, seleccionando las superficies *entrada* y *salida*. Los resultados se muestran en la figura 3.9

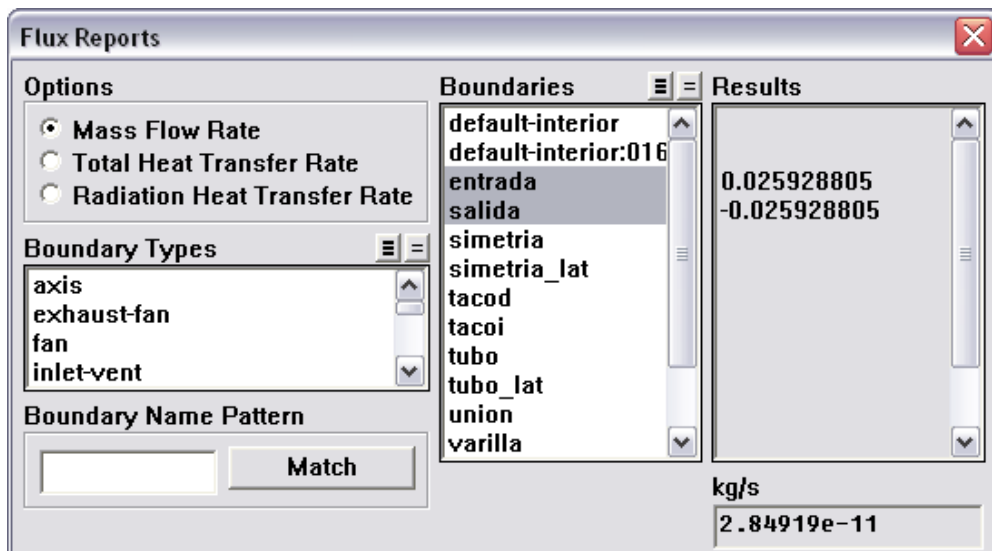


Figura 3.9

3.2 METODOLOGÍA DE SIMULACIÓN

3.2.1 SIMULACIÓN ESTACIONARIA

En la metodología de simulación que se lleva a cabo para abordar la resolución del campo de velocidades en un *OBR* es preciso distinguir entre simulación estacionaria y simulación no estacionaria. El problema del flujo oscilatorio es inherentemente no estacionario puesto que las leyes de gobierno del movimiento del fluido son leyes sinusoidales temporales (véase el capítulo 1). En la expresión general ec.3.1 es necesario establecer el **caso de inicio del ciclo** a través de la fase φ , es decir, fijar el gasto másico.

$$\dot{m}(t) = \dot{m}_0 \text{sen}(2\pi ft + \varphi) \quad (\text{ec.3.1})$$

Para la oscilación del fluido, se inicia el ciclo de oscilación en el punto de **máximo gasto másico**, es decir, para $\varphi = \pi/2$. Así, la ley temporal sinusoidal para el gasto másico se expresa como:

$$\dot{m}(t) = \rho \left(\frac{\pi D^2}{4} \right) 2\pi f x_0 \cos(2\pi ft) \quad (\text{ec.3.2})$$

En la figura 3.10 se representa la evolución del gasto másico, partiendo del caso estacionario con gasto másico constante:

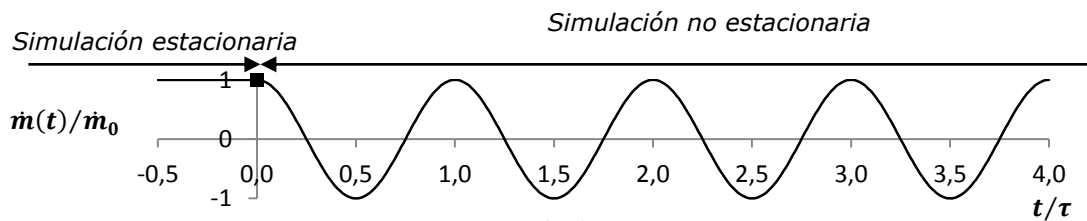


Figura 3.10

3.2.2 Simulación no estacionaria.

A partir del caso de inicio del ciclo comienza la simulación no estacionaria, con gasto másico variable en el tiempo. Para ello es preciso discretizar la ecuación ec.3.2 imponiendo un paso de tiempo en el eje temporal, que se obtiene mediante la **condición de Courant-Friedrich-Levy** (condición CFL), y que para el caso de n dimensiones toma la forma:

Capítulo 3 Preparación del caso en Fluent. Obtención del repositorio de campo de derivadas, etc.

$$\Delta t \sum_{i=1}^n \frac{u_{xi}}{\Delta x_i} \leq C \quad (\text{ec.3.3})$$

siendo Δt el paso de tiempo, Δx_i el tamaño de la celda en la dirección i , u_{xi} la velocidad en la celda de tamaño Δx_i y C el número de Courant.

Matemáticamente, la condición CFL es una condición de **convergencia** para ecuaciones diferenciales en derivadas parciales. Para esquemas explícitos es $C = 1$, mientras que para esquemas implícitos es $C = \infty$. Nótese que la aplicación en FLUENT del *solver Pressure Based*, por tratarse de flujo incompresible, conlleva obligatoriamente el uso de un esquema implícito. Sin embargo, desde el punto de vista de la **estabilidad** de la solución, sí existen recomendaciones acerca de la limitación del número de Courant. Para las simulaciones no estacionarias efectuadas en el presente Proyecto se establece como valor máximo $C = 30$, de acuerdo con el rango $20 \leq C \leq 40$

Una vez fijado el paso de tiempo Δt , el número total de casos n en los que se discretiza un ciclo de la ley sinusoidal depende de la frecuencia de oscilación f . Para reducir el volumen de datos generado se opta por guardar en cada ciclo, no el número total de casos resueltos n , sino un número de casos menor $n_0 = 64$ suficientemente representativo.

$$n = \frac{1}{f\Delta t} \quad (\text{ec.3.7})$$

En consecuencia, el cálculo del paso de tiempo definitivo se realiza según:

$$n' = \frac{1}{f\Delta t_1} \quad (\text{ec.3.8})$$

$$n = \text{ent} \left(\frac{n'}{n_0} \right) n_0 \quad (\text{ec.3.9})$$

$$\Delta t = \frac{1}{nf} \quad (\text{ec.3.10})$$

Dada la periodicidad espacial del patrón de movimiento del flujo, existe la posibilidad de reproducir la misma solución obtenida en los pasos espaciales centrales de la simulación completa de 8 pasos con un modelo de un único paso, aplicando condiciones de contorno periódicas. Así, la **simulación periódica** sobre un dominio computacional de **1 paso espacial** permite resolver el mismo problema fluidodinámico con un menor coste computacional.

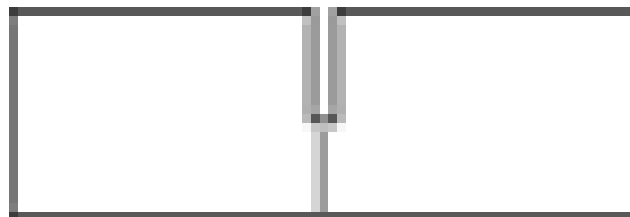


Figura 3.11

Lo que hay representado en la figura 3.11 no es más que uno de los tanques que componen el reactor completo. Un reactor puede tener entre 8 y 15 tanques. Como comentario adicional, el estudio sobre el rendimiento de un reactor de 8 tanques frente uno de 15 o más se llevaría a cabo con un modelo de fluent diferente (ya teniendo en cuenta que el flujo no tiene porqué ser laminar) y respecto a este tema puedo citar el artículo de investigación “EVALUATION OF AXIAL DISPERSION AND MIXING PERFORMANCE IN OSCILLATORY BAFFLED REACTORS USING CFD” 2012, Xiong-Wei Ni, Mikko Manninen, Kirsi Immonen, Elena Gorshkova; “HANDBOOK OF INDUSTRIAL MIXING” editado por Edward L. Paul, Suzanne M. Kresta.

Tras aclarar esto, el punto importante es que este tanque se elige después de desechar los primeros que se ven afectados por la entrada. Si aprovechamos que $L \gg D$ (longitud del tanque, diámetro del deflector) podemos parar en el tercer o cuarto tanque del reactor, Y usar las condiciones de contorno estacionarias, además de definir las zonas de entrada y salida como se explica en el aptdo. 3.1.1

3.3 CONDICIONES DE CONTORNO . SOLVER

El programa Fluent trabaja de la siguiente manera:

Se introduce como ‘Periodic Condition’ el gasto másico que va a pasar por el tanque. De este modo el Software, calcula el gradiente de presión que debe mantener. De este modo las condiciones necesarias para resolver las ec. diferenciales ya están definida.

La ecuación con la que calculamos el gasto másico es la citada en el punto 3.2.1 de este capítulo:

$$\dot{m}(t) = \rho \left(\frac{\pi D^2}{4} \right) 2\pi f x_0 \cos(2\pi f t) \quad (\text{ec. 3.2})$$

Como vemos según queramos estudiar diferentes Reynolds oscilatorios (fig 3.12) y Strouhal oscilatorios (fig 3.13), debemos cambiar la condición de gasto másico. Estos números se definen para el caso oscilatorio así:

$$Re_o = \frac{2\pi f D x_0}{\nu}$$

Fig. 3.12

$$St = \frac{D}{4\pi x_0}$$

Fig. 3.13

3.3.1 SOLVER

En el problema fluidodinámico tiene como objetivo obtener el comportamiento del flujo en ausencia de aporte de energía externo y disipación viscosa. Como consecuencia las ecuaciones de cantidad de movimiento y energía se encuentran desacopladas. Se emplea pues el *solver segregado* y no el *solver acoplado*, reduciendo así el tiempo de cálculo sin perder ninguna precisión. La figura 3.14 muestra la ventana del Solver.

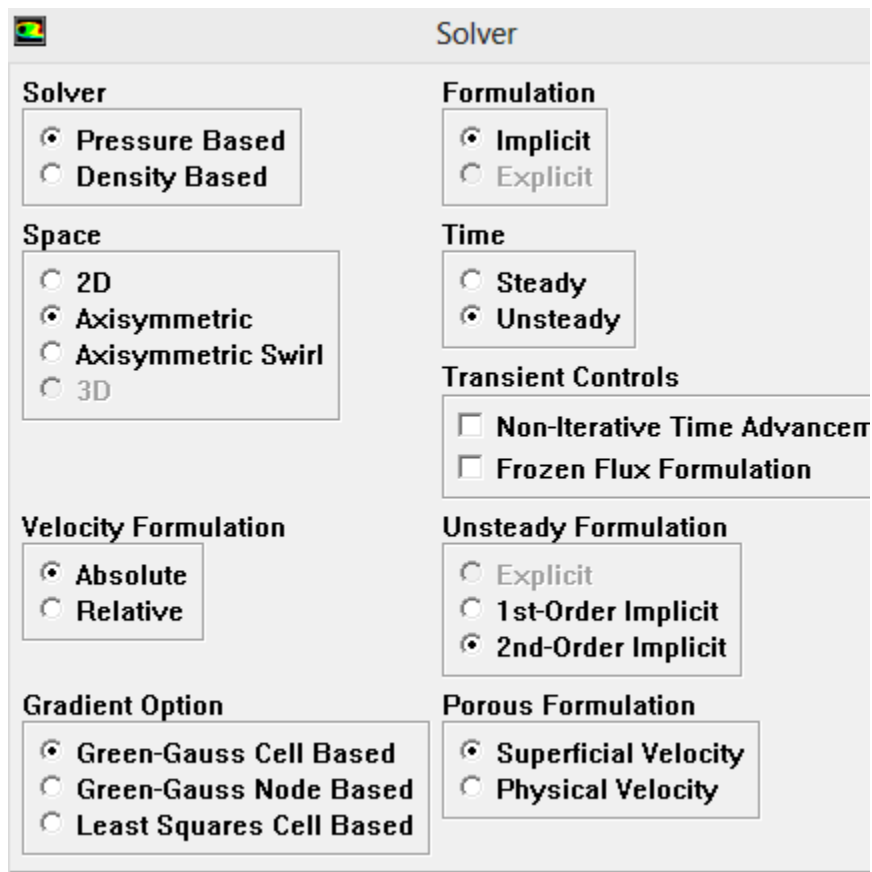


Fig. 3.14

El modelo basado en presión es el que se usa con fluidos incompresibles como es el caso. La formulación implícita como mencionábamos en el apartado 3.2.2 y condición de axisimetría.

3.4 OBTENCIÓN DE VARIABLES A TRATAR

Se utiliza un fichero Journal para exportar las variables que necesitamos. Este fichero se ha escrito usando un script de Matlab. Es Fluent quien lee el Journal para resolver y exportar el campo de velocidad, etc. arrojando como resultado los n=64 pasos que sirven para discretizar adecuadamente el ciclo continuo de oscilación (según el número de Courant).

