



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior

Trabajo Fin de Grado

ESTUDIO PARAMÉTRICO NUMÉRICO DEL FLUJO CEREBROESPINAL SIMPLIFICADO

Alumno: Víctor Vidal Muñoz

Tutor: Prof. D. Cándido Gutiérrez Montes
Dpto: Ingeniería Mecánica y Minera

Junio, 2017



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Mecánica y Minera

Don Cándido Gutiérrez Montes , tutor del Proyecto Fin de Carrera titulado: Estudio paramétrico numérico del flujo cerebro espinal simplificado, que presenta Víctor Vidal Muñoz, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Febrero de 2017

El alumno:

Víctor Vidal Muñoz

Los tutores:

Cándido Gutiérrez Montes

José Ignacio Jiménez González

Índice

1. Introducción.....	6
1.1. Fundamento médico y fisiológico	6
1.2. Circulación del líquido cerebroespinal	8
2. Breve introducción a la simulación numérica.....	10
2.1. Mecánica de fluidos computacional.....	10
2.2. Simulaciones fluido-estructura	10
3. Estudio de CFD - FSI	11
3.1. Motivación	11
3.1.1. Aplicaciones clínicas	11
3.2. Estudios previos.....	12
4. Fundamento teórico y matemático	14
4.1. Ecuaciones que gobiernan la mecánica de fluidos [13]	14
4.1.1. Teorema de la divergencia.....	14
4.1.2. Teorema de transporte de Reynolds	15
4.1.3. Ecuación de continuidad	15
4.1.4. Ecuación de conservación de la cantidad de movimiento.....	16
4.1.5. Ecuaciones de Navier-Stokes en forma convectiva.....	17
4.1.6. Algoritmo implementado para el cálculo numérico	17
4.2. Ecuaciones que gobiernan la mecánica del sólido [14]	17
4.2.1. Relación entre la tensión y deformación. Módulo de Young	18
4.2.2. Deformaciones transversales. Coeficiente de Poisson	19
4.2.3. Leyes de Hooke generalizadas	21
4.2.4. Ecuaciones de Lamé.....	22
4.2.5. Planteamiento general del problema elástico	22
4.2.6. Tensión y deformación plana	24
4.2.7. Módulo de compresibilidad.....	24
4.2.8. Leyes constitutivas.....	25
4.2.9. Ecuaciones de Navier-Lamé	25
4.2.10. Fórmulas implementadas para el cálculo numérico.....	26
4.3. Acoplamiento	27
5. Especificaciones del flujo[15]	29
5.1. Leading-order.....	29
5.2. Steady streaming	29

5.3.	Parámetros adimensionales	30
5.3.1.	Número de Womersley	30
5.3.2.	Número adimensional de la onda elástica	30
5.3.3.	Parámetro ϵ	31
6.	Problemas FSI (Fluid-Structure interaction)	32
6.1.	Conceptos básicos	32
6.2.	Condiciones dinámicas y cinemáticas	33
7.	Mallado y configuraciones	35
7.1.	Generación de la malla: ANSYS ICEM CFD	35
7.1.1.	Introducción al programa	35
7.1.2.	Creación de la malla	35
7.2.	Conversión de la malla	39
8.	Parámetros de los materiales	41
8.1.	Líquido	41
8.2.	Sólido	41
9.	Resolución usando el software OpenFOAM®	42
9.1.	Introducción	42
9.2.	Instalación del programa	42
9.2.1.	Instalación del paquete FSI	43
9.2.2.	Instalación de ParaView®	43
9.3.	Sistema de archivos de OpenFoam®	44
10.	Solving	47
10.1.	Control de entrada y salida ('controlDict')	47
10.1.1.	Control del tiempo	47
10.1.2.	Control de la salida	47
10.1.3.	Librería y funciones	47
10.2.	Esquema numérico ('fvSchemes')	48
10.2.1.	Esquema de tiempo	48
10.2.2.	Esquema del gradiente	48
10.2.3.	Esquema de divergencia	48
10.2.4.	Esquema del laplaciano	48
10.2.5.	Esquema de interpolación	49
10.2.6.	Esquema de discretización de los gradientes normales a las caras	49
10.2.7.	Cálculo de flujo	49
10.3.	Control de la solución ('fvSolution')	49
10.3.1.	'Solvers'	49

10.3.2.	Factores de relajación	49
10.3.3.	PISO y SIMPLE.....	50
10.4.	Procesado en paralelo	50
10.5.	Aplicación: fsiFoam.....	50
11.	Configuraciones.....	51
12.	Análisis genérico.....	52
12.1.	Configuración A	52
12.1.1.	Solución promedio – ‘Steady streaming’	60
12.2.	Configuración B	66
12.2.1.	Solución promedio – ‘Steady streaming’	72
12.3.	Configuración C	74
12.3.1.	Solución promedio – ‘Steady streaming’	82
13.	Estudio paramétrico	87
14.	Conclusiones	92
14.1.	Estudios futuros	93
Referencias		95
15.	Anexo A – Archivos OpenFoam.....	97
15.1.	Condiciones de contorno – Líquido.....	97
15.1.1.	P	97
15.1.2.	pointMotionU	98
15.1.3.	U	99
15.2.	Condiciones de contorno – Sólido	100
15.2.1.	D	100
15.2.2.	DD.....	101
15.2.3.	pointD	102
15.2.4.	pointDD	103
15.3.	Archivos de la carpeta ‘constant’ - Líquido.....	103
15.3.1.	dynamicMeshDict	104
15.3.2.	flowProperties	104
15.3.3.	fsiProperties	105
15.3.4.	transportProperties.....	105
15.4.	Archivos de la carpeta ‘constant’ – Sólido.....	106
15.4.1.	rheologyProperties	106
15.4.2.	stressProperties	106
15.5.	Archivos de la carpeta ‘system’	107
15.5.1.	controlDict	107

15.5.2.	decomposeParDict.....	109
15.5.3.	fvSchemes	110
15.5.4.	fvSolution	111
16.	Anexo B – Archivos MatLab.....	112
16.1.	Lectura presión.....	112
16.2.	Lectura velocidad.....	113
16.3.	Perfiles de velocidad media.....	114

1. Introducción

El objeto de este trabajo de fin de grado es el estudio del comportamiento de las partes líquidas y sólidas que componen la fisiología del sistema nervioso en la columna vertebral. Dicho estudio se llevará a cabo mediante un software comercial de volúmenes finitos con el objetivo de definir los parámetros que intervienen en el problema.

Dado la compleja geometría del problema en cuestión, tomaremos una geometría simplificada del mismo que nos permitirá extrapolar resultados.

1.1. Fundamento médico y fisiológico

El líquido o fluido cerebroespinal, que denominaremos como 'CSF' del término en inglés Cerebrospinal Fluid, es una sustancia líquida incolora que está presente en el sistema nervioso tanto en el cerebro y la columna vertebral. También se denomina líquido cefalorraquídeo. Este fluido realiza varias funciones, de protección y del mantenimiento de la presión intracraneal y del estado de salud del cerebro.

El CSF circula entre las meninges que son membranas de tejido conectivo (o conjuntivo, un conjunto heterogéneo de tejidos orgánicos que comparten un origen común) que cubren todo el sistema nervioso central, añadiéndole una protección blanda frente a la dureza de la estructura ósea, por lo tanto, aportan una amortiguación mecánica. Además, las meninges actúan como una barrera química e impiden la entrada de sustancias perjudiciales para nuestro sistema nervioso. Esto nos protege de infecciones y de daño neurológico.

En la figura 1.1 podemos observar como se distribuyen las meninges en el cráneo.

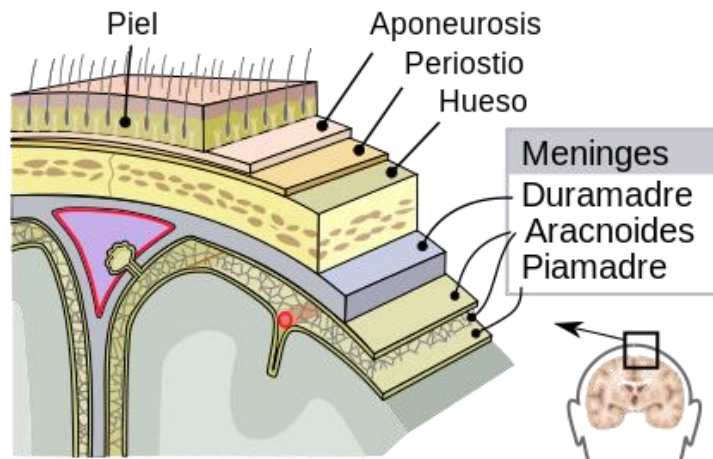


Figura 1.1. Estructura de las meninges [1]

Estas meninges son la duramadre, que es la capa más externa; los aracnoides y la pia madre, una capa delgada en contacto con el encéfalo. Existen, además, dos espacios intermedios, el espacio subaracnoideo, que contiene el líquido cerebroespinal y se encuentra entre la pia madre y el aracnoides, también lo nombraremos como SAS (abreviatura del inglés Subarachnoid Space); y el espacio subdural, es muy estrecho pero también contiene algo de CSF, se encuentra entre el aracnoides y la duramadre.

Más abajo, en la columna vertebral la configuración se mantiene, pero la pia madre está en contacto con la médula espinal en este caso.

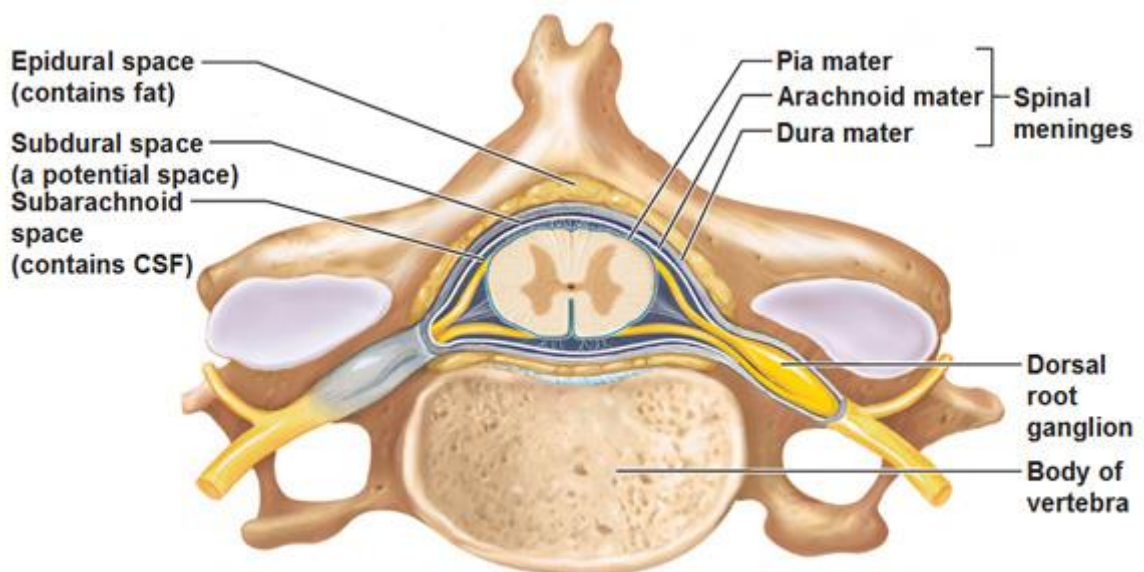


Figura 1.2. Corte transversal de la columna vertebral [2]

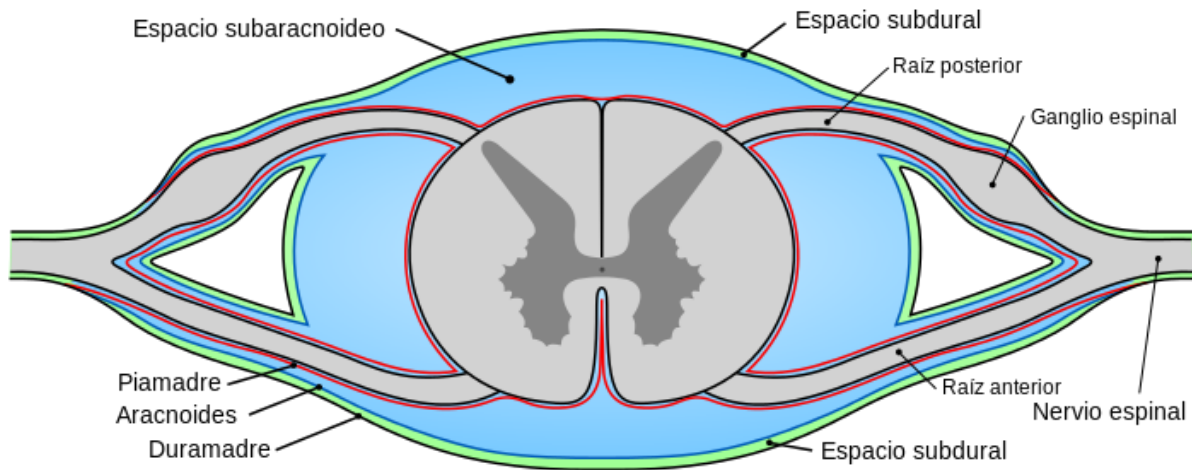


Figura 1.3. Detalle de la sección de la médula espinal y sus membranas [3]

La figura 1.2 y la 1.3 muestran un corte transversal de la distribución de la médula espinal y de las meninges.

1.2. Circulación del líquido cerebroespinal

La circulación del LCR (o CSF) permite el intercambio de nutrientes y productos de desecho entre la sangre y el tejido nervioso. La mecánica del movimiento de este fluido en nuestro organismo no se comprende totalmente, lo que ha impulsado estudios como este. No obstante, existen ciertos factores que pueden generar el movimiento:

- **Impulso:** se produce desde las áreas donde se produce el fluido hasta donde este se absorbe.
- **Oscilación:** el LCR oscila continuamente con una amplitud que aumenta conforme se aproxima al cuarto ventrículo del cerebro, que es una cavidad de forma triangular por encima de la médula espinal.
- **Movimiento pulsátil:** el flujo, por lo general, se sincroniza con el pulso arterial. Dichas oscilaciones se originan por la expansión del cerebro y sus arterias durante la sístole.