Un ejemplo de fichero obtenido se muestra en la fig. 3.15 donde aparecen el 0.01 % de los datos que contiene cada fichero. Lo cual teniendo en cuenta el número de Reynolds osc. Ensayados y de Strouhal sería menos del 0.00001%.



nodenumber	x-coordinate	y-coordinate	cell-volume-2d	dradial-velocity-dy	dradial-velocity-dx	daxial-velocity-dy	daxial-velocity-dx	radial-v	
1	1.875000089E-002	6.250000297E-003	9.765625464E-008	2.130805588E+001	3.924533844E+001	-3.512887192E+001	-1.316251659E+001	0.000000000E+000	
2	1.843750088E-002	6.250000297E-003	9.765626174E-008	1.077258587E+001	3.155922318E+001	-2.733337975E+001	-5.686859608E+000	-3.518792987E-002	
3	1.812500086E-002	6.250000297E-003	9.765626174E-008	5.472638130E+000	-7.233711720E+000	-1.693504333E+001	1.961508751E+000	-3.496824205E-002	
4	1.781250085E-002	6.250000297E-003	9.765626174E-008	3.529185295E+000	-1.008902740E+001	-1.094312286E+001	2.107031107E+000	-3.219684586E-002	
5	1.750000083E-002	6.250000297E-003	9.765626174E-008	2.260094643E+000	-7.761373520E+000	-7.369390488E+000	2.85358036E+000	-2.958233468E-002	
6	1.718750082E-002	6.250000297E-003	9.765626174E-008	1.423534632E+000	-6.375424385E+000	-4.958215714E+000	3.077129841E+000	-2.739205584E-002	
7	1.687500080E-002	6.250000297E-003	9.765626174E-008	8.118160963E-001	-5.441282272E+000	-3.167779446E+000	3.348775387E+000	-2.554919012E-002	
8	1.656250079E-002	6.250000297E-003	9.765626174E-008	3.349416852E-001	-4.868919373E+000	-1.765269160E+000	3.533119678E+000	-2.393852733E-002	
9	1.625000077E-002	6.250000297E-003	9.765626174E-008	-5.142170191E-002	-4.574452877E+000	-6.270018220E-001	3.667780876E+000	-2.246284485E-002	
10	1.593750076E-002	6.250000297E-003	9.765626174E-008	-3.694748878E-001	-4.499992371E+000	3.188711107E-001	3.748933315E+000	-2.104477026E-002	
11	1.562500074E-002	6.250000297E-003	9.765626174E-008	-6.311222911E-001	-4.606257915E+000	1.114638090E+000	3.777772427E+000	-1.962175779E-002	
12	1.531250073E-002	6.250000297E-003	9.765626174E-008	-8.430976882E-001	-4.867118359E+000	1.786388516E+000	3.749981642E+000	-1.814137958E-002	
13	1.500000071E-002	6.250000297E-003	9.765626174E-008	-1.011402488E+000	-5.266873837E+000	2.351692677E+000	3.660244942E+000	-1.655781455E-002	
14	1.468750070E-002	6.250000297E-003	9.765626174E-008	-1.150641799E+000	-5.802965641E+000	2.819380283E+000	3.522884130E+000	-1.482871640E-002	
15	1.437500068E-002	6.250000297E-003	9.765626174E-008	-1.325631499E+000	-6.471571445E+000	3.251307487E+000	3.389610052E+000	-1.291287504E-002	
16	1.406250067E-002	6.250000297E-003	9.765626174E-008	-1.530296922E+000	-7.184043884E+000	3.685509443E+000	3.238561153E+000	-1.077944972E-002	
17	1.375000065E-002	6.250000297E-003	9.765626174E-008	-1.587973118E+000	-7.749307156E+000	4.005216122E+000	2.928914309E+000	-8.442210034E-003	
18	1.343750064E-002	6.250000297E-003	9.765626174E-008	-1.455438018E+000	-8.104001999E+000	4.142820358E+000	2.393215656E+000	-5.965813994E-003	
19	1.312500062E-002	6.250000297E-003	9.765626174E-008	-1.195181370E+000	-8.303170204E+000	4.142577171E+000	1.684225082E+000	-3.403845010E-003	
20	1.281250061E-002	6.250000297E-003	9.765626174E-008	-8.457965851E-001	-8.288764000E+000	4.066173553E+000	9.278842211E-001	-8.085679146E-004	
21	1.250000059E-002	6.250000297E-003	9.765626174E-008	-4.794314802E-001	-8.040581703E+000	3.931887627E+000	1.947067380E-001	1.743622939E-003	
22	1.218750058E-002	6.250000297E-003	9.765626174E-008	-1.443408430E-001	-7.668605804E+000	3.763757467E+000	-5.445547104E-001	4.197418224E-003	
23	1.187500056E-002	6.250000297E-003	9.765626174E-008	1.351042539E-001	-7.265549183E+000	3.586818933E+000	-1.228099346E+000	6.530535873E-003	
24	1.156250055E-002	6.250000297E-003	9.765626174E-008	3.711321652E-001	-6.875562191E+000	3.381188869E+000	-1.809690595E+000	8.741185069E-003	
25	1.125000053E-002	6.250000297E-003	9.765626174E-008	5.689532757E-001	-6.575037003E+000	3.156320572E+000	-2.327293873E+000	1.084214821E-002	
26	1.093750052E-002	6.250000297E-003	9.765626174E-008	7.503499389E-001	-6.482596397E+000	2.928959846E+000	-2.808607101E+000	1.287791785E-002	
27	1.062500050E-002	6.250000297E-003	9.765626174E-008	8.502571583E-001	-6.583522320E+000	2.668365479E+000	-3.227798462E+000	1.491351239E-002	
28	1.031250049E-002	6.250000297E-003	9.765626174E-008	7.940149903E-001	-6.566685677E+000	2.336071014E+000	-3.508802176E+000	1.696307398E-002	
29	1.000000047E-002	6.250000297E-003	9.765626174E-008	6.658435464E-001	-5.886806965E+000	1.941145062E+000	-3.690945148E+000	1.889615506E-002	
30	9.687500460E-003	6.250000297E-003	9.765626174E-008	5.482489467E-001	-3.415613413E+000	1.442747355E+000	-3.863924503E+000	2.038398199E-002	
31	9.375000445E-003	6.250000297E-003	9.765626174E-008	5.136787891E-001	-3.780338764E-001	8.147930503E-001	-4.007360935E+000	2.102933452E-002	

CAPÍTULO 4 ALGORITMO MATLAB PARA EL SEGUIMIENTO LAGRANGIANO Y RESOLUCIÓN DE LAS ECUACIONES DEL STRETCHING & FOLDING

4.1 DATOS DE PARTIDA

El punto de partida son los ficheros .dat que contienen posiciones de las partículas numeradas según criterios del análisis computacional de Fluent. Así que utilizando funciones internas de Matlab hay que gestionarlos para utilizar el campo de soluciones.

Los ficheros de datos contienen 5000 filas * 11 columnas, es decir 55000 datos numéricos. Los maneja el algoritmo "CoordenadasMalla". Está basado en que cada elemento (z,r) tiene asociada su solución de velocidades, derivadas, volumen de celda; entonces lo que hago es separar esas parejas de (z,r) y usar 'z' separado de 'r'. Formando un vector Z de longitud 1x121 y otro R de 1x41.

Una vez tengo esto yo puedo recorrer con un bucle los vectores dejando uno fijo y variando el otro para obtener la velocidad, derivada...en una coordenada concreta.

El siguiente problema a resolver, es obtener la velocidad, derivada...de una partícula para cualquier coordenada. Sin embargo yo tengo las velocidades...para las celdas concretas cada punto (z,r). La forma de afrontarlo es utilizar la interpolación.

Para interpolar, lo primero que se necesita es saber donde se encuentra la partícula, mejor dicho, entre qué celdas estaba. Para ello se crea la función "campovelocidad".

Para disminuir tiempo de trabajo de la CPU, inicializo a cero las variables que voy a utilizar. Lo cual requiere un esfuerzo previo por parte del programador para tener previstas las variables con las que va a contar y el espacio que va a darle a cada una. Esto además permitiría traducir a Fortran el código de forma más rápida ya que es un lenguaje que sí te obliga a definir las variables desde el principio.

4.2 FUNCIONES

4.2.1 COORDENADASMALLA

En el programa principal cargo el primer fichero de datos. Y asigno las coordenadas 'z' y 'r' a las variables 'ord1' y 'ord2'. La función, recibe estas dos variables de tamaño 4995 (en nuestro caso); como salida da los vectores 'Z' (tamaño 1x121) y 'R'(tamaño 1x41).

El script inicializa el vector Z=zeros(1,40) y el R también. El hecho de poner 40 es sólo a modo de tener algo de espacio reservado en memoria, pero realmente al principio no se sabe el tamaño del vector, conviene poner un número pequeño y una vez conocido el tamaño aumentar. Esto mejora la velocidad sólo cuando tenemos operaciones lógicas 'if'...siendo este un caso.

Aparece definida una variable 'lim' con valor muy cercano a cero (ya que Matlab cuando resta valores que vienen de ficheros .dat no da resultados igual a cero aunque sean iguales los dos números).

La variable 'ord1' se recorre con un bucle for y la operación que se realiza es restar la posición actual a la siguiente, detectando las que son iguales. Así solo manejaré los datos que no estén repetidos y se almacenan en el vector 'Z'. La función hace lo mismo para la 'R'.

4.3.2 CALCULA_DT

Recibe el valor del Strouhal y del Reynolds oscilatorio. Y teniendo en cuenta las características del fluido, en este caso contiene las del agua a 20°C; calcula el paso de tiempo para resolver las ecuaciones con el método numérico que hay programado en el programa principal.

4.3.3 MATRIZ_A

La siguiente función tiene como salida la matriz A. Construida de forma que tenga para cada una de las 4995 filas la posición Z y R. Por eso tiene 3 columnas (filas, posición R, posición Z). Lo que hace la función es recorrer todas las filas para la columna 2 de los datos importados de fluent. Esta columna contiene coordenada z, luego haré lo mismo con la r. Voy detectando cuando la coordenada z coincide con alguna del vector Z(1x121), que recorro con un for anidado. Y guardo la posición donde sucede. Así puedo luego preguntarme que coordenada es la que tengo en la fila 1320 por ejemplo.

4.3.4 CARGA DE FICHEROS

En esta parte comienza a cargar ficheros de datos, tengo 64. Y va almacenando las soluciones de velocidades y derivadas. La necesidad de seguir el recorrido de las partículas durante varios ciclos (64 pasos son un ciclo), se resuelve guardando los datos en las variables durante 50 ciclos. Como el proceso de carga de ficheros es lento un vez hecho el primer ciclo. El programa está preparado para copiar esos datos 50 veces y usarlos más adelante en el seguimiento.

4.3.5 CAMPOVELOCIDAD

Recibe las variables seguimiento, los vectores velocidad y el paso de tiempo en el que estoy. Para ver entre que posiciones del vector Z está la partícula, resta el valor actual de 'z' que ocupa la partícula con cada uno de los valores que conforman el eje Z en la malla del reactor. Se queda con los dos valores mas cercanos. Este método me serviría para procesar cualquier tipo de elemento, incluso datos 3D ya que utiliza sólo coordenadas, nunca formas predeterminadas de elemento etc.

Para considerar las celdas más cercanas, se queda con la posición donde se ha dado el mínimo valor de la resta. Y para ver si el nodo está a la derecha o la izquierda resta la posición de la partícula con la celda de la derecha y la izquierda y comparo los valores. Aquella que resulte en el valor más pequeño será la más cercana. Hace lo mismo con la coordenada r de la partícula.

Una vez tiene su posición bien definida dentro de la malla, utiliza la función de interpolación para obtener las componentes de la velocidad. La función tiene por salida velocidad axial y radial.

4.3.6 CAMPODERIVADAS

Trabaja del mismo modo que la función campo derivadas, lo único distinto es que lo hace para dos partículas; las que forman el elemento línea. Pues para trabajar con la velocidad usa la de cada partícula; sin embargo las derivadas que se deben introducir en la ecuación son las interpoladas entre dos nodos que forman un elemento línea. De ahí que en el código de la función evalúe esta para dos partículas y luego llegue a una solución final en la que intervienen los valores de las derivadas de ambas para su posición en el espacio.

4.3.7 CAMPOVOLUMENES

El volumen de celda, es la importancia que le damos el valor de las magnitudes asociadas a una partícula a la hora de obtener valores promedio. La función trabaja exactamente como lo hace CAMPODERIVADAS, usando los valores de volumen de celda que se han cargado del fichero de datos de Fluent al inicio.

Capítulo 4 Algoritmo Matlab para el seguimiento Lagrangiano y resolución de las ecuaciones del Stretching & Folding

**CAPITULO 5
RESULTADOS. VALORES ALFA
MEDIO PARA DISTINTOS VALORES
DE REYNOLDS OSCILATORIO Y
STROUHAL OSC.**

5.1 $Re_o=10$, $Re_o=60$, $Re_o=120$; Strouhal=1

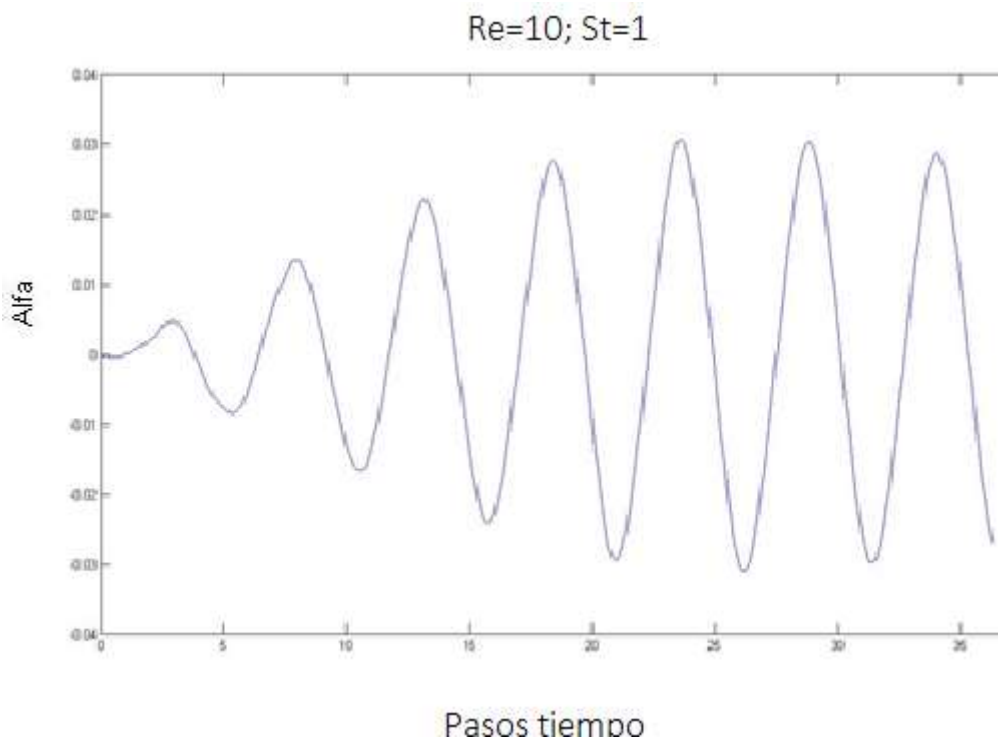


Fig.5.1

La grafica de la fig. 5.1 muestra el valor de alfa para un Reynolds oscilatorio igual a 10 frente al tiempo. Se aprecia el periodo transitorio y otro estacionario de mezclado. Además oscila entre valores muy cercanos, esto se debe al bajo número de Reynolds, que hace que la intensidad de mezclado sea insignificante.

Capítulo 5 RESULTADOS. VALORES ALFA MEDIO PARA DISTINTOS VALORES DE REYNOLDS OSCILATORIO Y STROUHAL OSC.

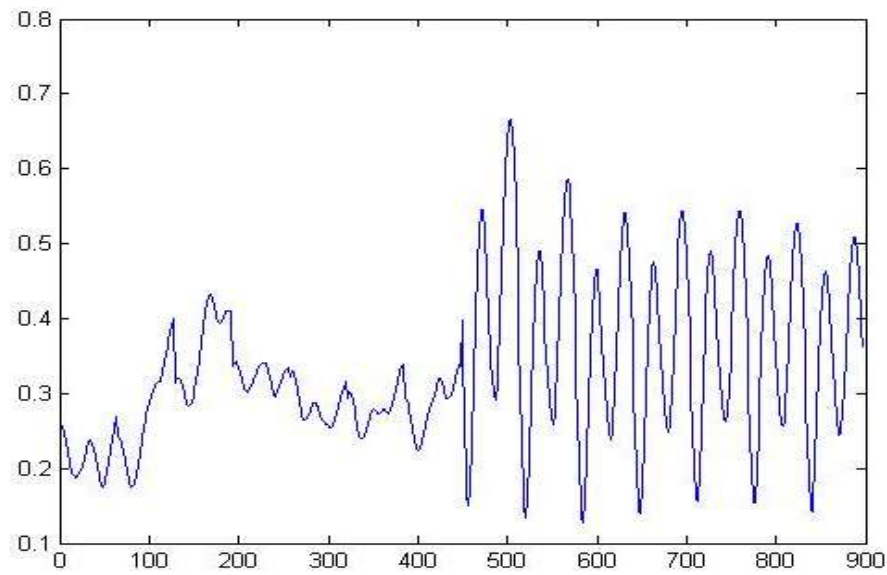


Fig. 5.2

Esta gráfica, fig. 5.2 muestra el valor de alfa para un Reynolds oscilatorio igual a 60 frente al tiempo. Se aprecia el periodo transitorio y otro estacionario de mezclado. Tenemos un transitorio que termina a los 400 pasos de tiempo y se aprecia luego un mezclado algo más caótico. Es en el paso 400 donde comienza la transición entre un periodo transitorio y uno estacionario de mezclado, en cuando a intensidad se refiere.

La figura 5.3 muestra el periodo transitorio que se prolonga algo más allá del paso de tiempo 400. Que para un $Re_o = 120$ son unos 4.2 segundos. Se observa una amplitud del parámetro alfa mucho mayor que antes, lo cual es coherente con el Reynolds osc. Mayor que habla sobre la intensidad del mezclado.

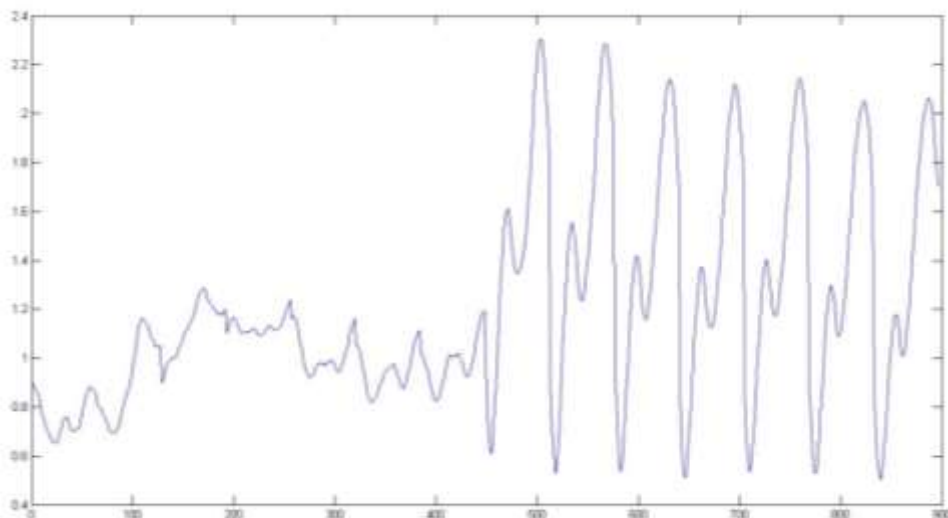


Fig. 5.3

5.2 Strouhal=0.5

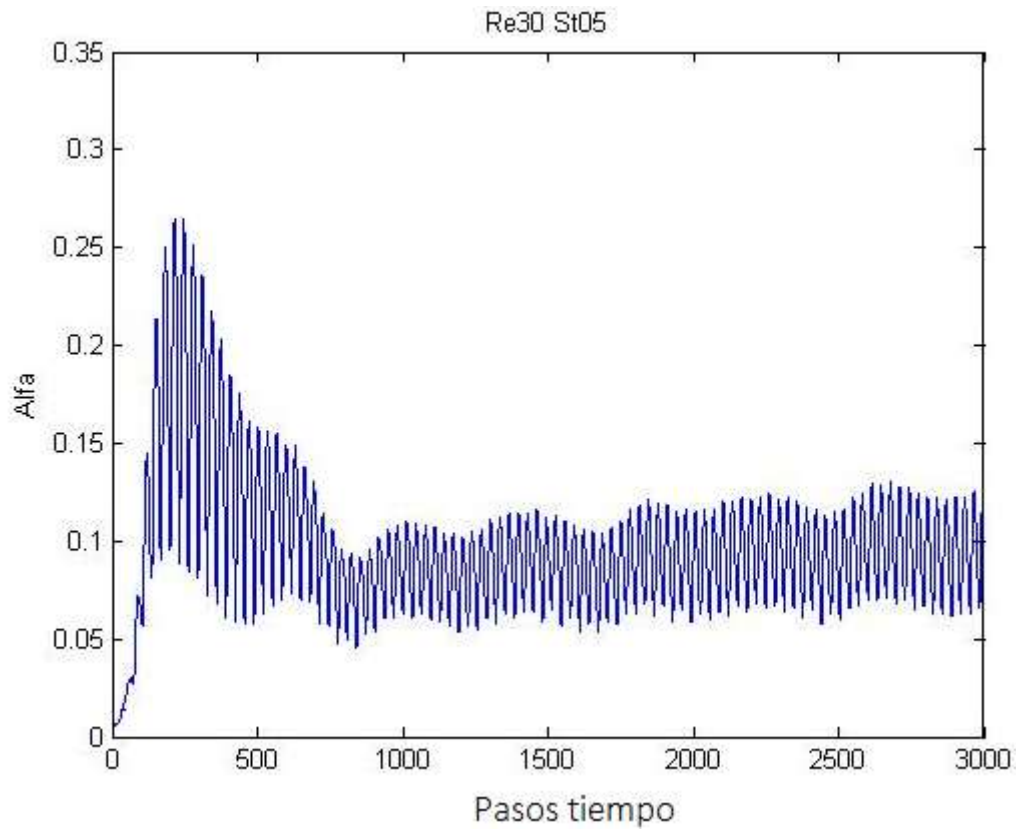
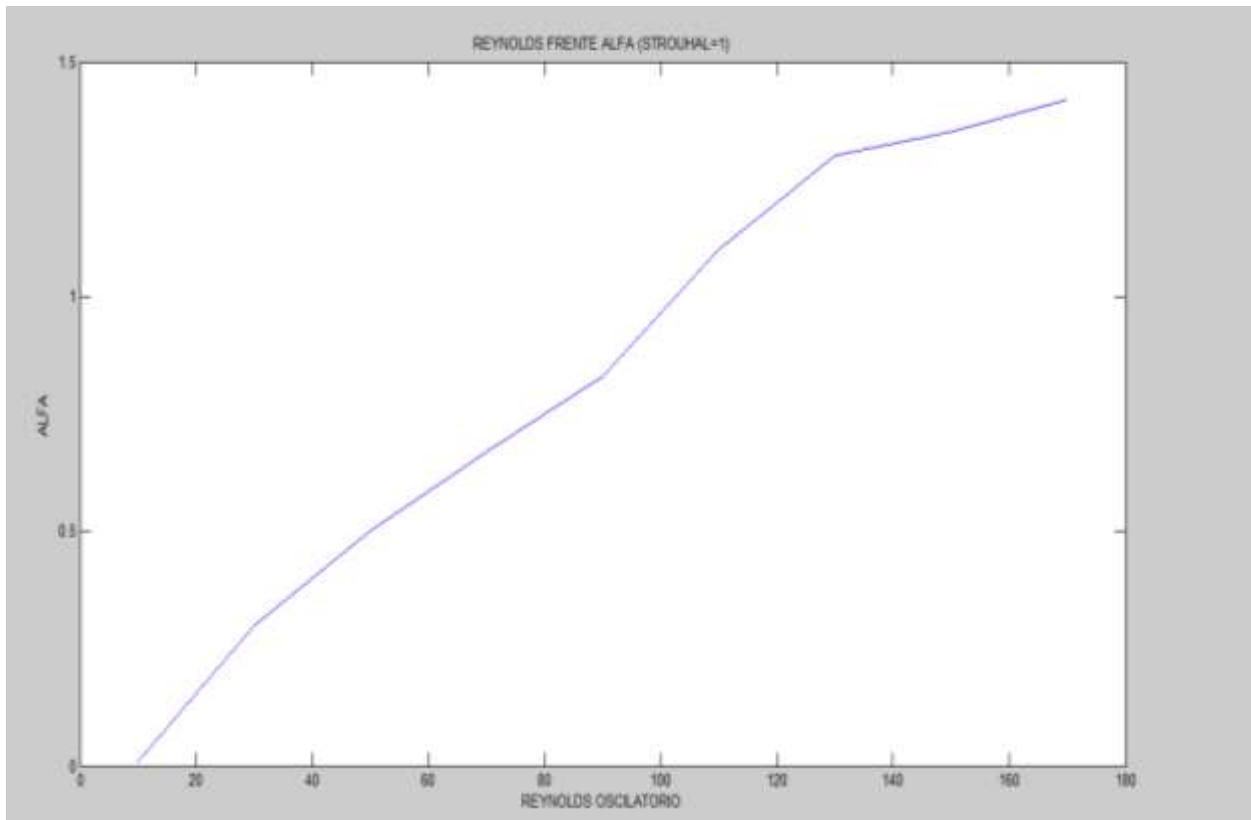


Fig.5.4

Esta figura 5.4 muestra la curva del parámetro alfa, para un $St=0.5$ y un Reynolds oscilatorio=30. Se aprecia un transitorio más largo que con $St=1$. En este caso acaba a los 600 pasos de tiempo (el tamaño de estos pasos se puede calcular introduciendo los valores de St y Re_o en el script `calcula_dt`).