La figura 1.4 muestra el flujo general del líquido cerebroaraquídeo por el conjunto cerebro-columna.

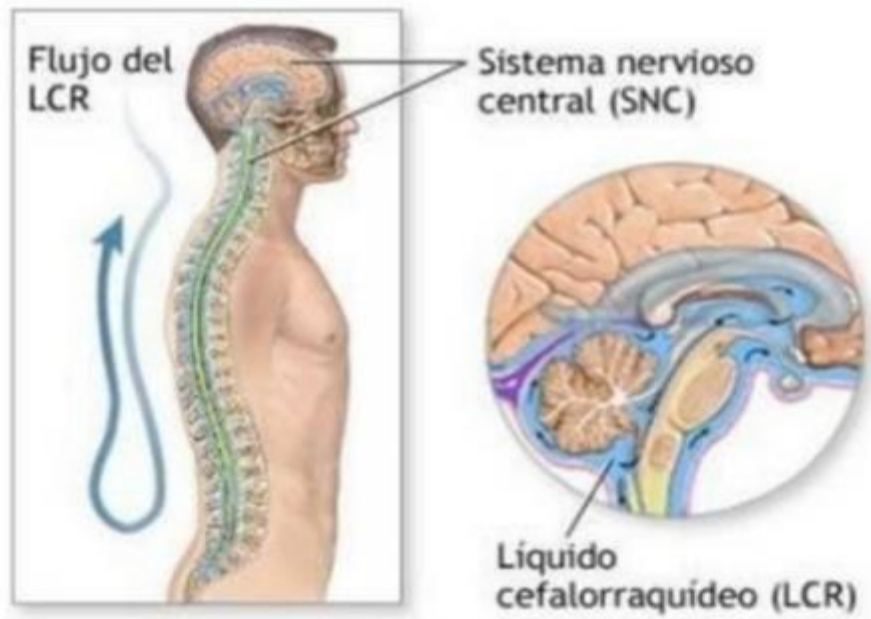


Figura 1.4. Esquema de la circulación del LCR [4]

2. Breve introducción a la simulación numérica

Una simulación por computadores es un programa informático que reproduce el comportamiento de un sistema usando un modelo matemático. Estas simulaciones se han convertido en una herramienta muy útil para el modelado matemático de sistemas naturales en física, química, elasticidad y resistencia de materiales, etc. Esto hace que estas simulaciones estén muy presentes en la ingeniería.

El modelo computacional es un conjunto de algoritmos y ecuaciones usados para mostrar el comportamiento del sistema modelado.

2.1. Mecánica de fluidos computacional

Un caso concreto de simulaciones por computación es la simulación de fluidos. La rama que se encarga de este estudio se conoce como la dinámica de fluidos computacional (CFD, del inglés 'Computational Fluid Dynamics').

Esta rama de la mecánica de fluidos usa algoritmos y métodos numéricos para resolver y analizar problemas que abarcan fluidos.

2.2. Simulaciones fluido-estructura

Este tipo específico de simulación consiste en acomodar problemas de dinámica de fluidos computacional con problemas de un sólido. Se usan ampliamente las siglas FSI (del inglés 'Fluid-structure interaction') para referirnos a estas simulaciones.

El proceso de funcionamiento de estas simulaciones será definido más detalladamente en apartados posteriores.

3. Estudio de CFD - FSI

Como hemos mencionado antes, no conocemos completamente la naturaleza de la mecánica de la circulación del LCR, por esto, el estudio numérico ha sido propuesto para estos estudios.

El problema que se plantea es un caso de interacción fluido-estructura donde el fluido entra con una presión oscilante.

3.1. Motivación

La principal motivación de estos estudios es conocer la mecánica del flujo cerebroespinal, esto podría aportarnos conocimientos útiles para la medicina u otras ciencias. Por ejemplo, se podría arrojar luz sobre ciertas enfermedades, como malformaciones en la columna vertebral, o bien, comprenderíamos cómo el cuerpo absorbe un determinado medicamento que se inyecta a través de la columna vertebral.

Concretamente, este trabajo estudiará los parámetros que intervienen en el problema para un posterior análisis.

Los modelos computacionales son una herramienta útil para resolver los campos de flujo y presión en espacio y en tiempo. Son capaces de indicar pequeños cambios en los patrones de velocidad del LCR. Los resultados que se obtienen pueden ser fácilmente comparados con los resultados 'in vivo'.

En muchos de estos estudios, que se nombran en apartados posteriores, se omite la iteración fluido-estructura, lo que genera una discrepancia entre los datos de un paciente y los obtenidos.

3.1.1. Aplicaciones clínicas

Aparte de por la simple ganancia de conocimiento, trabajos como este pueden ser potencialmente beneficiosos para el avance de la medicina en tres aspectos: el diseño de dispositivos biomédicos, con herramientas de CFD y la comprensión de la mecánica de fluidos en el cuerpo humano seremos capaces de optimizar los dispositivos para el correcto funcionamiento; la inyección intratecal (en el espacio lleno de LCR entre las capas delgadas de tejidos que cubren el cerebro y la médula espinal)

de medicamentos es ampliamente usada, estas herramientas nos guían a la aplicación efectiva de dicho medicamento (se ha demostrado que la amplitud y la frecuencia son factores clave en esta aplicación); y por último, ayuda a predecir ciertos fenómenos de transporte usando imágenes médicas.

3.2. Estudios previos

Se han realizado numerosas investigaciones sobre el movimiento del flujo cerebroespinal pulsante. Además, se han considerado enfermedades un gran número de estudios que afectan a la normal circulación de este líquido, como, por ejemplo, hidrocefálea,iringomielia o la malformación de Chiari tipo I.

Existen estudios que tratan el acoplamiento entre el sistema cardiovascular y el sistema nervioso que, como hemos mencionado antes, se cree que el movimiento pulsátil del líquido cerebroespinal viene dominado por la dilatación sistólica. Pero a la vez, se estudió la angioarquitectura del cerebro (la estructura de los vasos sanguíneos) y su distribución de presiones [5,6], figura 3.1.; y la interacción de las fuerzas, la presión sistólica infla los vasos sanguíneos que provoca el desplazamiento del LCR.

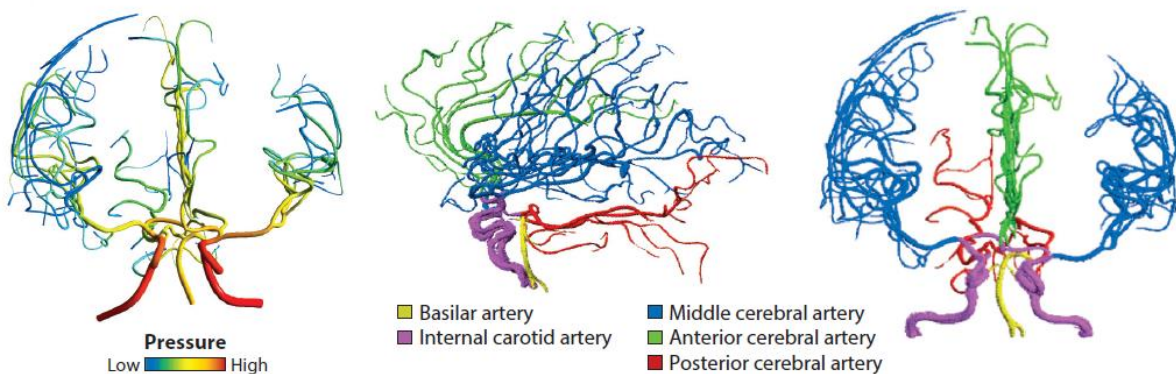


Figura 3.1. Angioarquitectura del cerebro humano en un estudio CFD [7]

Los mencionados estudios ingenieriles han sido acompañados de mediciones ‘in vivo’ de las características, geometría y mecánica de los fluidos del problema.

Cabe destacar el estudio de Loth et al. (2000) [8]. Loth usa un modelo de pequeñas secciones del sistema nervioso central con geometría idealizadas a lo largo

de la espina dorsal ajustando las condiciones de contorno para cada segmento. Así consigue el estudio de la espina dorsal en su totalidad, Figura 2.2.:

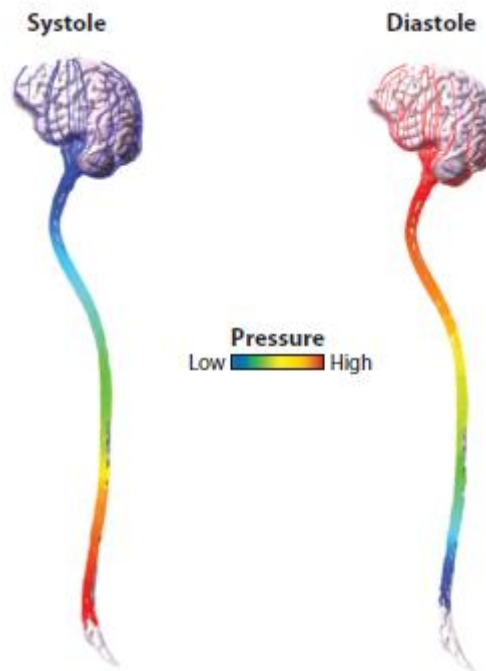


Figura 3.2. Modelo de la espina dorsal completo [5]

Por otro lado, se ha analizado la dispersión y distribución de una sustancia (como puede ser un medicamento o un contraste) en tiempo, distancia y concentración por el LCR que se inyecta a través de la columna. El estudio más reciente en este tema ha sido el de Tangen (2015)[9].

Simultáneamente, se han llevado a cabo mediciones sobre las ondas de velocidad que presentan las venas y arterias presentes en el cerebro que actúan directamente sobre la presión del mismo [10].

Al igual que en este trabajo han habido estudios de CFD como el de Kuttlet (2010)[11]. Pero la ausencia de algunas características anatómicas y de la omisión de los modelos FSI ha contribuido a discrepancias entre predicciones CFD y las medidas en pacientes.

4. Fundamento teórico y matemático

En los problemas de interacción fluido-estructura existe una dependencia mutua entre la parte del fluido y el sólido. El comportamiento del fluido depende de la forma de sólido y de su movimiento; y el movimiento y la deformación del sólido dependen de las fuerzas que ejerce el fluido.

Este apartado introducirá las ecuaciones parciales diferenciales que gobiernan el sólido y el fluido de nuestro problema.

4.1. Ecuaciones que gobiernan la mecánica de fluidos [13]

La parte de la mecánica de fluido está gobernada por las ecuaciones de Navier-Stokes para flujos incompresibles. Para trabajar con estas ecuaciones empleamos como fundamentos matemáticos el teorema de la divergencia y el teorema de transporte de Reynolds.

Dichas ecuaciones serán nombradas y estudiadas en términos generales y en tres dimensiones, aunque la disposición de nuestro problema sea en dos dimensiones.

Además de las que mostraremos, está la ecuación de conservación de la energía, aunque ésta no sea necesaria para el problema que nos ocupa.

4.1.1. Teorema de la divergencia

En cálculo vectorial, el teorema de la divergencia, también conocido como teorema de Gauss, relaciona el flujo de un campo vectorial a través de una superficie cerrada con la integral de su divergencia en un volumen delimitado por dicha superficie.

$$\int_{S_f} \Phi \vec{v} \cdot \vec{n} d\sigma = \int_{V_f} \nabla \cdot (\Phi \vec{v}) dV$$

(Fórmula 4.1)

Donde para un estudio general en tres dimensiones, el operador nabla es:

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}$$

(Fórmula 4.2)

En la fórmula 4.1 el vector $\vec{n}$ es normal a la superficie y va hacia fuera; y Φ es un campo vectorial de cualquier propiedad en derivadas parciales de primer orden continuas.

4.1.2. Teorema de transporte de Reynolds

El teorema de transporte de Reynolds estudia la variación respecto con el tiempo de la cantidad de una cierta magnitud fluida extensiva contenida en un volumen fluido $V_f(t)$ que está delimitado por la superficie $S_f(t)$.

$$\frac{d}{dt} \left[\int_{V_f(t)} \Phi \, dV \right]$$

(Fórmula 4.3)

Al evaluar la derivada en (4.3) aplicando la regla de Leibniz:

$$\frac{d}{dt} \left[\int_{V_f(t)} \Phi \, dV \right] = \int_{V_f(t)} \frac{\partial \Phi}{\partial t} \, dV + \int_{S_f(t)} \Phi \, \vec{v} \cdot \vec{n} \, d\sigma$$

(Fórmula 4.4)

A la derecha del igual, el primer término indica la variación temporal de la magnitud Φ en el volumen fluido; y el segundo, el flujo convectivo de Φ en las paredes del volumen fluido.

4.1.3. Ecuación de continuidad

Por definición, la masa en un sistema cerrado debe conservarse, es decir, no se crea ni se destruye masa. Por ello, la ecuación de conservación de la masa quedaría:

$$\frac{d}{dt} \left[\int_{V_f(t)} \rho dV \right] = 0$$

(Fórmula 4.5)

Con el teorema de Reynolds, previamente expuesto, podemos reescribir la ecuación 4.5 para un volumen de control,

$$\frac{d}{dt} \left[\int_{V_c(t)} \rho dV \right] + \int_{S_c(t)} \rho [(\vec{v} - \vec{v}_c) \cdot \vec{n}] d\sigma = 0$$

(Fórmula 4.6)

Esto muestra que el incremento de masa por unidad de tiempo en un volumen de control es igual a la masa que entra por unidad de tiempo a través de la superficie que delimita el volumen de control.