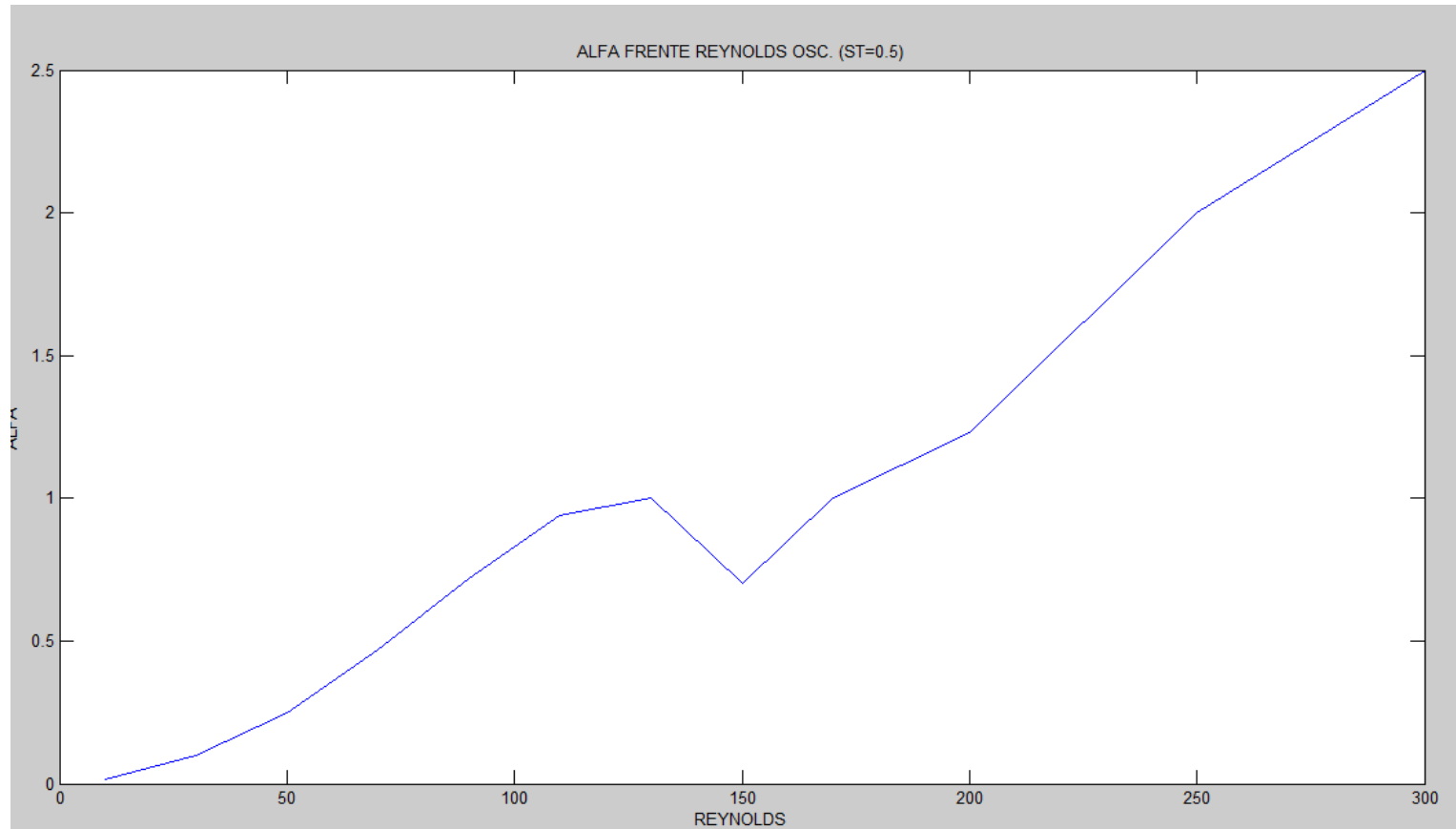
5.3 CURVAS ALFA MEDIO FRENTE A REYNOLDS

5.3.1 PARA STROUHAL, $St=1$

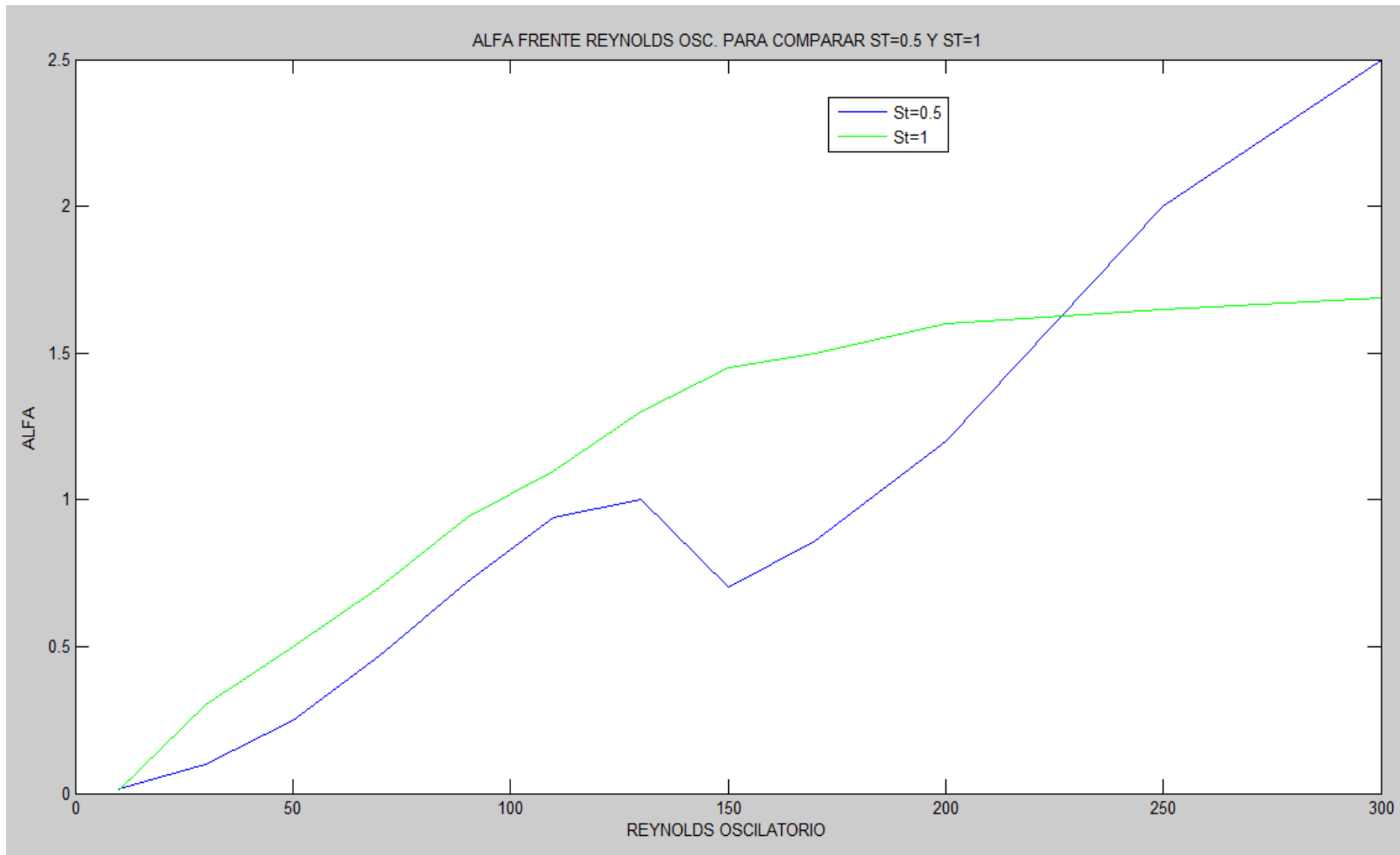


Observamos como a partir de Reynolds oscilatorio 50 empieza el mezclado a ser cada vez mas intenso, y aún sube más cuando se acerca a $Re=110$, pero cuando superamos $Re=170$ la intensidad del mezclado ya aumenta poco a poco.

5.3.2 PARA STROUHAL, $St=0.5$



5.4 COMPARACIÓN ENTRE NÚMEROS DE STROUHAL



5.5 COMENTARIO DE LOS RESULTADOS

El $St=0.5$ despega del $St=1$ para $Re\ osc.= 225$ (aprox.), es decir para un Strouhal menor, si aumenta la intensidad del mezclado lo suficiente, podemos tener un reactor más efectivo a la hora de desarrollar un flujo caótico. El que un flujo sea caótico repercute en la eficiencia alta de transmisión de calor, concentración de trazadores, mejor mezclado, etc.

Para Reynolds menores de 30, en ambos casos se obtiene un alfa casi nulo. Esto es lo que los autores denominan MICROMIXING, micromezclado. Que es mezclado a pequeña escala, donde si se miran los campos vectoriales de velocidad, se aprecian las microestructuras de mezclado.

Sin embargo, esto no entra en el alcance de la presente memoria.

A la vista de los resultados queda demostrado, la importancia de encontrar una solución de compromiso entre Reynolds oscilatorio y Strouhal. Para ello hay que obtener la mejor combinación de frecuencia, amplitud de oscilación y CR(radio de constricción).

CAPITULO 6 CONCLUSIONES

6.1 COMPARACIÓN GRÁFICAS ARTÍCULO M. MACKLEY

La figura 6.1 corresponde a la curva obtenida por M. Mackley en el artículo "The simulation of Stretch rates for the quantitative prediction and mapping of mixing within a canal flow" 1995, E.P.L. ROBERTS & M.R. MACKLEY.

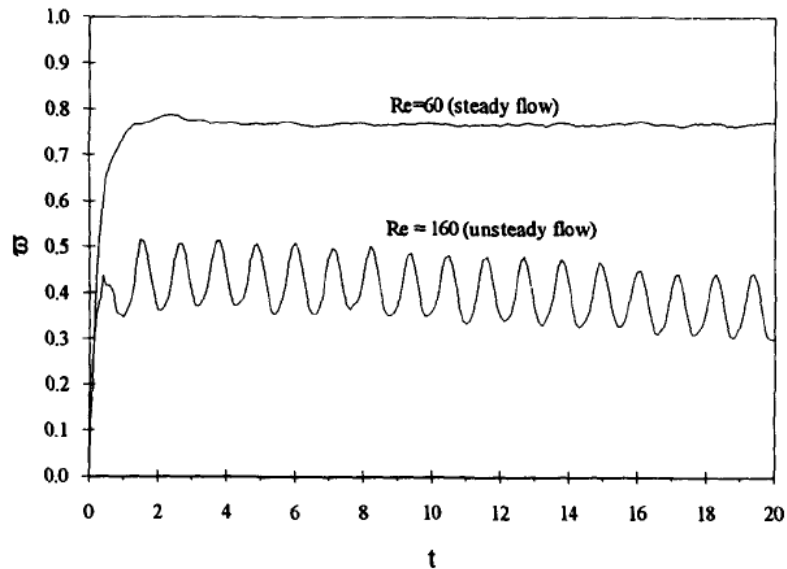


Fig. 6.1

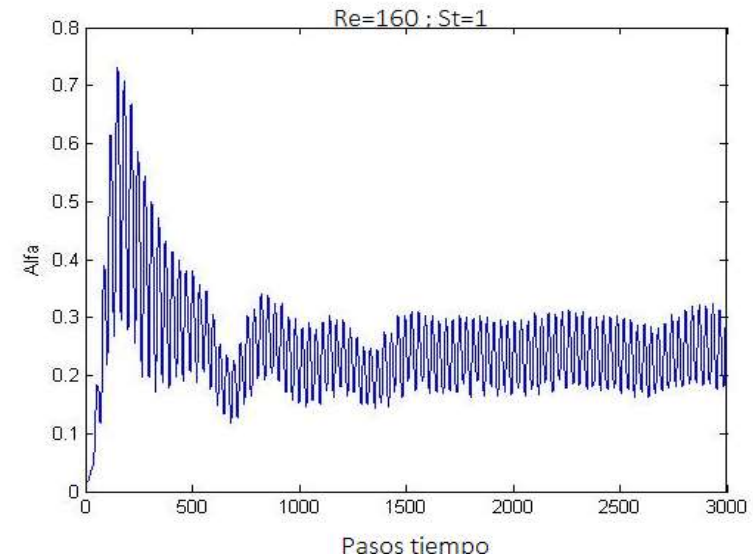


Fig. 6.2

Atendiendo al Re 160 (unsteady flow), apreciamos en la figura 6.2 siguiente una cierta similitud. Esta figura ha sido obtenida tras estudiar los artículos de M. Mackley citados durante la memoria del presente proyecto y usando el algoritmo que el autor de este PFC ha creado. El valor medio que obtiene Mackley es $\alpha=0.4$; en este proyecto, el valor $\alpha=0.3$ aunque no son iguales, el orden de magnitud es coherente.

6.2 TAREAS FUTURAS

Queda pendiente el estudio de las frecuencias naturales de mezclado posibles. Que se aprecian a partir de $Re = 60$, para $St=1$; como vemos en la figura 6.3 Esto debería hacerse con análisis de Fourier, para identificar la frecuencia predominante a partir del paso de tiempo donde la intensidad del mezclado (oscilación del parámetro alfa) sea ya estacionaria.

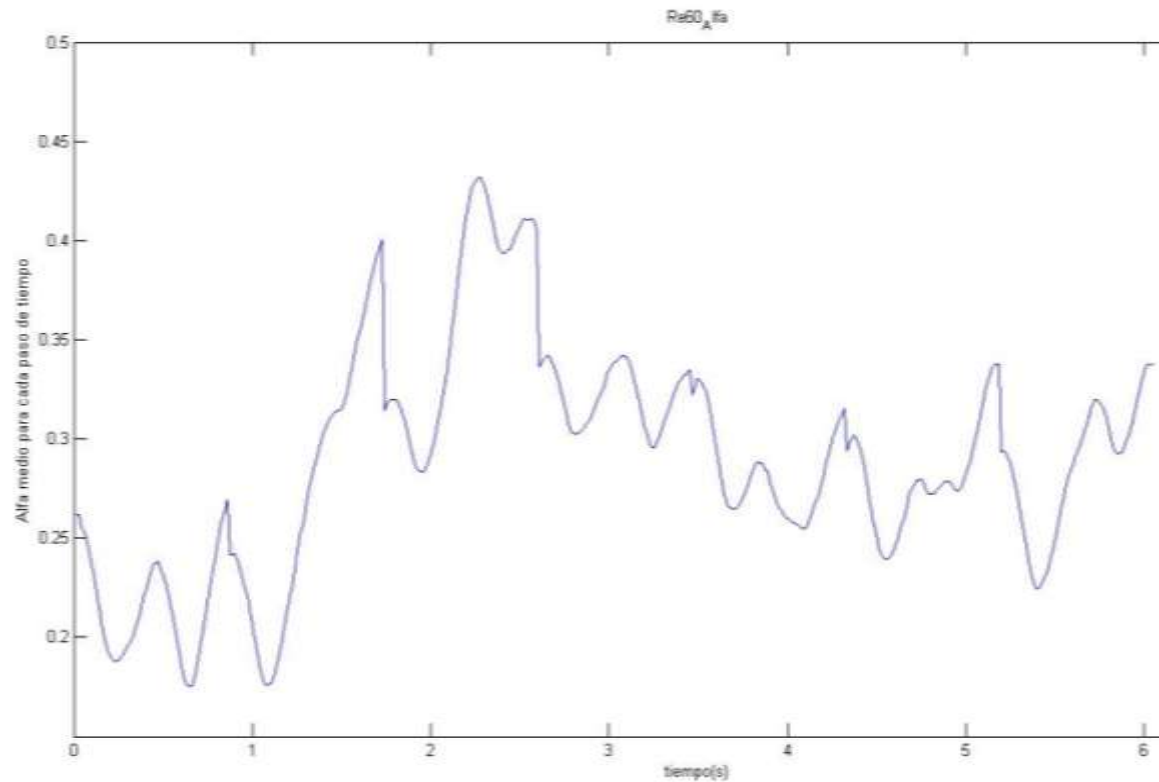


Fig. 6.3

BIBLIOGRAFÍA

Danielson, T.J. & Ottino, J.M. (1990). Structural stability in 2-dimensional flows-Lagrangian and Eulerian turbulence.

Mackley, M.R. (1987) Using oscillatory flow to improve performance Chem.

Khakhar, D.V., Rising & Ottino, J.M. (1986), ANalysis of chaotc mixing in two model systems.

Mackley, M.R. & Neves Saraiva (2002) The quantitative description of fluid mixing using Lagrangian and concentration-based numerical approaches.

Roberts, E.P.R. & Mackley, M.R. (1995) The simulation of stretch rates for the quantitative prediction and mapping of mixing within a cannel flow

MATHWORKS. MATLAB DOCS.

CAPITULO 7 ANEXOS

7.1. FUNCIÓN PRINCIPAL

%Bucle para ejecutar los distintos Reynolds

```
z=0;
for reynolds=30:30:180
z=z+1;
```

**%Funcion para obtener el paso de tiempo que debo aplicar en las ecuaciones
%a la hora de usar el método de elementos finitos**

```
[pasotiempo]=calcula_dt(reynolds,1);
a=0;
Vz=zeros(41,121,1000);
Vr=zeros(41,121,1000);
derivada_vaxial_r =zeros(41,121,1000);
derivada_vradial_r=zeros(41,121,1000);
derivada_vaxial_z=zeros(41,121,1000);
derivada_vradial_z=zeros(41,121,1000);
seguimiento_z1=zeros(41,121,1000);
seguimiento_r1=zeros(41,121,1000);
VlnL1=zeros(38,118,1000);
VlnL1_alt=zeros(38,118,1000);
base=zeros(39,119,1000);
altura=zeros(39,119,1000);
fi1=zeros(38,118,1000);
```

**%Carga el fichero de datos para formar la matriz A, que me permite
%gestionar el campo de velocidad, derivadas.**

```
importado=sprintf('Re10_St1_1.dat');
datos_importados=importdata(importado);
datos=datos_importados.data;
ord1 = sort(datos(:,2));
ord2 = sort(datos(:,3));
[Z,R]=CoordenadasMalla(ord1,ord2);
auxiliar=0;
vol= 9.7656e-008;
grados=[0 pi/4 pi/2 135*pi/180];
A=zeros(length(datos),3);
```

```
for ii=1:length(datos)
    auxiliar=auxiliar+1;
    for jj=1:length(Z)
        if ( abs(datos(ii,2)-Z(jj))<1e-9 )
            break
        end
    end
end
```

```
for kk=1:length(R)
```

```

if( (abs(datos(ii,3)-R(kk)<1e-9)) )
    break
end
end
    A(ii,1:3)=[ii,kk,jj];
end

%En este paso almaceno las velocidades y derivadas
for n=1:64
importado=sprintf('Re%d_St1_%d.dat',reynolds,n);
datos_importados=importdata(importado);
datos=datos_importados.data;
for nod=1:length(datos)
    Vz(A(nod,2),A(nod,3),n)=datos(A(nod,1),10);
    Vr(A(nod,2),A(nod,3),n)=datos(A(nod,1),9);
    derivada_vaxial_r(A(nod,2),A(nod,3),n) = datos(A(nod,1),7);
    derivada_vradial_r(A(nod,2),A(nod,3),n) = datos(A(nod,1),5);
    derivada_vradial_z(A(nod,2),A(nod,3),n) = datos(A(nod,1),6);
    derivada_vaxial_z(A(nod,2),A(nod,3),n) = datos(A(nod,1),8);
end
end

for nn=1:50
    for numero=1:64
n=64*a+numero;
for nod=1:length(datos)
    Vz(A(nod,2),A(nod,3),n)= Vz(A(nod,2),A(nod,3),numero);
    Vr(A(nod,2),A(nod,3),n)= Vr(A(nod,2),A(nod,3),numero);
    derivada_vaxial_r(A(nod,2),A(nod,3),n) =
derivada_vaxial_r(A(nod,2),A(nod,3),numero);
    derivada_vradial_r(A(nod,2),A(nod,3),n) =
derivada_vradial_r(A(nod,2),A(nod,3),numero);
    derivada_vradial_z(A(nod,2),A(nod,3),n) =
derivada_vradial_z(A(nod,2),A(nod,3),numero);
    derivada_vaxial_z(A(nod,2),A(nod,3),n) =
derivada_vaxial_z(A(nod,2),A(nod,3),numero);
end
end
a=a+1;
end

%Aquí resuelvo la posición de la partícula en cada instante para hacer el
%seguimiento desde el punto de vista Lagrangiano

for zetas=1:length(Z)
    for erres=1:length(R)

seguimiento_z1(erres,zetas,1)=Z(zetas);

```

```

seguimiento_r1(erres,zetas,1)=R(erres);
end
end

for zetas=1:length(Z)
for erres=1:length(R)
seguimiento_z1(erres,zetas,2) =
seguimiento_z1(erres,zetas,1)+Vz(erres,zetas,1)*pasotiempo;
seguimiento_r1(erres,zetas,2) =
seguimiento_r1(erres,zetas,1)+Vr(erres,zetas,1)*pasotiempo;

%Esta operación lógica sirve para que cuando una partícula llegue al
%borde vuelva al borde opuesto; ya que trabajo en periodico.
if (seguimiento_z1(erres,zetas,2)>Z(length(Z)))
seguimiento_z1(erres,zetas,2)=seguimiento_z1(erres,zetas,2)-
Z(length(Z));
end

if (seguimiento_z1(erres,zetas,2)<Z(1))

seguimiento_z1(erres,zetas,2)=seguimiento_z1(erres,zetas,2)+Z(length(Z));
end

if (seguimiento_r1(erres,zetas,2)>R(length(R)))
seguimiento_r1(erres,zetas,2)=seguimiento_r1(erres,zetas,2)-
R(length(R));
end
end
end

for numero=2:3200
for zetas=1:length(Z)
for erres=1:length(R)

[vz1,vr1]=campovelocidad(R,Z,Vz,Vr,seguimiento_z1,seguimiento_r1,erres,zetas,
numero);

seguimiento_z1(erres,zetas,numero+1) =
seguimiento_z1(erres,zetas,numero)+vz1*pasotiempo;
seguimiento_r1(erres,zetas,numero+1) =
seguimiento_r1(erres,zetas,numero)+vr1*pasotiempo;
%Esta operación lógica sirve para que cuando una partícula llegue al
%borde vuelva al borde opuesto; ya que trabajo en periodico.
if (seguimiento_z1(erres,zetas,numero)>Z(length(Z)))

seguimiento_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero)-
Z(length(Z));
end

```

```

    if (seguimiento_z1(erres,zetas,numero)<Z(1))

seguimiento_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero)+Z(l
engh(Z));
    end

    if (seguimiento_r1(erres,zetas,numero)>R(length(R)))

seguimiento_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero)-
R(length(R));
    end

    if (seguimiento_r1(erres,zetas,numero)<R(length(R)))

seguimiento_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero)+R(le
ngth(R));
    end
    end
    end
end

    %Formo el elemento línea a cero grados
aux_z=1;
aux_r=0;

for numero=1:3200
    for zetas=3:(length(Z)-3)
        for erres=3:(length(R)-3)

            %Uso la base y la altura para obtener el valor del seno y coseno ya que
Matlab
            %suele dar errores según el cuadrante de la goniométrica.
            base(erres,zetas,numero)=seguimiento_z1(erres+aux_r,zetas+aux_z,numero)-
seguimiento_z1(erres,zetas,numero);

            altura(erres,zetas,numero)=seguimiento_r1(erres+aux_r,zetas+aux_z,numero)-
seguimiento_r1(erres,zetas,numero);

            hipot=sqrt(base(erres,zetas,numero)^2+altura(erres,zetas,numero)^2);

            coseno=base(erres,zetas,numero)/hipot;

            seno=altura(erres,zetas,numero)/hipot;

            %Obtengo las derivadas de la partícula según la posición en la que se
encuentra

            [dvaxr,dvrrar,dvrarz,dvaxz]=campoderivadas(R,Z,derivada_vaxial_r,derivada_vra
dial_r,deriv