4.1.4. Ecuación de conservación de la cantidad de movimiento

Escribiendo la segunda ecuación de Newton en forma integral obtenemos:

$$\frac{d}{dt} \left[\int_{V_f(t)} \rho \vec{v} dV \right] + \int_{S_c(t)} \rho [(\vec{v} - \vec{v}_c) \cdot \vec{n}] d\sigma = - \int_{S_f(t)} p \vec{n} d\sigma + \int_{S_f(t)} \vec{\tau} \cdot \vec{n} d\sigma + \int_{V_f(t)} \rho \vec{f}_m dV$$

(Fórmula 4.7)

Si aplicamos el teorema de Reynolds, la ecuación 3.6 queda:

$$\frac{d}{dt} \left[\int_{V_c(t)} \rho \vec{v} dV \right] + \int_{S_c(t)} \rho \vec{v} [(\vec{v} - \vec{v}_c) \cdot \vec{n}] d\sigma = - \int_{S_c(t)} p \vec{n} d\sigma + \int_{S_c(t)} \vec{\tau} \cdot \vec{n} d\sigma + \int_{V_c(t)} \rho \vec{f}_m dV$$

(Fórmula 4.8)

Esta ecuación expresa cómo el incremento por unidad de tiempo de la cantidad de movimiento que hay en un volumen de control es igual a la suma del flujo convectivo de cantidad de movimiento que entra en el volumen de control a través de las paredes $-\int_{S_c(t)} \rho \vec{v} [(\vec{v} - \vec{v}_c) \cdot \vec{n}] d\sigma$ y los tres términos de la derecha. El primero es la resultante de las fuerzas de presión, el segundo la resultante de las fuerzas de viscosidad y por último, la resultante de las fuerzas másicas.

4.1.5. Ecuaciones de Navier-Stokes en forma convectiva

Hasta ahora hemos visto las ecuaciones en forma integral. Veamos ahora la ecuaciones en forma convectiva. Las ecuaciones para de Navier-Stokes para flujos incompresibles se pueden escribir como en la fórmula 4.9 y 4.10 en un dominio Ω_t con $t \in (0, T)$.

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u - \nu \nabla^2 u = -\nabla w + g$$

(Fórmula 4.9)

$$\text{Con } \nabla w = \frac{1}{\rho} \nabla p$$

$$\nabla \cdot u = 0$$

(Fórmula 4.10)

La primera ecuación corresponde a la conservación de momento y la segunda a la conservación de la masa. Estas son las fórmulas que se implementan para el cálculo numérico.

4.1.6. Algoritmo implementado para el cálculo numérico

El algoritmo que se emplearemos para la resolución de flujo incompresible es PISO que viene de las siglas del inglés ‘Pressure Implicit with Splitting of Operator’. Esto es un procedimiento para resolver las ecuaciones de Navier-Stokes definidas anteriormente.

El algoritmo PISO fue desarrollado inicialmente para problemas transitorios pero se ha extendido también a problemas estacionarios.

4.2. Ecuaciones que gobiernan la mecánica del sólido [14]

La mecánica del sólido incluye varias ramas de estudio: teoría de elasticidad (comportamiento elástico); teoría de la plasticidad, viscoelasticidad y viscoplasticidad (comportamiento plástico); y resistencia de materiales (cualquier tipo de comportamiento bajo hipótesis simplificativas). Por la naturaleza de nuestro estudio,

solo me centraré en la teoría de la elasticidad. En un caso bidimensional, concretamente.

4.2.1. Relación entre la tensión y deformación. Módulo de Young

La tensión y la deformación de un sólido son causa y efecto, por lo que necesitamos algo que las relacione. Para esto se recurrió al laboratorio, al conocido ensayo de tracción.

El ensayo de tracción consiste en someter a una fuerza de tracción en dirección axial, como su propio nombre indica, a una probeta de sección Ω . Esta fuerza genera una tensión en el interior del material que suponemos uniforme para cualquier sección recta. La tensión normal σ vendrá dada por:

$$\sigma = \frac{F}{\Omega}$$

(Fórmula 4.11)

La probeta se alarga debido al esfuerzo. Definimos un término adimensional de deformación unitaria ε :

$$\varepsilon = \frac{\Delta L}{L_0}$$

(Fórmula 4.12)

En la ecuación 4.12, L_0 es la longitud inicial de la probeta.

Para la realización del ensayo, vamos aumentando la fuerza aplicada y representamos la tensión frente a la deformación unitaria.

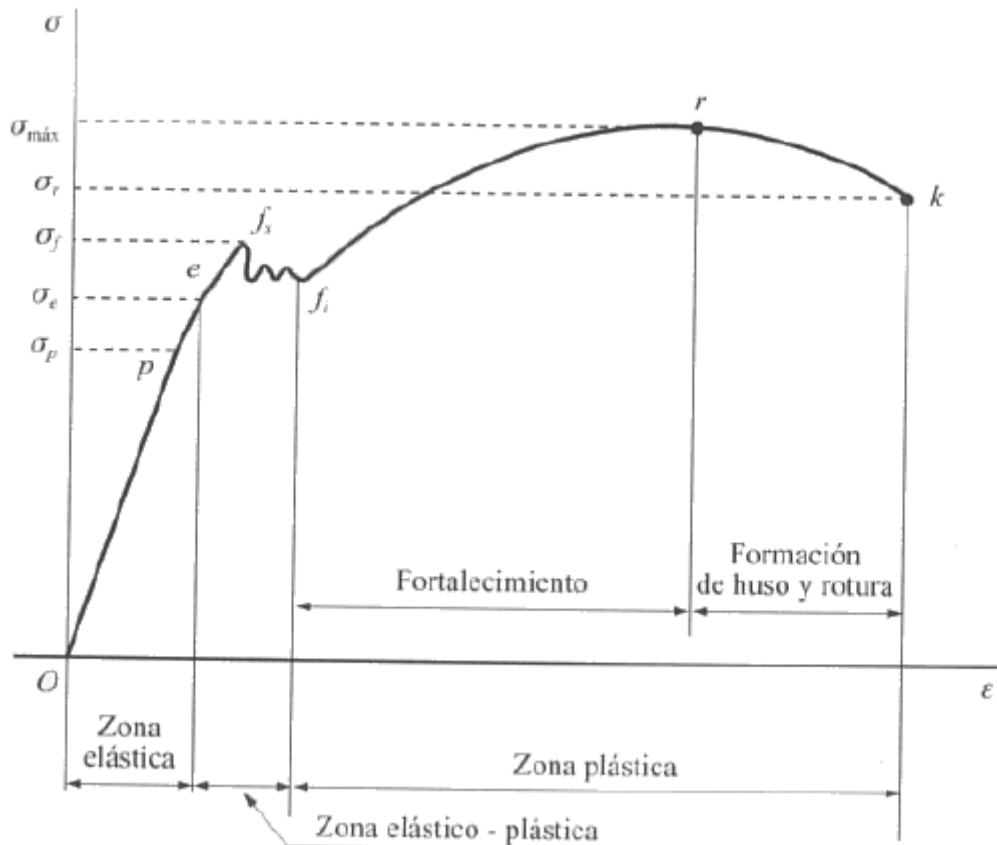


Figura 4.1. Diagrama tensión de formación [14]

La figura 4.1 representa un ejemplo de diagrama tensión-deformación para acero dulce, donde se distinguen las distintas zonas y puntos importantes.

En la zona elástica la gráfica es proporcional, siguiendo la relación de la ecuación 4.13.

$$\sigma = E \varepsilon$$

(Fórmula 4.13)

Siendo E una constante conocida como módulo de elasticidad longitudinal o módulo de Young. Esta expresión constituye la ley de Hooke en zona elástica. Las unidades del módulo de Young que se deducen de la expresión 3.10 son $[F][L]^{-2}$, que son unidades de tensión.

4.2.2. Deformaciones transversales. Coeficiente de Poisson

El ensayo de tracción descrito previamente no contempla la deformación transversal que existe, se observa una disminución de las dimensiones transversales.

Si consideramos un prisma rectangular como el de la figura 4.2, apreciamos este fenómeno.

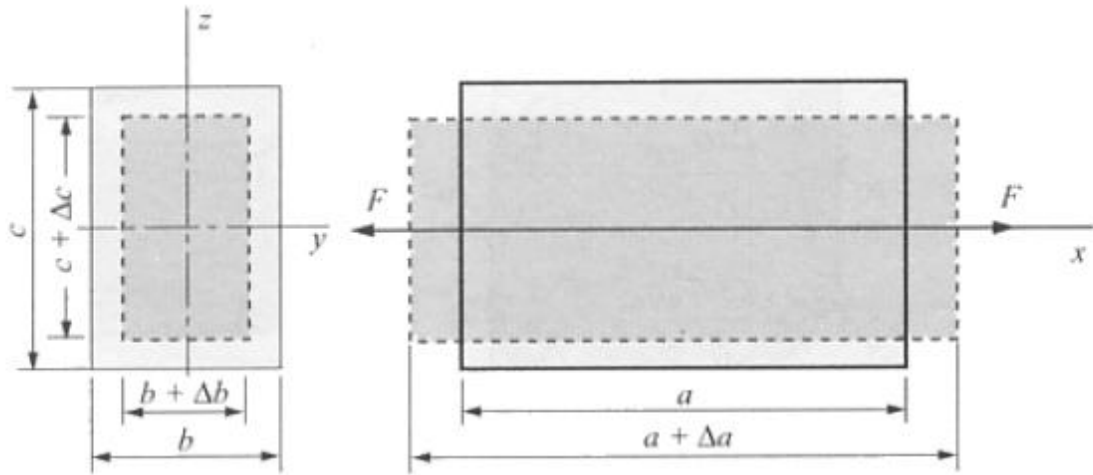


Figura 4.2. Esquema para unas deformaciones transversales [14]

Las deformaciones unitarias en los ejes y, z son:

$$\varepsilon_y = \frac{\Delta b}{b} \quad \varepsilon_z = \frac{\Delta c}{c}$$

(Fórmula 4.14)

Estas dos deformaciones unitarias son iguales cuando se trata de un material isotrópico, es decir, todas sus propiedades mecánicas y térmicas son las mismas en todas las direcciones.

Entonces, Poisson demostró que la relación entre la deformación lateral y longitudinal es constante. Esta relación constante es denominada como coeficiente de Poisson ν .

$$\nu = \frac{|\varepsilon_y|}{\varepsilon_x} = \frac{|\varepsilon_z|}{\varepsilon_x}$$

(Fórmula 4.15)

Otra propiedad destacable es la dilatación cúbica, esto se trata de la suma de las deformaciones en todos los ejes.

$$e = \varepsilon_x + \varepsilon_y + \varepsilon_z$$

(Fórmula 4.16)

Todo esto también sería aplicable para el caso de compresión solamente cambiando los signos.

Por último, cabe mencionar que el hecho de que haya deformación en una dirección no quiere decir que haya también un esfuerzo en la misma dirección.

4.2.3. Leyes de Hooke generalizadas

Consideramos un cubo con tamaño de arista igual a la unidad que se deforma como indica la figura 4.3. Como se conservan las caras paralelas, se deduce que las direcciones principales de las matrices de tensión y deformación son coincidentes.

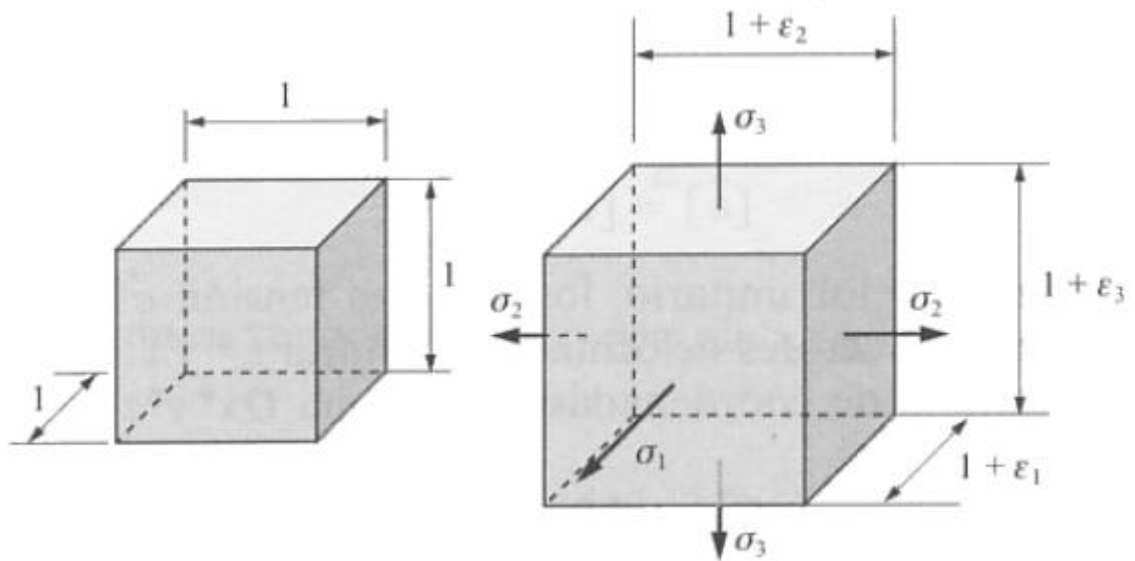


Figura 4.3. a) Cubo en reposo b) Cubo sometido a tensiones que experimenta una deformación [14]

Conociendo esto y aplicando el principio de superposición, concluimos que las deformaciones principales serán:

$$\varepsilon_1 = \frac{1}{E} [\sigma_1 - \nu(\sigma_2 + \sigma_3)]$$

$$\varepsilon_2 = \frac{1}{E} [\sigma_2 - \nu(\sigma_1 + \sigma_3)]$$

$$\varepsilon_3 = \frac{1}{E} [\sigma_3 - \nu(\sigma_1 + \sigma_2)]$$

(Fórmula 4.17)

4.2.4. Ecuaciones de Lamé

Las ecuaciones de Lamé, al contrario que la ley de Hooke, expresan las tensiones en función de las deformaciones considerando λ , el primer parámetro de Lamé, y G , el módulo de elasticidad transversal o módulo de cizalladura:

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad G = \frac{E}{2(1 + \nu)}$$

(Fórmula 4.18)

Obtemos de esta manera las ecuaciones de Lamé:

$$\sigma_x = \lambda e + 2G\varepsilon_x$$

$$\sigma_y = \lambda e + 2G\varepsilon_y$$

$$\sigma_z = \lambda e + 2G\varepsilon_z$$

$$\tau_{xy} = 2G\varepsilon_{xy} = G\gamma_{xy}$$

$$\tau_{yz} = 2G\varepsilon_{yz} = G\gamma_{yz}$$

$$\tau_{xz} = 2G\varepsilon_{xz} = G\gamma_{xz}$$

(Fórmula 4.19)

4.2.5. Planteamiento general del problema elástico

Si tenemos la matriz de deformaciones en los puntos del sólido elástico podemos obtener la matriz de tensiones con las ecuaciones de Lamé, o por el contrario, si tenemos la matriz de tensiones podemos obtener la matriz de deformaciones aplicado las leyes de Hooke.