```

```
ada_vradial_z,derivada_vaxial_z,seguimiento_z1,seguimiento_r1,erres,zetas,n
umero,aux_r,aux_z);
```

```
%estas son las ecuaciones propuestas por Mackley, pero ya escritas usando
elementos %finitos. Resolviendo para las soluciones discretas de las derivadas
obtenidas tras %resolver la posición de la partícula, según Lagrange.
```

```
Vfi1= coseno^2*dvrax + seno*coseno*( dvrar - dvaxz ) - seno^2*dvaxr;
fi1(erres,zetas,numero)= Vfi1*pasotiempo;
```

```
VlnL1_alt(erres,zetas,numero) = cos(fi1(erres,zetas,numero))^2*dvrax +
sin(fi1(erres,zetas,numero))*cos(fi1(erres,zetas,numero))*( dvrar + dvaxz ) +
sin(fi1(erres,zetas,numero))^2*dvaxr ;
```

```
VlnL1(erres,zetas,numero) = coseno^2*dvrax + seno*coseno*( dvrar + dvaxz )
+ seno^2*dvaxr ;
```

```
end
end
end
```

```
%Grafica del alfa definido por Mackley frente al tiempo
```

```
for i=1:3200
    B(i)=mean(mean(abs(VlnL1(:,i,i))));
end
```

```
paso=calcula_dt(reynolds,1);
t=(1:3200);
figure(z)
plot(t,B)
title('Mezclado transitorio+Estacionario')
xlabel('tiempo')
ylabel('alfa')
```

```
%Almacena los datos como fichero de texto
```

```
if(reynolds==30)
    save Re30_I
end
```

```
if(reynolds==60)
    save Re60_I
end
```

```
if(reynolds==90)
    save Re90_I
end
```

```
if(reynolds==120)
    save Re120_I
end
```

```
if(reynolds==100)
```

```

    save Re100_I
end

if(reynolds==10)
    save Re10_I
end

if(reynolds==150)
    save Re150_I
end

if(reynolds==180)
    save Re180_I
end

end

```

7.2 FUNCION PASO DE TIEMPO

```

function[pasotiempo]=calcula_dt(reo,st)

%se usa así:
%dt=calcula_dt(reo,st)

rho=998.2; %kg/m3 [Fluent]
mu=0.001003; %kg/m-s [Fluent]
D=25e-3;
N=192; %nro puntos en un semiciclo [0.5 Hz 768] [2 Hz 192]
x=D./(4*pi*st);
f=reo*mu/(2*pi*x*rho*D);
pasotiempo=1/(2*f*N);

```


7.3 FUNCION COORDENADAS DE MALLA

```
function [Z,R]=CoordenadasMalla(ord1,ord2)
```

```
%Para ello inicializo el vector Z=zeros(1,40) y el R también. El hecho de poner
40 es sólo a modo
%de tener algo de espacio reservado en memoria, pero realmente al principio
no se sabe el tamaño del
%vector, conviene poner un número pequeño y una vez conocido el tamaño
aumentar. Esto mejora la velocidad
%sólo cuando tenemos operaciones lógicas 'if'...siendo este un caso.
```

```
tam1 = length(ord1)-1;
tam2 = length(ord2)-1;
f=0;
g=0;
Z=zeros(1,20);
R=zeros(1,20);
```

```
for i=1:tam1
```

```
    %He definido una variable 'lim' con valor muy cercano a cero (ya que Matlab
    %cuando resta valores que vienen de ficheros .dat no da resultados igual a
    %cero aunque sean iguales los dos números).
```

```
    lim = abs( ord1(i)-ord1(i+1) );
    if (lim>1e-6 )
        f = f+1;
        Z(f) = ord1(i);
    end
end
f=f+1;
Z(f) = ord1(i+1);
```

```
for i=1:tam2
```

```
    lim = abs( ord2(i)-ord2(i+1) );
    if (lim>1e-6 )
        g = g+1;
        R(g) = ord2(i);
    end
end
g = g+1;
R(g) = ord2(i+1);
```

```
end
```

7.4 FUNCION CAMPO DE VELOCIDAD

%Campo de velocidad

function

```
[vz1, vr1]=campovelocidad(R,Z,Vz,Vr,seguimiento_z1,seguimiento_r1,erres,zetas,numero)
```

% Uso la variable seg_z1 para no escribir sobre la importante que es **%seguimiento_z1.**

```
seg_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero);
resta=abs(Z(:)-seg_z1(erres,zetas,numero));
[rrr,situacion_z11]=min(resta(:));
```

% Aquí evito que las partículas situadas en los bordes del reactor den **%problemas.**

```
if ( situacion_z11==1 )
    situacion_z12=situacion_z11+1;
elseif ( situacion_z11==length(Z) )
    situacion_z12=situacion_z11-1;
```

% Cuando la partícula no esté en los bordes, lo que hago es ver si el **%elemento más cercano está a la derecha o la izquierda del elemento **%donde sé que se encuentra la partícula****

```
elseif ( situacion_z11~=1 || situacion_z11~=length(Z) )
    resta1=Z(situacion_z11+1)-seg_z1(erres,zetas,numero);
    resta2=Z(situacion_z11-1)-seg_z1(erres,zetas,numero);
```

```
    if(resta1<resta2)
        situacion_z12=situacion_z11+1;
    elseif(resta1>resta2)
        situacion_z12=situacion_z11-1;
    end
```

end

```
seg_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero);
resta=abs(R(:)-seg_r1(erres,zetas,numero));
[rrr,situacion_r11]=min(resta(:));
```

```
if ( situacion_r11==1 )
    situacion_r12=situacion_r11+1;
elseif ( situacion_r11==length(R) )
    situacion_r12=situacion_r11-1;
```

```
elseif ( situacion_r11~=1 || situacion_r11~=length(R) )
    resta1=R(situacion_r11+1)-seg_r1(erres,zetas,numero);
    resta2=R(situacion_r11-1)-seg_r1(erres,zetas,numero);

    if(resta1<resta2)
        situacion_r12=situacion_r11+1;
    elseif(resta1>resta2)
        situacion_r12=situacion_r11-1;
    end
end
```

% Uso los valores *situacion_z12* y *situacion_r12* para introducir las funciones
% de interpolación.

```
vr1=(
Vr(situacion_r11,situacion_z11,numero)+Vr(situacion_r12,situacion_z12,numero)
)/2;
vz1=(
Vz(situacion_r11,situacion_z11,numero)+Vz(situacion_r12,situacion_z12,numero)
)/2;

end
```

7.5 FUNCION DERIVADAS

% Derivadas

```
function
[dvaxr,dvraz,dvaxz]=campoderivadas(R,Z,derivada_vaxial_r,derivada_vradial_r,derivada_vradial_z,derivada_vaxial_z,seguimiento_z1,seguimiento_r1,erres,zetas,numero,aux_r,aux_z)

if(numero==1)
    numero=2;
end

if (seguimiento_z1(erres,zetas,numero)>Z(length(Z)-2))
    seg_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero)-Z(length(Z)-2);
```

end

```
seg_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero);
resta=abs(Z(:)-seg_z1(erres,zetas,numero));
[rrr,situacion_z11]=min(resta(:));
```

```
if ( situacion_z11==1 )
    situacion_z12=situacion_z11+1;
elseif ( situacion_z11==length(Z) )
    situacion_z12=situacion_z11-1;
elseif ( situacion_z11~=1 || situacion_z11~=length(Z) )
    resta1=Z(situacion_z11+1)-seg_z1(erres,zetas,numero);
    resta2=Z(situacion_z11-1)-seg_z1(erres,zetas,numero);

    if(resta1<resta2)
        situacion_z12=situacion_z11+1;
    elseif(resta1>resta2)
        situacion_z12=situacion_z11-1;
    end
end
```

```
if (seguimiento_r1(erres,zetas,numero)>R(length(R)-2))
    seg_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero)-
R(length(R)-2);
end
seg_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero);
resta=abs(R(:)-seg_r1(erres,zetas,numero));
[rrr,situacion_r11]=min(resta(:));
```

```
if ( situacion_r11==1 )
    situacion_r12=situacion_r11+1;
elseif ( situacion_r11==length(R) )
    situacion_r12=situacion_r11-1;
elseif ( situacion_r11~=1 || situacion_r11~=length(R) )
    resta1=R(situacion_r11+1)-seg_r1(erres,zetas,numero);
    resta2=R(situacion_r11-1)-seg_r1(erres,zetas,numero);

    if(resta1<resta2)
        situacion_r12=situacion_r11+1;
    elseif(resta1>resta2)
        situacion_r12=situacion_r11-1;
    end
end
```

%Uso los valores situacion_z12 y situacion_r12 para introducir las funciones
%de interpolación.

```
if(situacion_r11>41)
    situacion_r11=situacion_r11-1;
end
```

```

if(situacion_z12>41)
    situacion_z12=situacion_z12-1;
end
nm=numero;
dvaxr1=(
derivada_vaxial_r(situacion_r11,situacion_z11,nm)+derivada_vaxial_r(situacion
_r12,situacion_z12,nm) )/2;
dvrr1=(
derivada_vradial_r(situacion_r11,situacion_z11,nm)+derivada_vradial_r(situaci
on_r12,situacion_z12,nm) )/2;
dvraz1=(
derivada_vradial_r(situacion_r11,situacion_z11,nm)+derivada_vradial_r(situaci
on_r12,situacion_z12,nm) )/2;
dvaxz1=(
derivada_vaxial_z(situacion_r11,situacion_z11,nm)+derivada_vaxial_z(situacio
n_r12,situacion_z12,nm) )/2;

```

%Ahora vamos con la particula dos

```

if (seguimiento_z1(erres+aux_r,zetas+aux_z,numero)>Z(length(Z)-2))

seg_z2(erres,zetas,numero)=seguimiento_z1(erres+aux_r,zetas+aux_z,numero
)-Z(length(Z)-2);
end
seg_z2(erres,zetas,numero)=seguimiento_z1(erres+aux_r,zetas+aux_z,numero
);
resta=abs(Z(:)-seg_z2(erres,zetas,numero));
[rrr,situacion_z21]=min(resta(:));

```

```

if ( situacion_z21==1 )
    situacion_z22=situacion_z21+1;
elseif ( situacion_z21==length(Z) )
    situacion_z22=situacion_z21-1;
elseif ( situacion_z21~=1 || situacion_z21~=length(Z) )
    resta1=Z(situacion_z21+1)-seg_z2(erres,zetas,numero);
    resta2=Z(situacion_z21-1)-seg_z2(erres,zetas,numero);

```

```

    if(resta1<resta2)
        situacion_z22=situacion_z21+1;
    elseif(resta1>resta2)
        situacion_z22=situacion_z21-1;
    end
end

```

```

if (seguimiento_r1(erres+aux_r,zetas+aux_z,numero)>R(length(R)-2))

```

```

seg_r2(erres,zetas,numero)=seguimiento_r1(erres+aux_r,zetas+aux_z,numero)
-R(length(R)-2);

```

```

end
seg_r2(erres,zetas,numero)=seguimiento_r1(erres+aux_r,zetas+aux_z,numero)
;
resta=abs(R(:)-seguimiento_r1(erres+aux_r,zetas+aux_z,numero));
[rrr,situacion_r21]=min(resta(:));

```

```

if (situacion_r21==1 || situacion_r21==length(R) )
    situacion_r22=situacion_r21+1;
elseif ( situacion_r21==length(R) )
    situacion_r22=situacion_r21-1;
elseif ( situacion_r21~=1 || situacion_r21~=length(R) )
    resta1=R(situacion_r21+1)-seg_r2(erres,zetas,numero);
    resta2=R(situacion_r21-1)-seg_r2(erres,zetas,numero);

```

```

        if(resta1<resta2)
            situacion_r22=situacion_r21+1;
        elseif(resta1>resta2)
            situacion_r22=situacion_r21-1;
        end
end

```

```
end
```

% Uso los valores **situacion_z2** y **situacion_r2** para introducir las funciones
% de interpolación.

```

if(situacion_r21>41)
    situacion_r21=situacion_r21-1;
end
if(situacion_z21>121)
    situacion_z21=situacion_z21-1;
end
if(situacion_r22>41)
    situacion_r22=situacion_r22-1;
end
if(situacion_z22>121)
    situacion_z22=situacion_z22-1;
end
end

```

```

dvaxr2=( derivada_vaxial_r(situacion_r21,situacion_z21,nm)+
derivada_vaxial_r(situacion_r22,situacion_z22,nm) )/2;
dvrar2=( derivada_vradial_r(situacion_r21,situacion_z21,nm)+
derivada_vradial_r(situacion_r22,situacion_z22,nm) )/2;
dvraz2=( derivada_vradial_z(situacion_r21,situacion_z21,nm)+
derivada_vradial_z(situacion_r22,situacion_z22,nm) )/2;
dvaxz2=(
derivada_vaxial_z(situacion_r11,situacion_z11,nm)+derivada_vaxial_z(situacio
n_r12,situacion_z12,nm) )/2;

```

% El valor de las derivadas interpolado

```
dvaxr=(dvaxr2+dvaxr1)/2;
```

```
dvrar=(dvrar2+dvrar1)/2;
dvrar=(dvrar2+dvrar1)/2;
dvaxz=(dvaxz2+dvaxz1)/2;
end
```

7.6 FUNCION VOLUMEN

```
%Volumen
```

```
function
[volumencelda]=campovolumenes(R,Z,volumen,seguimiento_z1,seguimiento_r
1,erres,zetas,numero,aux_r,aux_z)
if(numero==1)
    numero=2;
end
if (seguimiento_z1(erres,zetas,numero)>Z(length(Z)-2))
    seg_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero)-
Z(length(Z)-2);
end

seg_z1(erres,zetas,numero)=seguimiento_z1(erres,zetas,numero);
resta=abs(Z(:)-seg_z1(erres,zetas,numero));
[rrr,situacion_z11]=min(resta(:));

if ( situacion_z11==1 )
    situacion_z12=situacion_z11+1;
elseif ( situacion_z11==length(Z) )
    situacion_z12=situacion_z11-1;
elseif ( situacion_z11~=1 || situacion_z11~=length(Z) )
    resta1=Z(situacion_z11+1)-seg_z1(erres,zetas,numero);
    resta2=Z(situacion_z11-1)-seg_z1(erres,zetas,numero);

    if(resta1<resta2)
        situacion_z12=situacion_z11+1;
    elseif(resta1>resta2)
        situacion_z12=situacion_z11-1;
    end
end

if (seguimiento_r1(erres,zetas,numero)>R(length(R)-2))
```

```

    seg_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero)-
R(length(R)-2);
end
seg_r1(erres,zetas,numero)=seguimiento_r1(erres,zetas,numero);
resta=abs(R(:)-seg_r1(erres,zetas,numero));
[rrr,situacion_r11]=min(resta(:));

if ( situacion_r11==1 )
    situacion_r12=situacion_r11+1;
elseif ( situacion_r11==length(R) )
    situacion_r12=situacion_r11-1;
elseif ( situacion_r11~=1 || situacion_r11~=length(R) )
    resta1=R(situacion_r11+1)-seg_r1(erres,zetas,numero);
    resta2=R(situacion_r11-1)-seg_r1(erres,zetas,numero);

    if(resta1<resta2)
        situacion_r12=situacion_r11+1;
    elseif(resta1>resta2)
        situacion_r12=situacion_r11-1;
    end
end

%Usa los valores situacion_z12 y situacion_r12 para introducir las funciones
%de interpolación.
if(situacion_r11>41)
    situacion_r11=situacion_r11-1;
end
if(situacion_z12>41)
    situacion_z12=situacion_z12-1;
end
nm=numero;
volumen1=(volumen(situacion_r11,situacion_z11,nm) +
volumen(situacion_r12,situacion_z12,nm)) / 2;

%Ahora vamos con la particula dos
if (seguimiento_z1(erres+aux_r,zetas+aux_z,numero)>Z(length(Z)-2))

seg_z2(erres,zetas,numero)=seguimiento_z1(erres+aux_r,zetas+aux_z,numero
)-Z(length(Z)-2);
end
seg_z2(erres,zetas,numero)=seguimiento_z1(erres+aux_r,zetas+aux_z,numero
);
resta=abs(Z(:)-seg_z2(erres,zetas,numero));
[rrr,situacion_z21]=min(resta(:));

if ( situacion_z21==1 )
    situacion_z22=situacion_z21+1;
elseif ( situacion_z21==length(Z) )

```



```

    situacion_z22=situacion_z21-1;
elseif ( situacion_z21~=1 || situacion_z21~=length(Z) )
    resta1=Z(situacion_z21+1)-seg_z2(erres,zetas,numero);
    resta2=Z(situacion_z21-1)-seg_z2(erres,zetas,numero);

    if(resta1<resta2)
        situacion_z22=situacion_z21+1;
    elseif(resta1>resta2)
        situacion_z22=situacion_z21-1;
    end
end

if (seguimiento_r1(erres+aux_r,zetas+aux_z,numero)>R(length(R)-2))

seg_r2(erres,zetas,numero)=seguimiento_r1(erres+aux_r,zetas+aux_z,numero)
-R(length(R)-2);
end
seg_r2(erres,zetas,numero)=seguimiento_r1(erres+aux_r,zetas+aux_z,numero)
;
resta=abs(R(:)-seguimiento_r1(erres+aux_r,zetas+aux_z,numero));
[rrr,situacion_r21]=min(resta(:));

if (situacion_r21==1 || situacion_r21==length(R) )
    situacion_r22=situacion_r21+1;
elseif ( situacion_r21==length(R) )
    situacion_r22=situacion_r21-1;
elseif ( situacion_r21~=1 || situacion_r21~=length(R) )
    resta1=R(situacion_r21+1)-seg_r2(erres,zetas,numero);
    resta2=R(situacion_r21-1)-seg_r2(erres,zetas,numero);

    if(resta1<resta2)
        situacion_r22=situacion_r21+1;
    elseif(resta1>resta2)
        situacion_r22=situacion_r21-1;
    end
end

end

%Usa los valores situacion_z2 y situacion_r2 para introducir las funciones
%de interpolación.
if(situacion_r21>41)
    situacion_r21=situacion_r21-1;
end
if(situacion_z21>121)
    situacion_z21=situacion_z21-1;
end
if(situacion_r22>41)
    situacion_r22=situacion_r22-1;
end
end

```

```

if(situacion_z22>121)
    situacion_z22=situacion_z22-1;
end

volumen2=(volumen(situacion_r21,situacion_z21,nm) +
volumen(situacion_r22,situacion_z22,nm)) / 2;

%El valor de las derivadas interpolado

volumencelda=(volumen1+volumen2) / 2;
end

```

7.7 FUNCION ELEMENTOS LÍNEA 45° Y 90°

```

for reynolds=30:30:180
[pasotiempo]=calcula_dt(reynolds,1);

%Cargo los ficheros que he guardado para el elemento línea cero grados.
%Puesto que hay datos que voy a aprovechar.
if(reynolds==30)
    load Re30
end

if(reynolds==60)
    load Re60
end

if(reynolds==90)
    load Re90
end

if(reynolds==120)
    load Re120
end

if(reynolds==100)
    load Re100
end

if(reynolds==10)
    load Re10
end

if(reynolds==150)

```

```

    load Re150
end

if(reynolds==180)
    load Re180
end

%Este bucle sirve para calcular los otros dos elementos restantes
for elemento=2:3
    grados=[0 pi/4 pi/2];

    %Inicializo las variables necesarias para aumentar la eficiencia
    VlnL1=zeros(38,118,1000);
    VlnL1_alt=zeros(38,118,1000);
    VlnL1_alt2=zeros(38,118,1000);
    base=zeros(39,119,1000);
    altura=zeros(39,119,1000);
    fi1=zeros(38,118,1000);
    fi2=zeros(38,118,1000);

    %Aquí según el elemento que quiera formar le doy un valor u otro a las
    %variables auxiliares. El elemento a 45° por ejemplo lo formo, para el caso
    %de una malla uniforme, tomando de
    seguimiento_z1(erres+aux_r,zetas+aux_z,numero)
    %la posición del siguiente nodo en z y en r así consigo 45°.

    switch elemento

        case 1
            aux_z=1;
            aux_r=0;

        case 2
            aux_z=1;
            aux_r=1;

        case 3
            aux_z=0;
            aux_r=1;

    end

    %Aquí resuelvo las ecuaciones de Mackley para elemento línea a 45° y 90°
    for numero=1:3200
        for zetas=3:(length(Z)-3)
            for erres=3:(length(R)-3)

                base(erres,zetas,numero)=seguimiento_z1(erres+aux_r,zetas+aux_z,numero)-
                seguimiento_z1(erres,zetas,numero);
            end
        end
    end
end

```

```

altura(erres,zetas,numero)=seguimiento_r1(erres+aux_r,zetas+aux_z,numero)-
seguimiento_r1(erres,zetas,numero);
hipot=sqrt(base(erres,zetas,numero)^2+altura(erres,zetas,numero)^2);
coseno=base(erres,zetas,numero)/hipot;
seno=altura(erres,zetas,numero)/hipot;

```

%Obtengo las derivadas de la partícula según la posición en la que se encuentra

```

[dvaxr,dvraz,dvaxz]=campoderivadas(R,Z,derivada_vaxial_r,derivada_vra
dial_r,derivada_vradial_z,derivada_vaxial_z,seguimiento_z1,seguimiento_r1,err
es,zetas,numero,aux_r,aux_z);

```

```

Vfi1= coseno^2*dvraz + seno*coseno*( dvraz - dvaxz ) - seno^2*dvaxr;
fi1(erres,zetas,numero)= Vfi1*pasotiempo;
fi2(erres,zetas,numero)= Vfi1*pasotiempo-gradados(elemento);

```

```

VlnL1_alt(erres,zetas,numero) = cos(fi1(erres,zetas,numero))^2*dvraz +
sin(fi1(erres,zetas,numero))*cos(fi1(erres,zetas,numero))*( dvraz + dvaxz ) +
sin(fi1(erres,zetas,numero))^2*dvaxr ;
VlnL1(erres,zetas,numero) = coseno^2*dvraz + seno*coseno*( dvraz + dvaxz )
+ seno^2*dvaxr ;
VlnL1_alt2(erres,zetas,numero) = cos(fi2(erres,zetas,numero))^2*dvraz +
sin(fi2(erres,zetas,numero))*cos(fi2(erres,zetas,numero))*( dvraz + dvaxz ) +
sin(fi2(erres,zetas,numero))^2*dvaxr ;

```

```

end
end
end

```

%Almaceno los resultados en ficheros de texto. El II y el III corresponden %la elemento línea 45° y 90°.

```

if(reynolds==30)

    if elemento==3
        save Re30_III
    end

    if elemento==2
        save Re30_II
    end
end

if(reynolds==60)

```

```
if elemento==3  
  save Re60_III  
end
```

```
if elemento==2  
  save Re60_II  
end  
end
```

```
if(reynolds==90)  
  
  if elemento==3  
    save Re90_III  
  end  
  
  if elemento==2  
    save Re90_II  
  end  
end
```

```
if(reynolds==120)  
  
  if elemento==3  
    save Re120_III  
  end  
  
  if elemento==2  
    save Re120_II  
  end  
end
```

```
if(reynolds==100)  
  
  if elemento==3  
    save Re100_III  
  end  
  
  if elemento==2  
    save Re100_II  
  end  
end
```

```
if(reynolds==10)  
  
  if elemento==3  
    save Re10_III
```

```
end

if elemento==2
save Re10_II
end
end

if(reynolds==150)

if elemento==3
save Re150_III
end

if elemento==2
save Re150_II
end
end

if(reynolds==180)

if elemento==3
save Re180_III
end

if elemento==2
save Re180_II
end
end

end

end
```

7.8 JOURNALS

```
f=1;%Hz
x=7.5; %mm
if f<1
f=f*10;
cadena=sprintf('AUX_x%dmm_f0%dHz.jou',round(x),round(f));
```

```

profile=sprintf('AUX_x%dmm_f0%dHz',round(x),round(f));
else
cadena=sprintf('AUX_x%dmm_f%dHz.jou',round(x),round(f));
profile=sprintf('AUX_x%dmm_f%dHz',round(x),round(f));
end

fid = fopen(cadena,'w');
ff=sprintf('rcd %s.cas.gz',profile);
fwrite(fid,ff);
fprintf(fid,'\n');
fwrite(fid,'/define/user-defined/interpreted-functions "perfil.dat" "cpp" 10000 no');
fprintf(fid,'\n');
fwrite(fid,'/define/boundary-conditions/velocity-inlet');
fprintf(fid,'\n');
fwrite(fid,'inlet.1');
fprintf(fid,'\n');
fwrite(fid,'no');
fprintf(fid,'\n');
fwrite(fid,'no');
fprintf(fid,'\n');
fwrite(fid,'yes');
fprintf(fid,'\n');
fwrite(fid,'yes');
fprintf(fid,'\n');
fwrite(fid,'yes');
fprintf(fid,'\n');
fwrite(fid,'yes');
fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fwrite(fid,'/solve/set/reporting-interval 10');
fprintf(fid,'\n');
fwrite(fid,'/solve/set/time-step 0.0052083333');
fprintf(fid,'\n');
fwrite(fid,'/solve/dual-time/iterate 768 200');
fprintf(fid,'\n');
for i=1:384
fwrite(fid,'/solve/dual-time/iterate 1 200');
fprintf(fid,'\n');
fwrite(fid,'/file/write-profile');
fprintf(fid,'\n');
perfil=sprintf('perfil_%d.dat',i);
fwrite(fid,perfil);
fprintf(fid,'\n');
fwrite(fid,'interior-12');
fprintf(fid,'\n');
fprintf(fid,'\n');
fwrite(fid,'x-velocity');