Por otro lado tenemos las ecuaciones de equilibrio interno (4.20), las componentes de la matriz de deformación en función de las componentes del vector desplazamiento (u , v y w) (4.21) y las de compatibilidad (4.22).

$$X + \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} = 0$$

$$Y + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} = 0$$

$$Z + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} = 0$$

(Fórmula 4.20)

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad \varepsilon_z = \frac{\partial w}{\partial z}$$

$$\varepsilon_{xy} = \frac{1}{2} \gamma_{xy} = \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\varepsilon_{xz} = \frac{1}{2} \gamma_{xz} = \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)$$

$$\varepsilon_{yz} = \frac{1}{2} \gamma_{yz} = \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)$$

(Fórmula 4.21)

$$2 \frac{\partial^2 \varepsilon_x}{\partial y \partial z} = \frac{\partial}{\partial x} \left(-\frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{xz}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} \right)$$

$$2 \frac{\partial^2 \varepsilon_y}{\partial z \partial x} = \frac{\partial}{\partial y} \left(\frac{\partial \gamma_{yz}}{\partial x} - \frac{\partial \gamma_{xz}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} \right)$$

$$2 \frac{\partial^2 \varepsilon_z}{\partial x \partial y} = \frac{\partial}{\partial z} \left(\frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{xz}}{\partial y} - \frac{\partial \gamma_{xy}}{\partial z} \right)$$

$$\frac{\partial^2 \gamma_{yz}}{\partial y \partial z} = \frac{\partial^2 \varepsilon_z}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial z^2}$$

$$\frac{\partial^2 \gamma_{xz}}{\partial z \partial x} = \frac{\partial^2 \varepsilon_x}{\partial z^2} + \frac{\partial^2 \varepsilon_z}{\partial x^2}$$

$$\frac{\partial^2 \gamma_{yz}}{\partial x \partial y} = \frac{\partial^2 \varepsilon_y}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial y^2}$$

(Fórmula 4.22)

Estas ecuaciones constituyen un total de quince y, a su vez, tenemos quince incógnitas. El problema elástico queda totalmente definido.

4.2.6. Tensión y deformación plana

Con el ensayo de tracción hemos estudiado la tensión y deformación para una dimensión, veamos ahora en dos dimensiones. Hay numerosos casos, como es el que nos ocupa, donde el estado de tensión y deformación ocurren en un mismo plano. No obstante, hay que añadir que un caso de tensión plana no quiere decir que la deformación sea plana.

Para nuestro problema, tenemos tensión y deformación plana. Las matrices de deformación (4.23) y de tensión (4.24) se muestran a continuación.

$$\varepsilon = \begin{pmatrix} \varepsilon_x & \varepsilon_{xy} \\ \varepsilon_{xy} & \varepsilon_y \end{pmatrix}$$

(Fórmula 4.23)

$$\sigma = \begin{pmatrix} \tau_x & \varepsilon_{xy} \\ \varepsilon_{xy} & \tau_y \end{pmatrix}$$

(Fórmula 4.24)

4.2.7. Módulo de compresibilidad

El módulo de compresibilidad (K) de un material determinará cuanto se comprimirá un sólido sometido a una determinada presión externa. Por lo tanto, el módulo de compresibilidad relaciona la variación de presión a la que el cuerpo es sometido con la variación relativa de volumen:

$$K = -\frac{\Delta p}{\Delta V/V} = -V \frac{\Delta p}{\Delta V}$$

(Fórmula 4.25)

Las unidades son, por tanto, de presión.

4.2.8. Leyes constitutivas

Para nuestro estudio tomaremos a modo de simplificación la hipótesis de que el sólido es lineal, isótropo y homogéneo. Por esto, las propiedades elásticas son determinadas únicamente por dos módulos. Usaremos el módulo de Young y el de Poisson.

Las fórmulas que relacionan los parámetros fundamentales son las llamadas leyes constitutivas.

4.2.9. Ecuaciones de Navier-Lamé

Las ecuaciones vistas hasta ahora no son suficientes para llevar a cabo la simulación, ya que necesitamos unas fórmulas que podamos discretizar.

La formulación del problema se realizara con el principio de conservación del movimiento que viene de la segunda ley de Newton. De esta ecuación, que definiremos ahora, con unas condiciones de contorno previamente establecidas se obtiene el campo de desplazamientos que se origina por la acción de fuerzas externas. En el caso transitorio, como es nuestro caso, tendremos además que especificar el campo de desplazamientos y velocidades inicial.

$$\frac{d}{dt} \int_{\Omega} \rho \vec{v} d\Omega = \int_{\Omega} \vec{f} d\Omega + \int_{\partial\Omega} \vec{t} dS$$

(Fórmula 4.26)

Con $\vec{f}$ la densidad volumétrica de las fuerzas externas y la densidad superficial de las fuerzas externas ($\vec{t} = \sigma \vec{n}$) que actúan sobre la frontera del dominio $\partial\Omega$.

Ahora si aplicamos el teorema de Gauss y proyectamos la ecuación en los ejes de coordenadas obtenemos la ecuación de conservación de movimiento en forma diferencial:

$$\rho \frac{\partial^2 u_i}{\partial t^2} - \sum_{j=2}^3 \frac{\partial \sigma_{ij}}{\partial x_j} = f_i \quad i = 1, 2, 3$$

(Fórmula 4.27)

Siendo $u_i(\bar{x}, t)$ el campo de desplazamientos y $\sigma_{ij}(\bar{x}, t)$ el tensor de tensiones. Estos campos no son independientes. Se pueden relacionar con las ecuaciones o leyes constitutivas.

Finalmente, las ecuaciones de Navier-Lamé en coordenadas cartesianas en tres dimensiones quedan:

$$G \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + (\lambda + G) \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + F_x = 0$$

$$G \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + (\lambda + G) \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + F_y = 0$$

$$G \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + (\lambda + G) \frac{\partial}{\partial z} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + F_z = 0$$

(Fórmula 4.28)

4.2.10. Fórmulas implementadas para el cálculo numérico

El método de resolución para el sólido lineal elástico que implementa el programa de volúmenes finitos que vamos a usar utiliza la fórmula 4.29.

$$\int_{V_s} \rho_s \frac{\partial \delta v}{\partial t} dV_s - \oint_{S_s} n_u \cdot (2G + \lambda) \nabla \delta \vec{u} dS_s = \oint_{S_s} n_u \cdot \vec{q} dS_s + \int_{V_s} \rho_s \frac{\partial \delta f_b}{\partial t} dV_s$$

$$q = G(\nabla \delta \vec{u})^T + \lambda \text{tr}(\delta \vec{u})I - (G + \lambda) \nabla \delta \vec{u} + G \nabla \delta \vec{u} \cdot (\nabla \delta \vec{u})^T + \frac{1}{2} G(\nabla \delta \vec{u} : \nabla \delta \vec{u})I + \sum_s \cdot \delta F_s^T$$

$$+ \delta \sum_s \cdot \delta F_s^T$$

(Fórmula 4.29)

La fórmula 4.30 muestra como se lleva a cabo la discretización espacial:

$$\rho_{PS} V_{PS} \frac{\partial \delta v_P}{\partial t} - \sum_f (2G_f + \lambda) n_{fS} \cdot (\nabla \delta \vec{u})_f S_{fS} = \sum_f n_{fS} \cdot q_f S_{fS} + \sum_f \rho_{PS} (\delta f_b)_P V_{PS}$$

$$n_{fS} \cdot (\nabla \delta \vec{u})_f = |\Delta_{fS}| \frac{\delta \vec{u}_N - \delta \vec{u}_P}{|d_{fS}|} + (n_{fS} - \Delta_{fS}) \cdot (\nabla \delta \vec{u})_f$$

(Fórmula 4.30)

Por último, la discretización en tiempo se hace hacia atrás, se observa en la fórmula 4.31:

$$\delta v_P^n = \frac{3\delta \vec{u}_P^n - 4\delta \vec{u}_P^0 + \delta \vec{u}_P^{00}}{2\delta t}$$

$$\left(\frac{\partial \delta v_P}{\partial t} \right)_n = \frac{3\delta v_P^n - 4\delta v_P^0 + \delta v_P^{00}}{2\delta t}$$

(Fórmula 4.31)

4.3. Acoplamiento

El acoplamiento que se va a emplear es el implícito, ya que es el más adecuado para una interacción entre el fluido y el líquido fuerte, en la que existen grandes deformaciones. Este proceso de acoplamiento se esquematiza en la figura 4.4.

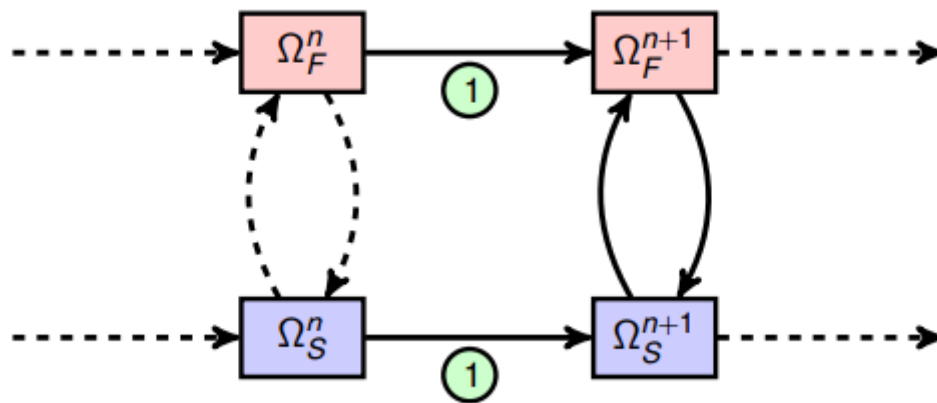


Figura 4.4. Acoplamiento implícito [13]

Este intercambio de información que se observa consiste en las variables de presión (p_r) y fuerzas viscosas ($\vec{t}_r$) del campo fluido a la estructura sólida; y, de campo de desplazamientos ($\vec{u}_r$) y velocidad ($\vec{v}_r$) de la estructura al campo fluido.

El proceso se llevará a cabo mediante iteración tal y como muestra la figura 4.5.

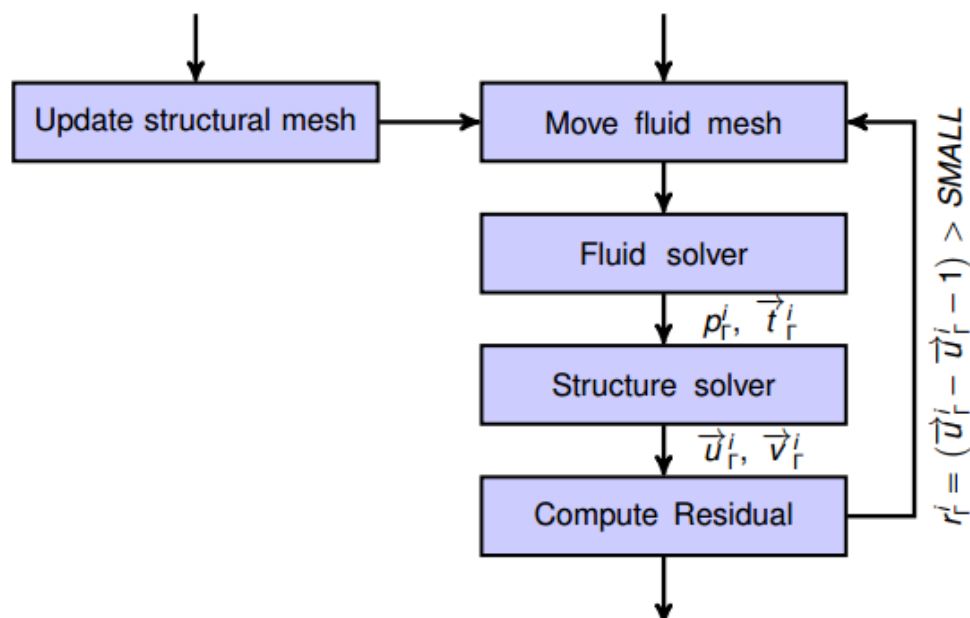


Figura 4.5. Precedimiento de iteración [16]

El acoplamiento de los 'solvers' se define dentro de la 'constant' en el archivo 'fsiProperties' como se observa en el anexo. Y el acoplamiento de las zonas se explica en el apartado de mallado.

5. Especificaciones del flujo[18]

5.1. Leading-order

Los términos ‘leading-order’ dentro de una ecuación, modelo o expresión matemática son los términos de mayor orden de magnitud. Los tamaños de los diferentes términos en una ecuación varían conforme lo hacen las variables, así que, también cambian los términos mayores.

Investigando qué términos de la ecuación son mayores es una forma de simplificar y entender los modelos matemáticos de un problema que nos ocupa.

Este proceso se lleva a cabo comúnmente en mecánica de fluidos para simplificar las ecuaciones de Navier-Stokes.

Para nuestro problema, un flujo oscilatorio, obtenemos que nuestra solución ‘leading-order’, es decir, la solución solo con los términos de mayor orden es puramente oscilatoria, mientras que las correcciones incluyen una componente estacionaria como resultado del movimiento del sólido.

5.2. Steady streaming

‘Steady streaming’ es un término que se usa para indicar que existe una corriente fija hacia la salida del conducto. A causa de interacciones no lineales entre el fluido y el sólido, asociadas a los términos convectivos y a las variaciones temporales y espaciales del espesor del canal, las correcciones de primer orden del flujo contienen una componente fija añadida a la componente oscilatoria.

Esta componente fija es la que estudiaremos calculando la media de las ondas de velocidad respecto al tiempo.

5.3. Parámetros adimensionales

5.3.1. Número de Womersley

El número de Womersley (α) es un número adimensional usado en biomecánica de fluidos. Esta expresión relaciona la frecuencia de un flujo pulsante con los efectos viscosos.

Definimos este número adimensional con la siguiente relación:

$$\alpha^2 = \frac{\text{fuerzas de inercias transitorias}}{\text{esfuerzos viscosos}} = \frac{\rho \omega U}{\mu U L^{-2}} = \frac{\omega L^2}{\mu \rho^{-1}} = \frac{\omega L^2}{\nu}$$

Donde L es la longitud característica del problema, en nuestro caso el espesor e ; ω la frecuencia de las oscilaciones; y ν la viscosidad cinemática del fluido.

$$\alpha^2 = \frac{\omega e^2}{\nu}$$

(Fórmula 5.1)

El número de Womersley es de orden la unidad dado que las fuerzas viscosas son comparables a la aceleración local.

$$\alpha \sim 1$$

5.3.2. Número adimensional de la onda elástica

Este número adimensional de la onda elástica ' K ' se define así:

$$K = \frac{L}{(E/\rho)^{1/2}/\omega} = \frac{e}{(E/\rho)^{1/2}/\omega}$$

(Fórmula 5.2)

Donde L es la longitud característica del problema, en nuestro caso el espesor e ; ω la frecuencia de las oscilaciones, E es el módulo de Young del material, y ρ la densidad del material.

Este número adimensional también será de orden la unidad.

$$K \sim 1$$

5.3.3. Parámetro ε

Además de los números adimensional expuestos, existe un parámetro $\varepsilon \ll 1$. Como hemos mencionado anteriormente en el apartado de 'leading order', existe una componente fija añadida a la componente oscilatoria. Las velocidades características de dicha velocidad fija corresponden a:

$$\varepsilon U = \varepsilon^2 \omega L$$

(Fórmula 5.3)

6. Problemas FSI (Fluid-Structure interaction)

6.1. Conceptos básicos

Este tipo de problemas es muy común en ingeniería, desde problemas aeroespaciales hasta biomecánica.

En los análisis de problemas fluido-estructura, las fuerzas del fluido se aplican al sólido y las deformaciones del fluido cambian el dominio del fluido. Para la mayoría de estos problemas, el dominio computacional se divide en dominio fluido y dominio sólido, cada dominio tiene sus propiedades definidas por separado. La interacción existente se produce en la interfase de los dos dominios. Para la simulación de estos problemas es necesario que exista un acoplamiento entre ambos dominios.

Los métodos que se usan para resolver estos problemas se pueden dividir en dos:

1. Método simultáneo o monolítico: la interacción entre el sólido y el fluido a través de la interfase mutua es tratada a la vez mediante un modelo matemático y un esquema de discretización unificado. Este método es preciso y estable pero costoso computacionalmente. Se recomienda cuando hay una fuerte dependencia fluido-sólido.
2. Método separado (partitioned): los modelos matemáticos se estudian por separado pero se comunican a través de la interfase. Suele ser menos estable y preciso que el simultáneo. Al contrario que este, se usa cuando la dependencia es menor entre el fluido y el sólido.

El paquete FSI de OpenFOAM usa el segundo método, pero dispone de un método de acoplamiento implícito que permite que se use en problemas donde la interacción es fuerte. La figura 6.1 muestra la diferencia entre el acoplamiento fuerte y el débil. Observamos que en el acoplamiento fuerte existe una retroalimentación más.

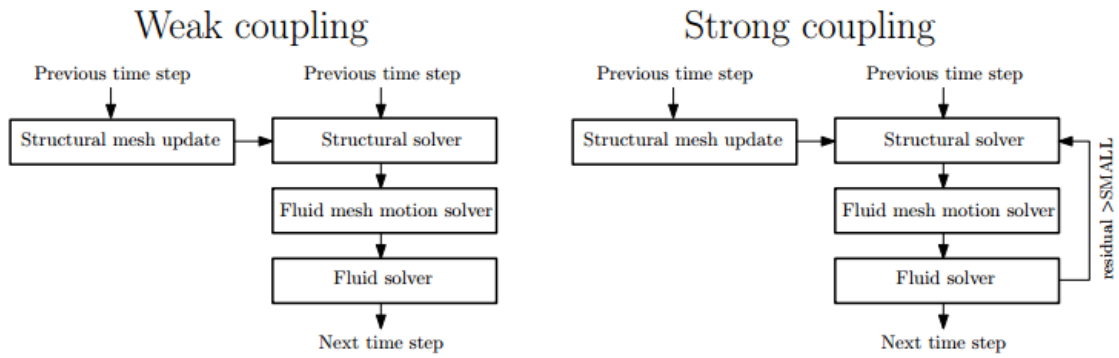


Figura 6.1. Proceso de resolución de estos problemas [17]

Este paquete FSI está aún en desarrollo y distinto a los demás 'solvers', ya que emplea dos 'solvers' (uno para el líquido y otro para el sólido) a su vez que intercambian continuamente información entre sí.

El paquete que nosotros hemos instalado cuenta con seis 'solvers': crackStressFoam, flowFoam, stressFoam, thermalStressFoam, weakFsiFoam y fsiFoam. Nosotros trabajaremos con este último.

6.2. Condiciones dinámicas y cinemáticas

Denotaremos como Ω_0 al dominio de nuestro problema, que incluye la parte sólida y la líquida. Este dominio admite que se descomponga de la siguiente manera:

$$\Omega_0 = \overline{(\Omega_1)_0 \cup (\Omega_2)_0}$$

(Fórmula 6.1)

Siendo $(\Omega_f)_0$ y $(\Omega_s)_0$ los dominios del sólido y del fluido. Los subíndices indican el tiempo de la simulación siendo el dominio, en general, Ω_t .

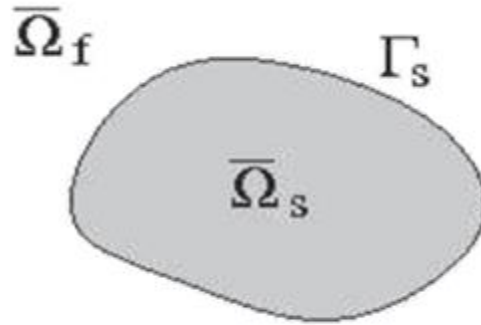


Figura 6.2 Dominio del problema [12]

La figura 6.2 muestra el dominio del problema. Se observa la interfase entre los subdominios denotada como Γ_s . En la interfase tendremos que el desplazamiento del fluido y del sólido son iguales y las tensiones las mismas. Además, la velocidad del fluido y del sólido serán la misma en la interfase si tenemos una condición de 'no-slip'. Las celdillas internas de la malla del fluido ajustarán sus posiciones conforme la cara entre el sólido y el líquido de mueva.

Para que esto sea así, tenemos que implementar en la interfase una condición de movimiento.

7. Mallado y configuraciones

La geometría real que comprende este estudio es muy compleja. Aunque se podría conseguir fácilmente mediante técnicas CAD ('Computer-Aided Design') con la disposición de imágenes transversales de la columna que se obtienen en un TAC (Tomografía Axial Computarizada). No obstante, este no es el objetivo de este trabajo y, además, esto generaría unas simulaciones computacionalmente muy costosas.

En lugar de esto, trabajaremos con geometrías simplificadas y con varias configuraciones que nos permitan arrojar luz sobre algunas dudas que plantea este trabajo.

7.1. Generación de la malla: ANSYS ICEM CFD

7.1.1. Introducción al programa

ANSYS ICEM CFD es un programa que permite la creación de geometrías complejas y generación de mallas, además, dispone de herramientas y diagnosis de estas.

Está diseñado para aplicaciones ingenieriles tanto para mecánicas de fluidos computacional como para análisis estructural. Es un programa muy potente que permite generar cualquier tipo de malla, este estudio abarca principalmente mallas planas estructuradas de cuadrados.

7.1.2. Creación de la malla

La creación de la malla se realiza al igual que en otros programas de mallado. Como se muestra en la figura 7.1, se sigue un determinado procedimiento:

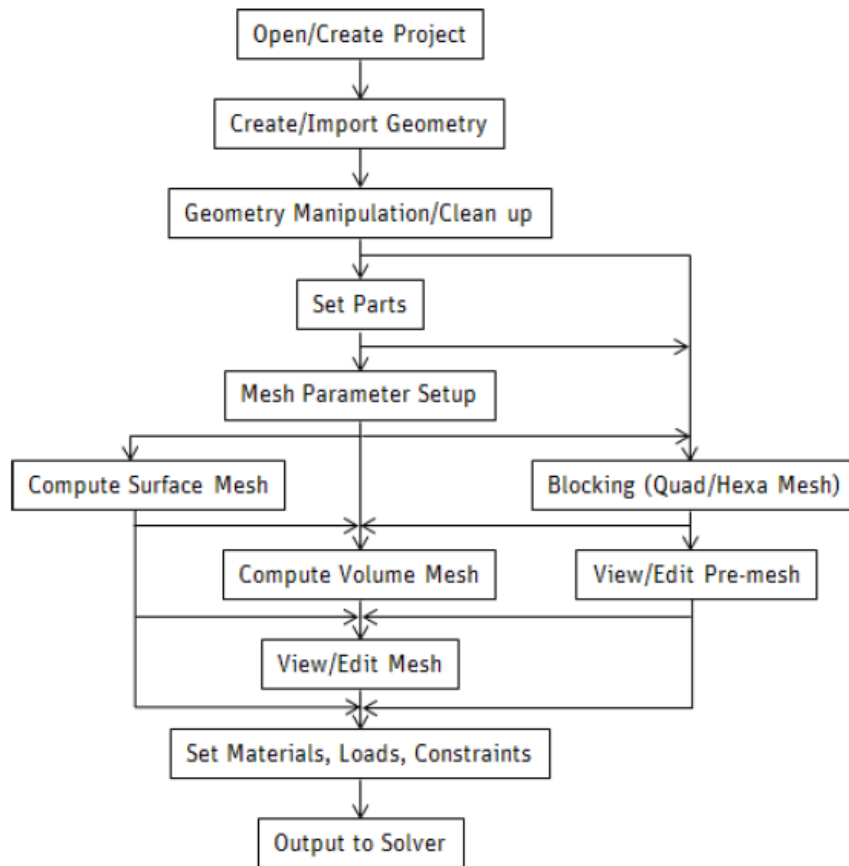


Figura 7.1. Proceso de generación de la malla [19]

A continuación, a modo de ejemplo, para mostrar el proceso seguido se comentará cómo se ha generado la malla de una de las configuraciones consideradas. Comenzamos la definición de la geometría de la configuración del problema, incluyendo la parte sólida y la parte líquida.

Primero introducimos los puntos que delimitan las fronteras mediante coordenadas. Continuamos uniendo los puntos con líneas (llamadas 'curves' en el programa), observamos el resultado en la figura 7.2.:

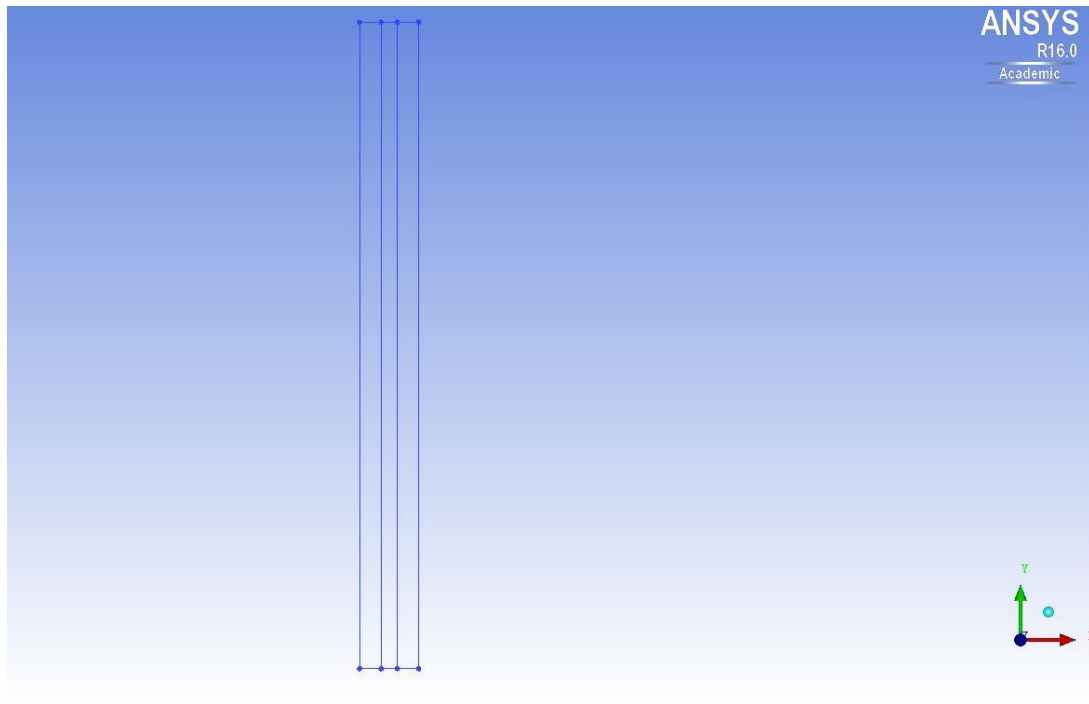


Figura 7.2. Puntos y líneas de la geometría

Ahora le damos a cada línea, es decir, a cada condición de contorno un nombre.

Una vez que tenemos definida la geometría de la configuración, debemos crear el bloque. Inicializamos un bloque plano. El bloque que se genera ocupa toda la geometría, por lo que hacemos dos cortes para tener tres que se ajusten a la geometría, llamamos al bloque del centro 'FLUID' y los del exterior 'SOLID'.

Disponiendo ya de los bloques que necesitamos, procedemos a asociar los puntos y líneas de la geometría con los vértices y bordes de los bloques, respectivamente.

Ya podemos generar la malla. Tenemos que definir los parámetros de los bordes de los bloques y así, creamos nuestra malla (figura 7.3.).