```

```

fprintf(fid, '\n');
fwrite(fid, 'y-velocity');
fprintf(fid, '\n');
fwrite(fid, 'z-velocity');
fprintf(fid, '\n');
fwrite(fid, 'q');
fprintf(fid, '\n');
end
%condiciones de contorno entrada/salida

EXPORTAR CASOS
clear all
close all

rho=998.2; %kg/m3 [Fluent]
mu=0.001003; %kg/m-s [Fluent]
ren=50;
reo=500;
st=1;
%
D=5e-3; %m
A=pi*D^2/4; %BUSCAR EN FLUENT (REPORT/SURFACE INTEGRALS) %m
delta=0; %time-step desde el que se inicia el caso
N=192/4; %nro puntos en un semiciclo [0.5 Hz 768] [2 Hz 192]
it=500; %iteraciones per-time-step

%c=[N/2 N/2 N/2 N/32]; %control time-steps en cada ciclo
%c=[N/2 N/2 N/192]; %control time-steps en cada ciclo
c=[N/2 N/2 1]; %control time-steps en cada ciclo
C=length(c); %nro de ciclos
n=2*N./c; %nro control points en cada ciclo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

un=ren*mu/(rho*D);

for k=1:length(st)
x0(k)=D./(4*pi*st(k));
%x0(k)=0.002 %mm
for h=1:length(reo)
f(k,h)=reo(h)*mu/(2*pi*x0(k)*rho*D);
%f(k,h)=8; %Hz

dt=1/(2*f(k,h)*N);
dur=C/f(k,h);
t=dt:dt:dur; %empezamos en 0+dt
u = un + 2*pi*f(k,h)*x0(k)*cos(2*pi*f(k,h)*t);
uini=un+2*pi*f(k,h)*x0(k);
plot(0,uini*1000,'r.')
hold on

```



```

plot(t,u*1000,'k.')

p=1;
pos=[0 0];
for i=1:length(n)
for j=1:n(i)
    p=p+1;
    pos(p)=pos(p-1)+c(i);
    cyc(p)=i;
end
%for j=1:n(i)
%ang(j)=pos(j+1)/pos(

end

for i=1:length(pos)-1
    pos(i)=pos(i+1);
    plot(t(pos(i)),u(pos(i))*1000,'ro')
end
x=pos+delta;
clear pos
pos(1)=0;
for i=2:length(x)
    pos(i)=x(i-1);
end

for j=2:length(pos)
max(j)=cyc(j)*2*N;
prev(j)=(cyc(j)-1)*2*N;
ang(j)=abs(pos(j)-prev(j))/(max(j)-prev(j));
end
cyc(1)=1;

%write_udf(st(k),reo(h),un,x0(k),f(h));
write_jou_test3(pos,cyc,ang,st(k),reo(h),dt,x0(k),f(h),it,rho,D,un);
%write_export(st(k),reo(h),pos(1:end-1),cyc(1:end-1));
write_pbs(x0(k),f(h),node,core);
ang(1)=0;
max(1)=0;

end
end

```

7.9 FUNCION WRITE_JOU_TEST3

```
function []=write_jou_test2(pos,cyc,ang,st,reo,dt,x,f,it,rho,D,un)

c1=2*pi*f*x;
c2=2*pi*f;
area=1.749812e-05; %pi*D^2/4; %%OJO, PONER VALOR REAL MUELLE

xm=x*1000;

if f<1
f=f*10;
cadena=sprintf('MH7x%dmm_f0%dHz.jou',round(xm),round(f));
profile=sprintf('MH7x%dmm_f0%dHz',round(xm),round(f));
else
cadena=sprintf('MH7x%dmm_f%dHz.jou',round(xm),round(f));
profile=sprintf('MH7x%dmm_f%dHz',round(xm),round(f));
end

fid = fopen(cadena,'w');
%fwrite(fid,'rcd mrfv0.cas.gz');
ff=sprintf('rcd %s.cas.gz',profile);
fwrite(fid,ff);
fprintf(fid,'\n');

%fwrite(fid,'/solve/initialize/initialize-flow');
%fprintf(fid,'\n');
fwrite(fid,'/solve/monitors/residual/convergence-criteria');
fprintf(fid,'\n');
fwrite(fid,'1e-9'); %continuity
fprintf(fid,'\n');
fwrite(fid,'5e-4'); %x-momentum
fprintf(fid,'\n');
fwrite(fid,'5e-4'); %y-momentum
fprintf(fid,'\n');
fwrite(fid,'5e-4'); %z-momentum
fprintf(fid,'\n');
fwrite(fid,'1e-3'); %z-momentum
fprintf(fid,'\n');
fwrite(fid,'/define/models/unsteady-2nd-order');
fprintf(fid,'\n');
fwrite(fid,'/solve/set/discretization-scheme/pressure 10');
fprintf(fid,'\n');
fwrite(fid,'/solve/set/discretization-scheme/mom 0');
fprintf(fid,'\n');
fwrite(fid,'/solve/set/reporting-interval 10');
fprintf(fid,'\n');
f3=sprintf('/solve/set/time-step %1.10f',dt);
fwrite(fid,f3);
fprintf(fid,'\n');
```

```
%condiciones de contorno entrada/salida
```

```
t=0;
```

```
for k=1:125 %250 ciclos
```

```
  for jk=1:2
```

```
    for i=10:105
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'file/read-profile');
```

```
      fprintf(fid,'\n');
```

```
      f1=sprintf('%s_C3A%d.dat',profile,round(1000*ang(i)));
```

```
      fwrite(fid,f1);
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'/define/boundary-conditions/velocity-inlet');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'inlet.1.1');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'n');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'y');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'y');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'y');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'y');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'n');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'"interior-12"');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'"x-velocity"');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'y');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'n');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'"interior-12"');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'"y-velocity"');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'y');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'n');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'"interior-12"');
```

```
      fprintf(fid,'\n');
```

```
      fwrite(fid,'"z-velocity"');
```

```
      fprintf(fid,'\n');
```

```

fwrite(fid,'n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fwrite(fid,'n');
fprintf(fid,'\n');
fwrite(fid,'n');
fprintf(fid,'\n');
fwrite(fid,'0');
fprintf(fid,'\n');
fwrite(fid,'quit');
fprintf(fid,'\n');
fwrite(fid,'quit');
fprintf(fid,'\n');
f4=sprintf('/solve/dual-time-iterate %d %d',1,it);
fwrite(fid,f4);
%%
if i==21 || i==33 || i==45 || i==57 || i==69 || i==81 || i==93 || i==105
fprintf(fid,'\n');
fwrite(fid,'file/export/tecplot');
fprintf(fid,'\n');
ciclo=(k-1)*2+jk; %escribe fichero de seguridad

tecplot=sprintf('TECx%dmm_f%dHz_C%d_A%d',round(xm),round(f),ciclo,round
(1000*ang(i)));
ff=sprintf('%s.plt',tecplot);
fwrite(fid,ff);
fprintf(fid,'\n');
fwrite(fid,'meridional');
fprintf(fid,'\n');
fprintf(fid,'\n');
fwrite(fid,'y-velocity');
fprintf(fid,'\n');
fwrite(fid,'z-velocity');
fprintf(fid,'\n');
fwrite(fid,'trazador'); %CREO QUE CON ESTA VARIABLE NOS
EXPORTARA LA FRACCIÓN MÁSCICA DE TRAZADOR - PERO PARA ESO
HAY QUE LLAMARLO "TRAZADOR" EN LA CONFIGURACIÓN (NO "AGUA",
U OTROS NOMBRES)
fprintf(fid,'\n');
fwrite(fid,'q');
end
%%

end
end %jk

ciclo=(k-1)*2+jk; %escribe fichero de seguridad
datos=sprintf('SECx%dmm_f%dHz_C%d',round(xm),round(f),ciclo);
fprintf(fid,'\n');
ff=sprintf('wd %s.dat.gz',datos);

```

```
fwrite(fid,ff);
%fprintf(fid,'\n');
```

```
end %k
fwrite(fid,'q');
fprintf(fid,'\n');
fwrite(fid,'exit');
fprintf(fid,'\n');
fwrite(fid,'yes');
fprintf(fid,'\n');
fclose(fid);
```

7.10 FUNCION WRITE_PBS

```
function []=write_pbs(x,f,node,core)
x=x*1000;
if f<1
f=f*10; %s=1.5 -> s=15;
cadena=sprintf('MH7x%dmm_f%dHz.pbs',round(x),round(f));
profile=sprintf('MH7x%dmm_f%dHz',round(x),round(f));
else
cadena=sprintf('MH7x%dmm_f%dHz.pbs',round(x),round(f));
profile=sprintf('MH7x%dmm_f%dHz',round(x),round(f));
end

define=sprintf('#PBS -lnodes=%d:ppn=%d',node,core);
fid = fopen(cadena,'w');
fwrite(fid,'#!/bin/bash');
fprintf(fid,'\n');
fwrite(fid,define);
fprintf(fid,'\n');
fwrite(fid,'#PBS -m e');
fprintf(fid,'\n');
fwrite(fid,'#PBS -M ALCARAZARCHENA@gmail.es');
fprintf(fid,'\n');
fwrite(fid,'cd $PBS_O_WORKDIR');
fprintf(fid,'\n');
fwrite(fid,'NB=`wc -l $PBS_NODEFILE | awk "{print $1}"`');
fprintf(fid,'\n');
fwrite(fid,'ulimit -l unlimited');
fprintf(fid,'\n');
```

```

fwrite(fid, '# echo "Numero de procesos: $NB"');
fprintf(fid, '\n');
fwrite(fid, '# echo "En los nodos:"');
fprintf(fid, '\n');
fwrite(fid, '# cat $PBS_NODEFILE');
fprintf(fid, '\n');
fwrite(fid, 'cd $PBS_O_WORKDIR');
fprintf(fid, '\n');
ff=sprintf('/share/apps/Fluent.Inc/bin/fluent 3ddp -t${NB} -ssh -pethernet -
cnf=${PBS_NODEFILE} -g -i %s.jou > %s-${NB}-procs.out', profile, profile);
fwrite(fid, ff);
fclose(fid);

```

7.11 VIDEO EN TECPLOT

```

clear all
close all

```

```

fid = fopen('video.mcr', 'w');

```

```

fwrite(fid, '#!MC 1200');
fprintf(fid, '\n');
fwrite(fid, '# Created by Tecplot 360 build 12.1.0.6712');
fprintf(fid, '\n');
fwrite(fid, '$!VarSet |MFBD| = "G:\MH1_Re100"');
fprintf(fid, '\n');
fwrite(fid, '$!EXTENDEDCOMMAND ');
fprintf(fid, '\n');
fwrite(fid, 'COMMANDPROCESSORID = "Link Solution Time"');
fprintf(fid, '\n');
fwrite(fid, ' COMMAND = "ENABLELINKING"');
fprintf(fid, '\n');
fwrite(fid, '$!EXPORTSETUP EXPORTFORMAT = AVI');
fprintf(fid, '\n');
fwrite(fid, '$!EXPORTSETUP EXPORTREGION = ALLFRAMES');
fprintf(fid, '\n');
fwrite(fid, '$!EXPORTSETUP IMAGEWIDTH = 955');
fprintf(fid, '\n');
fwrite(fid, '$!EXPORTSETUP USESUPERSAMPLEANTIALIASING = YES');
fprintf(fid, '\n');
fwrite(fid, '$!EXPORTSETUP SUPERSAMPLEFACTOR = 6');
fprintf(fid, '\n');

```

```

fwrite(fid,'$!EXPORTSETUP ANIMATIONSPEED = 6');
fprintf(fid,'\n');
fwrite(fid,'$!EXPORTSETUP EXPORTFNAME =
"G:\MH1_Re100\mh1_re100.avi");
fprintf(fid,'\n');
fwrite(fid,'$!EXPORTSTART ');
fprintf(fid,'\n');
fwrite(fid,' EXPORTREGION = ALLFRAMES');
fprintf(fid,'\n');
for i=1:64
fwrite(fid,'$!EXPORTNEXTFRAME ');
fprintf(fid,'\n');
f1=sprintf('$!GLOBALTIME SOLUTIONTIME = %d',i);
fwrite(fid,f1);
fprintf(fid,'\n');
end
fwrite(fid,'$!EXPORTFINISH ');
fprintf(fid,'\n');
fwrite(fid,'$!RemoveVar |MFBD|');
fprintf(fid,'\n');

fclose(fid);

```