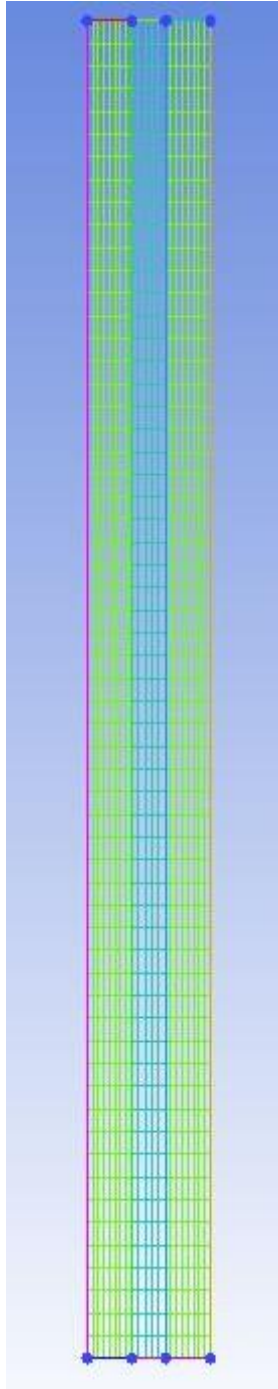


Figura 7.3. Malla obtenida

Por último, configuramos las unidades, seleccionamos las condiciones de contorno, elegimos el formato en el que queremos generar la malla (en nuestro caso formato de ANSYS Fluent, .msh) y ya tendremos la malla lista para exportarla.

Puesto que, como veremos posteriormente, OpenFOAM no permite una malla compuesta de diferentes materiales, debemos separarla en parte líquida y parte sólida. Esto se hace eliminando primero, por ejemplo, la parte líquida y exportando la

mallas del sólido, y viceversa. Llamamos a los dos archivos: 'fluid.msh' y 'solid.msh' (figuras 7.4.a. y 7.4.b.).

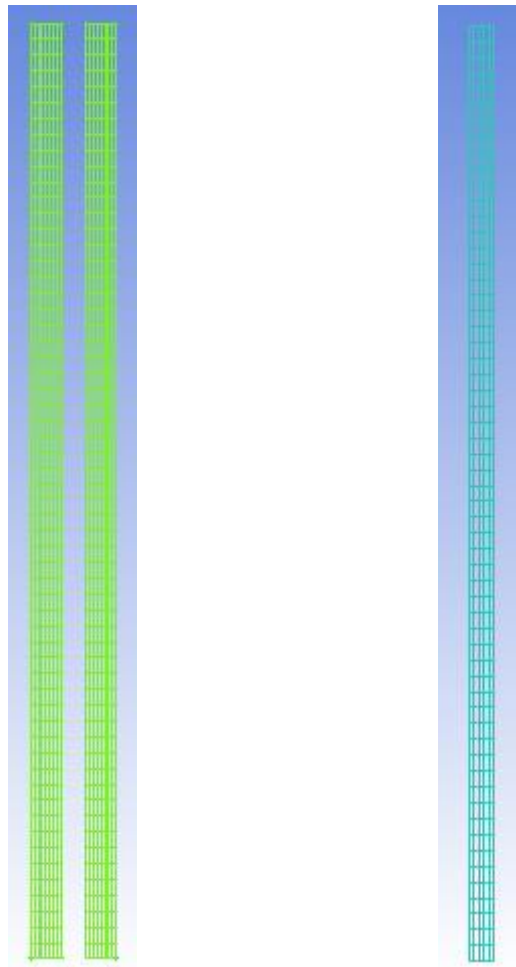


Figura 5.4. a) Malla solid.msh

b) Malla fluid.msh

Siguiendo este procedimiento, las dos mallas tienen una relación de posición entre ellas.

7.2. Conversión de la malla

OpenFOAM ofrece la posibilidad de generar mallas usando otros programas para, posteriormente, convertirlas en el formato que este software utiliza.

Puesto que, como hemos visto anteriormente, tenemos una malla para Fluent (.msh) debemos utilizar el convertidor `fluentMeshToFoam`. Esta aplicación no permite mallas de diferentes materiales por lo que la malla del líquido y la del sólido deberán de ser tratadas por separado. La correcta utilización de la aplicación es como sigue:

```
fluentMeshToFoam <meshFile>
```

Siendo <meshFile> un archivo de extensión '.msh'. Además, puesto que OpenFOAM trabaja con metros, deberemos aplicar una escala.

Sabiendo esto, creamos una aplicación propia para convertir automáticamente las mallas (del sólido y del líquido) y crear los sets, esto es acoplar ambas zonas para el intercambio de variables. Esto se hace escribiendo el código que se muestra a continuación en un archivo en blanco. Este archivo se situará en el directorio principal del caso con el que se está trabajando en OpenFOAM.

```
#!/bin/sh
# Source tutorial run functions
. $WM_PROJECT_USER_DIR /bin/tools/RunFunctions
# Get application name
application=`getApplication`

cd solid
runApplication fluentMeshToFoam -scale 0.001 solidMesh.msh
runApplication setSet -batch setBatch
runApplication setsToZones -noFlipMap

cd ../fluid
runApplication fluentMeshToFoam -scale 0.001 fluidMesh.msh
runApplication setSet -batch setBatch
runApplication setsToZones -noFlipMap

#IMPORTANT: Change the bonundaries property for an imported mesh.
#-writeSets -writeZones
# ----- end-of-file
```

Llamamos a esta aplicación escribiendo en la terminal en el directorio principal del problema:

```
./fluentMeshToFoam
```

8. Parámetros de los materiales

El objetivo de este apartado es dar una justificación de los valores escogidos para el sólido y en el líquido.

8.1. Líquido

El fluido cerebroespinal está constituido en una 99% por agua, por lo que tomar las propiedades de esta es una aproximación bastante acertada. Puesto que la temperatura del cuerpo humano oscila entre 35 y 37°C, tomamos las propiedades de densidad y viscosidad cinemática del agua a 36°C:

$$\rho_{CSF} = 993.6 \text{ kg/m}^3$$

$$\nu_{CSF} = 7.088 \cdot 10^{-7} \text{ m}^2/\text{s}$$

Con estas dos propiedades el líquido cerebroespinal queda totalmente definido.

8.2. Sólido

En lo referente al sólido existe una pequeña disparidad en los estudios referentes a este tema. Como se mencionó en el apartado de la mecánica de sólido, necesitamos dos parámetros para definir un sólido elástico lineal, isótropo y homogéneo; que son, el módulo de Young y el coeficiente de Poisson.

Observando estudios anteriores, los módulos de Young se sitúan entre 3kPa y 2MPa. El coeficiente de Poisson de la médula espinal se ha estudiado menos, y pocos estudios lo mencionan, este se encuentra entre 0.3 y 0.5.

Para la determinación de la densidad del sólido no se ha encontrado literatura. Pero tomando de referencia la densidad del líquido cerebroespinal y del hueso, 1900 kg/m³ establecemos la densidad de la espina dorsal como:

$$\rho_{SC} = 1.2 \cdot \rho_{CSF}$$

Solo quedaría determinar la porosidad, pero dado que en este estudio no abarcamos poroelasticidad, no será necesario. Aunque varios estudios previos toman como valor $\kappa = 1.4 \cdot 10^{-15} \text{ m}^2$.

9. Resolución usando el software OpenFOAM®

9.1. Introducción

OpenFOAM es una librería en lenguaje C++ que se compone de aplicaciones (ejecutables). Estas aplicaciones se dividen en: 'solvers', que son diseñados para resolver un problema específico de mecánica del medio continuo; y 'utilities', que realizan las tareas de manipulación de datos.

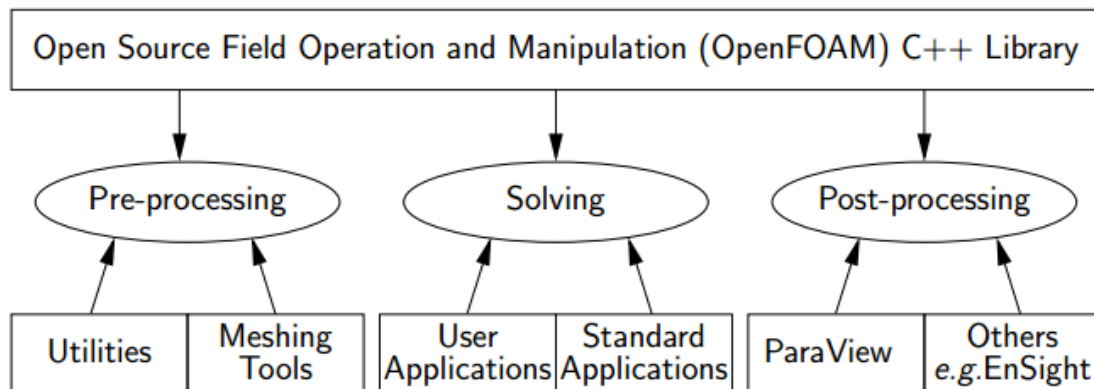


Figura 9.1. Estructura de OpenFOAM [21]

La figura 6.1 la estructura general de OpenFOAM que se divide en:

- Preprocesado
- Resolución
- Postprocesado

9.2. Instalación del programa

OpenFOAM trabaja sobre una distribución Linux. Por ello, debemos instalar primero un sistema operativo con esta base. En nuestro caso instalamos Ubuntu (64bits) 14.04 LTS en una partición de disco duro para poder ejecutar las simulaciones con el máximo rendimiento.

Una vez hecho esto, procedemos a descargar y compilar el programa siguiendo las instrucciones, se remite a bibliografía [20].

9.2.1. Instalación del paquete FSI

Esta versión de OpenFOAM no incluye el paquete FSI, por lo que este debe de ser compilado independientemente.

Primero tenemos que descargar y descomprimir el código fuente:

```
mkdir -p $WM_PROJECT_USER_DIR
cd $WM_PROJECT_USER_DIR
wget http://openfoamwiki.net/images/5/52/Fsi_31.tar.gz
c -xzf Fsi_31.tar.gz
```

A continuación, hacemos unas modificaciones:

```
cd FluidStructureInteraction/src/fluidStructureInteraction/stressModels/constitutiv
eModel

wget https://raw.githubusercontent.com/wyldckat/FluidStructureInteraction/fe31/src/
fluidStructureInteraction/stressModels/constitutiveModel/constitutiveModel.C -O con
stitutiveModel.C

wget https://raw.githubusercontent.com/wyldckat/FluidStructureInteraction/fe31/src/
fluidStructureInteraction/stressModels/constitutiveModel/constitutiveModel.H -O con
stitutiveModel.H

cd ../../..
```

Escribimos el comando './Allwmake' y cuando acabe de ejecutarse tendremos listo el programa y el paquete FSI listos para trabajar.

9.2.2. Instalación de ParaView®

ParaView es el programa de análisis y visualización de datos que usaremos para leer los resultados de OpenFOAM. Para la instalación debemos escribir en la terminal lo siguiente:

```
sudo apt-get update
sudo apt-get install paraview
```

9.3. Sistema de archivos de OpenFoam®

Como ya sabemos OpenFOAM es solo una biblioteca en lenguaje C++. Por esto, la forma de trabajar es mediante archivos. El usuario asigna un nombre al caso con el que va a trabajar que será el nombre del directorio que contendrá los datos del problema. Este directorio se recomienda que esté dentro del subdirectorio 'run' pero puede estar en cualquier lugar.

La estructura básica del directorio del caso con el que se va a trabajar es la que se muestra en la figura 9.2.

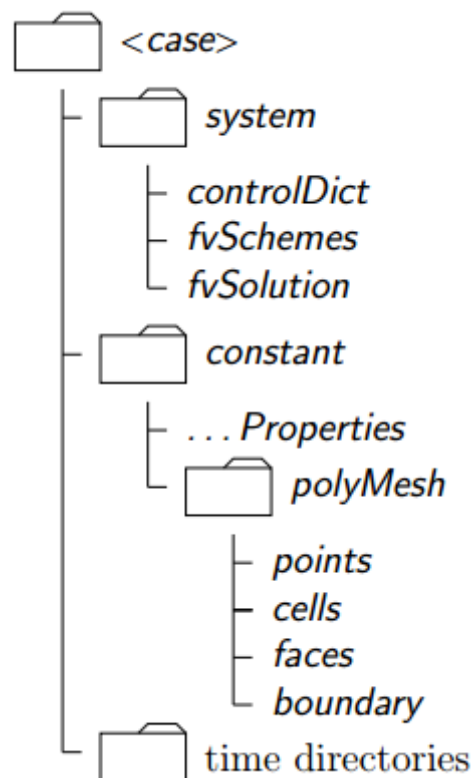


Figura 9.2. Estructura de archivos de OpenFOAM [21]

En el caso de un problema de fluido-estructura, el directorio cuenta con dos subdirectorios para el líquido y el sólido, que son 'fluid' y 'solid'. Estas carpetas, a su vez se componen de los subdirectorios de los casos convencionales:

- 'system': contiene los parámetros asociados a la solución, como son por ejemplo, el método numérico empleado, el paso de tiempo, etc.
- 'constant': se incluyen la descripción de la malla y las propiedades físicas de esta.
- Directorios de tiempo: antes de iniciar la simulación, tenemos el directorio '0' que indica este tiempo. Conforme avance la simulación se crearán archivos con cada tiempo que queramos que se genere. El directorio '0' contiene las

condiciones iniciales y las condiciones de contorno que el usuario debe especificar.

Para un caso del paquete FSI, la estructura de archivos es ligeramente distinta. Veamos en la figura 9.3 el sistema de archivos que tendremos para nuestras simulaciones:

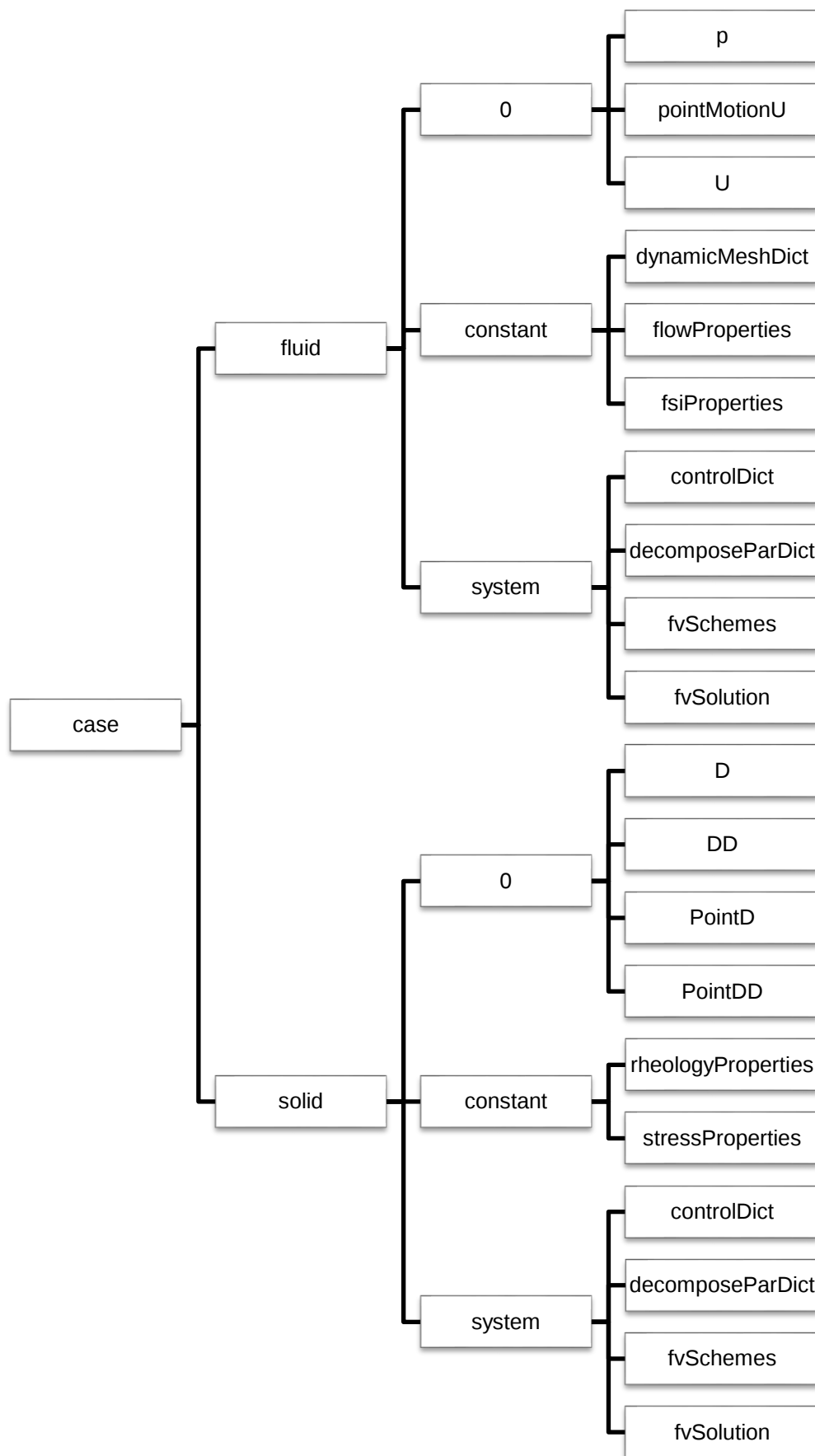


Figura 9.3. Estructura de archivos de cada caso

10. Solving

En este apartado se describe el proceso de resolución que se lleva a cabo: el control del tiempo, la salida de datos, esquema numérico, la aplicación que se utiliza ('solver'), y el control y algoritmo de la solución. Esto se aplica en el directorio 'system' definido anteriormente.

10.1. Control de entrada y salida ('controlDict')

10.1.1. Control del tiempo

- 'startFrom': aquí definimos que queremos empezar desde 'startTime', que es fijado a 0.
- 'stopAt': este apartado es equivalente al anterior.
- 'deltaT': define el paso de tiempo de la simulación, que se ha demostrado que funciona bien con un valor de 0,0001. Además, en 'adjustTimeStep' seleccionamos que el paso de tiempo no sea variable, por lo que no se tendrá en cuenta el número de Courant.

10.1.2. Control de la salida

- 'writeControl': escribe datos cada número de veces que indica el escalar 'writeInterval'.
- 'writeFormat': indicamos que el formato de escritura sea ASCII.
- 'writePrecision': establecemos una precisión de 10.
- 'writeCompression': indica si queremos que los datos generados se compriman o no, como es nuestro caso.
- 'timeFormat': especificamos el formato general por defecto.
- 'timePrecision': establece la precisión del tiempo.
- 'runTimeModifiable': en ese apartado se indica si queremos o no que se lea el archivo 'controlDict' al comienzo de cada paso de tiempo para que podamos cambiar los parámetros en cualquier momento de la simulación.

10.1.3. Librería y funciones

En este archivo también llamamos a las librerías que son necesarias cargar. Y también a las funciones. Una de las funciones que hemos llamado es 'probes'. Esta función permite conocer los datos de las propiedades que indiquemos en los puntos de tiempo y posición que queramos. Estos datos se escriben en un documento para su posterior procesamiento.

10.2. Esquema numérico ('fvSchemes')

En este archivo se establecen los esquemas de discretización que se aplican: de interpolación, del gradiente, de los gradientes normales a las caras, esquema del laplaciano, de la divergencia y de tiempo.

Nos hemos ayudado del tutorial 3dTube que incorpora OpenFOAM para definir los esquemas de discretización. Esto se aplica tanto para el líquido como para el sólido.

10.2.1. Esquema de tiempo

La discretización del término de la primera derivada respecto al tiempo $\left(\frac{\partial}{\partial t}\right)$ se especifica en el sub-diccionario 'ddtSchemes'. Tenemos indicado un esquema 'backward' que es de segundo orden e implícito (incondicionalmente estable).

10.2.2. Esquema del gradiente

El esquema se indica en el sub-diccionario 'gradSchemes'. Por defecto tenemos como esquema 'leastSquares', es decir, la técnica de análisis numérico de mínimos cuadrados de segundo orden.

10.2.3. Esquema de divergencia

Este se indica en el sub-diccionario 'divSchemes'. El término convectivo típico es $\nabla \cdot (\rho U U)$ que se representa en OpenFOAM como $\text{div}(\phi, U)$ siendo $\phi = \rho U$.

El esquema de Gauss es la única elección posible de discretización que, a su vez, también requiere un esquema de interpolación para el campo dependiente, en nuestro caso es la velocidad.

10.2.4. Esquema del laplaciano

El sub-diccionario 'laplacianSchemes' es el que contiene los parámetros de discretización del laplaciano Δ ó ∇^2 . Uno término común en la ecuaciones de mecánica de fluidos es $\nabla \cdot (\nu \nabla U)$, que se representa como $\text{laplacian}(\nu, U)$.

En este caso, el esquema de Gauss es otra vez la única opción. Se necesita también un esquema de discretización de interpolación para el coeficiente de difusión (ν en el ejemplo) y uno de superficies normales a los gradientes (para ∇U en el ejemplo) que veremos más adelante.

10.2.5. Esquema de interpolación

El sub-diccionario 'interpolationSchemes' incluye las interpolaciones que se pueden realizar desde el centro de la celda al centro de una cara.

10.2.6. Esquema de discretización de los gradientes normales a las caras

Este esquema se aplica con el sub-diccionario 'snGradSchemes'. El gradiente es evaluado en la cara de la celda.

10.2.7. Cálculo de flujo

Por último, existe otro sub-diccionario no mencionado anteriormente 'fluxRequired'. Aquí se nombran los campos por donde el flujo es generado en la aplicación.

10.3. Control de la solución ('fvSolution')

Las ecuaciones de los 'solvers', tolerancia y algoritmos se establecen en el archivo 'fvSolution' en la carpeta 'system'. Este archivo contiene a su vez una serie de sub-diccionarios que se explican a continuación.

10.3.1. 'Solvers'

En este sub-diccionario se especifica que 'solver' lineal se usa para cada ecuación discretizada.

Cada entrada dentro del sub-diccionario se corresponde con la variable que se está evaluando (p, pointMotionU, U, D, DD, PointD, PointDD).

Existen cinco 'solvers' lineales. Por defecto tenemos PCG o PBiCG de las iniciales en inglés 'Precondicioned (Bi-)Conjugate Gradient', el primero se usa para mallas simétricas y el segundo para asimétricas. También se emplea GAMG para geometrías generalizadas algebraicas.

10.3.2. Factores de relajación

Estos factores controlan la relajación, esto es una técnica para mejorar la estabilidad de la simulación, sobre todo en problemas estacionarios. Para nuestro caso, un problema transitorio, no hay necesidad de aplicarlo.

10.3.3. PISO y SIMPLE

Estos algoritmos iterativos se usan para resolver ecuaciones para velocidad y presión. Evalúan soluciones iniciales para después corregirlas. PISO se usa para problemas transitorios y SIMPLE, para estacionarios. En nuestro caso no usamos ninguno.

10.4. Procesado en paralelo

Existe un método de procesado en paralelo que se define en el archivo 'descomposeParDict' dentro del directorio 'system'. Este método consiste en descomponer la geometría y los campos asociados a ella en diferentes partes para resolverlas en distintos procesadores. Al ser una malla multibloque ya está dividida por las zonas de contorno definidas como entrefase.

10.5. Aplicación: fsiFoam

Como definimos anteriormente existen dos formas de acoplamiento del problema: fuerte y débil. El paquete FSI de OpenFOAM aplica el acoplamiento fuerte entre el líquido y el sólido con la aplicación fsiFoam que es el ejecutable con el que nosotros trabajaremos.

La aplicación que vamos a ejecutar se indica en el archivo 'controlDict' como se muestra en el anexo.

11. Configuraciones

Como mencionamos en apartados anteriores, vamos trabajar con geometrías y disposiciones simplificadas en dos dimensiones. Una de las configuraciones consiste en considerar los dos canales exteriores a la médula suponiendo esta deformable (llamada configuración A).

Otra configuración es considerar solo un canal alrededor de la médula, despreciando el otro, que fluye entre dos paredes deformables o solo una.

Una alternativa a la anterior es añadir una parte sólida y deformable en el fondo lo que constituye una configuración cerrada. Esto nos dará una idea de cómo se comporta el LCR al final de la espina dorsal.

Estas configuraciones han sido las más dignas de mención, aunque ha habido más. Además, hemos trabajado con varias medidas para garantizar unos resultados correctos, pero siempre dentro de valores acordes a la realidad.

Por otro lado, establecimos distintas condiciones de contorno en la entrada y salida del flujo y en sólido, con condiciones de pared móvil y/o de deslizamiento.

Finalmente, también se han usado varios valores de las propiedades. Todo esto nos ha permitido obtener resultados adecuados que se presentan en el apartado siguiente

12. Análisis genérico

Las configuraciones que estudiamos a continuación tienen unas condiciones de contorno concretas que se definen en cada apartado y en los anexos se indicarán. No obstante, a la entrada siempre tendremos una onda de presión de amplitud la unidad.

Este apartado incluye los resultados de las simulaciones; y la presentación e interpretación de dichos resultados obtenidos.

12.1. Configuración A

La primera configuración que estudiamos dispone de una geometría líquido-sólido-líquido como hemos mencionado anteriormente. La geometría tiene una altura de 60mm y un ancho de los canales de 4mm; para el sólido tenemos un ancho de 10mm con un módulo de Young igual a 3MPa.

Se observan para concretos instantes de tiempo como se distribuyen las presiones y velocidades en ParaView (Figura 12.1.a., Figura 12.1.b, Figura 12.2.a., Figura 12.2.b, Figura 12.3.a., Figura 12.3.b, Figura 12.4.a., Figura 12.4.b).

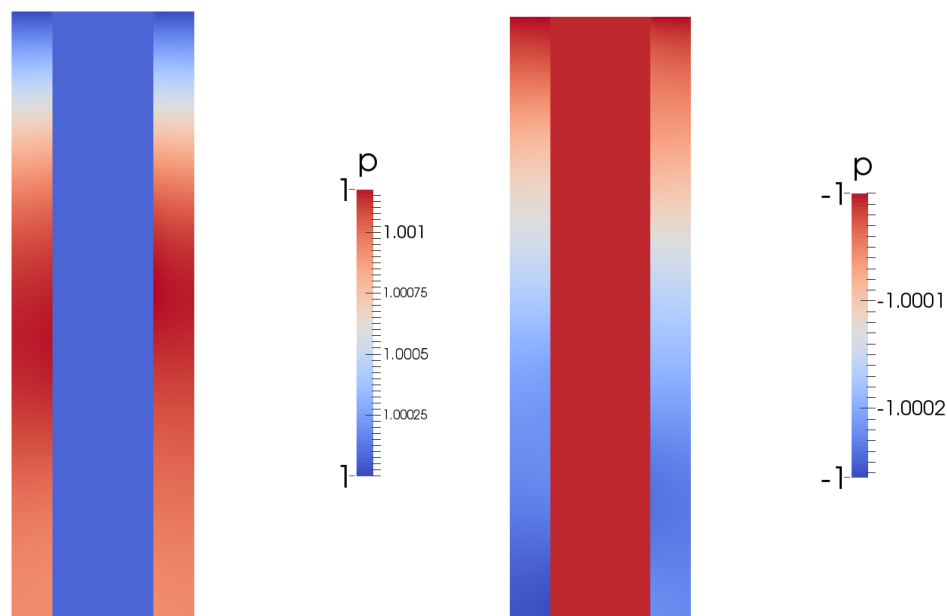


Figura 12.1. a) Distribución de la presión en el instante de tiempo $t=8s$ (izquierda) b) Distribución de la presión para el instante de tiempo $t=8.5s$ (derecha)

En estos instantes de tiempo se alcanzan el máximo y el mínimo valor de presión, el valor absoluto de la amplitud, es decir, un pascal. Se observa cómo se distribuye la presión a lo largo de esta configuración de la médula espinal.

A penas existen variaciones de presión desde la entrada hasta la salida del fluido, pero la presión es mayor conforme nos acercamos a la salida en ambos casos.

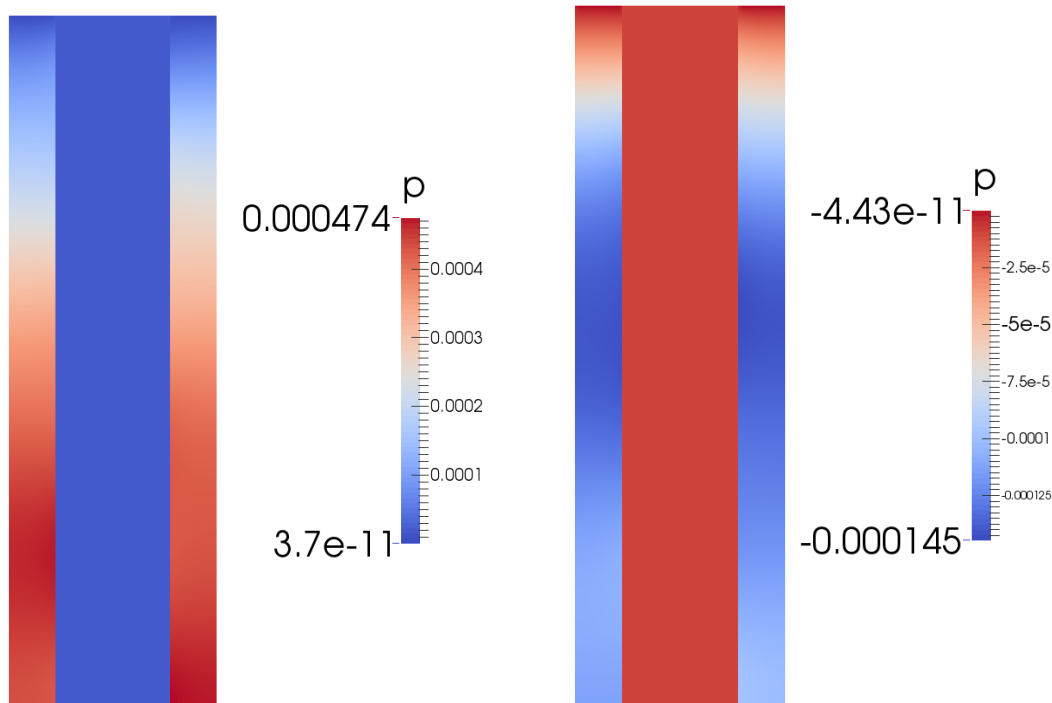


Figura 12.2. a) Distribución de la presión en el instante de tiempo $t=8.25s$ b) Distribución de la presión para el instante de tiempo $t=8.75s$

En los intervalos descritos entre los instantes de tiempo anteriormente mencionados, la presión es mayor entre el 'inlet' y el 'outlet'. Tomemos, por ejemplo, los instantes que se encuentran en el centro de dicho intervalo, es decir, en los segundos y un cuarto; y en los segundos y tres cuartos. La variación de presión en estos puntos la mayor. Como veremos posteriormente, estos puntos coincidirán con los máximos y mínimos de velocidad en el eje 'y'.

A continuación, mostraremos esta variación de presión a lo largo del eje 'y' en la figura 12.3.a. y 12.3.b. en los instantes de tiempo mencionados. Esto se hace tomando una línea vertical y representando los valores de presión a lo largo de ésta. Obtenemos como resultado unas gráficas de presión frente a la longitud de la línea descrita.

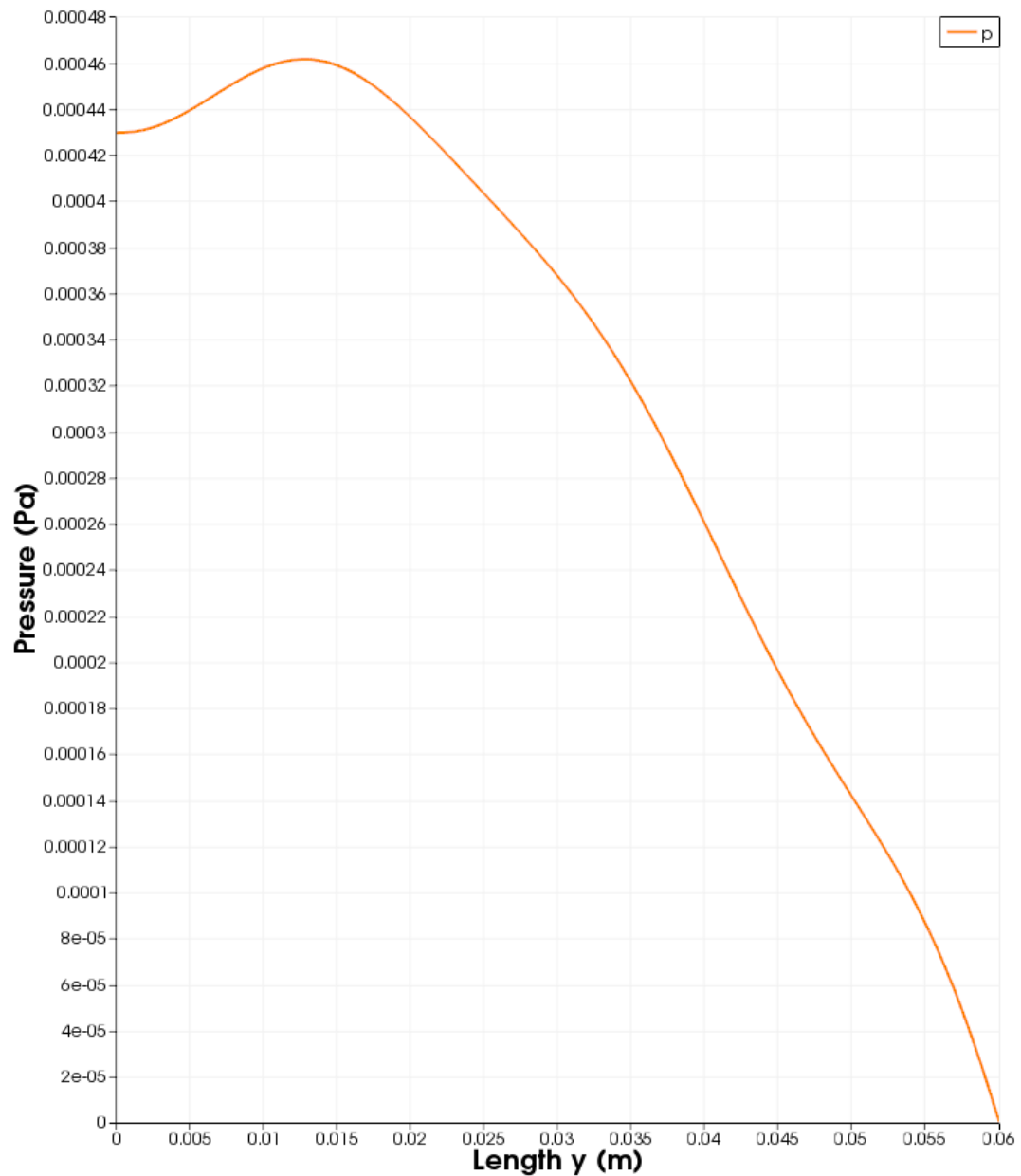


Figura 12.3.a. Variación de presión a lo largo de una línea vertical en $t=8.25s$

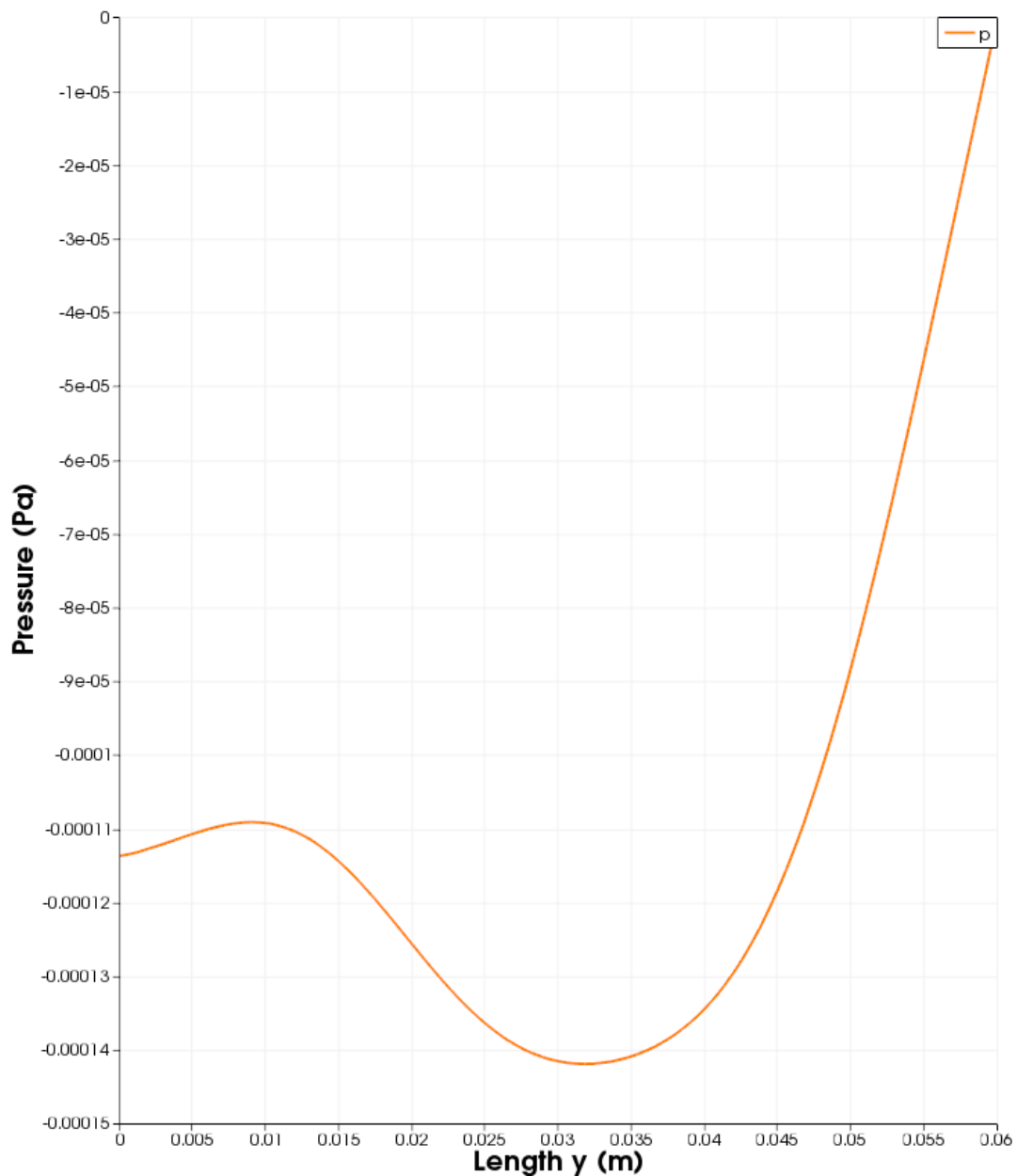


Figura 12.3.b. Variación de presión a lo largo de una línea vertical en $t=8.75s$

Para ambos instantes de tiempo la presión evoluciona desde un determinado valor negativo hasta cero. Extraemos como resultado que cerca del 'outlet' la tendencia es parabólica hasta que se alcanza una linealidad.

La onda de presión viaja a lo largo de la longitud aguas arriba y aguas abajo, y eso introduce una perturbación en el sólido y por lo tanto, en el volumen de control líquido que puede explicar esta distribución irregular al principio.

A continuación, estudiaremos la velocidad en la dirección del eje 'y', seguiremos el mismo procedimiento de estudio que para la presión. Se muestra la distribución de velocidades en las figuras 9.4 y 9.5.

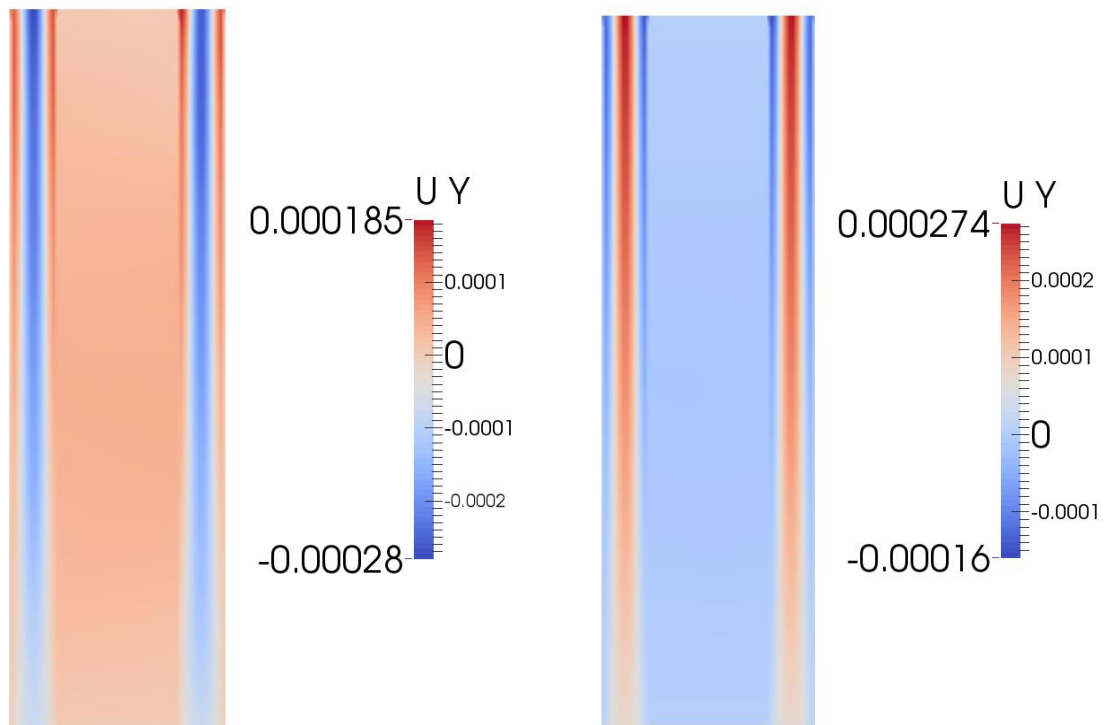


Figura 12.4.a) Distribución de la velocidad en el instante de tiempo $t=8s\ 9.4$. b) Distribución de la velocidad para el instante de tiempo $t=8.5s$

En estos instantes de tiempo, observamos que existe un flujo en la entrada en ambos sentidos. Para segundos enteros, es decir, sin decimales, tenemos un flujo que entra por el centro, y un flujo que sale por las zonas más exteriores. En segundos enteros y medio, tenemos el caso opuesto: flujo de entrada por el exterior y flujo de salida por el interior. Se observa de forma definida en las figuras 9.6.a y c.

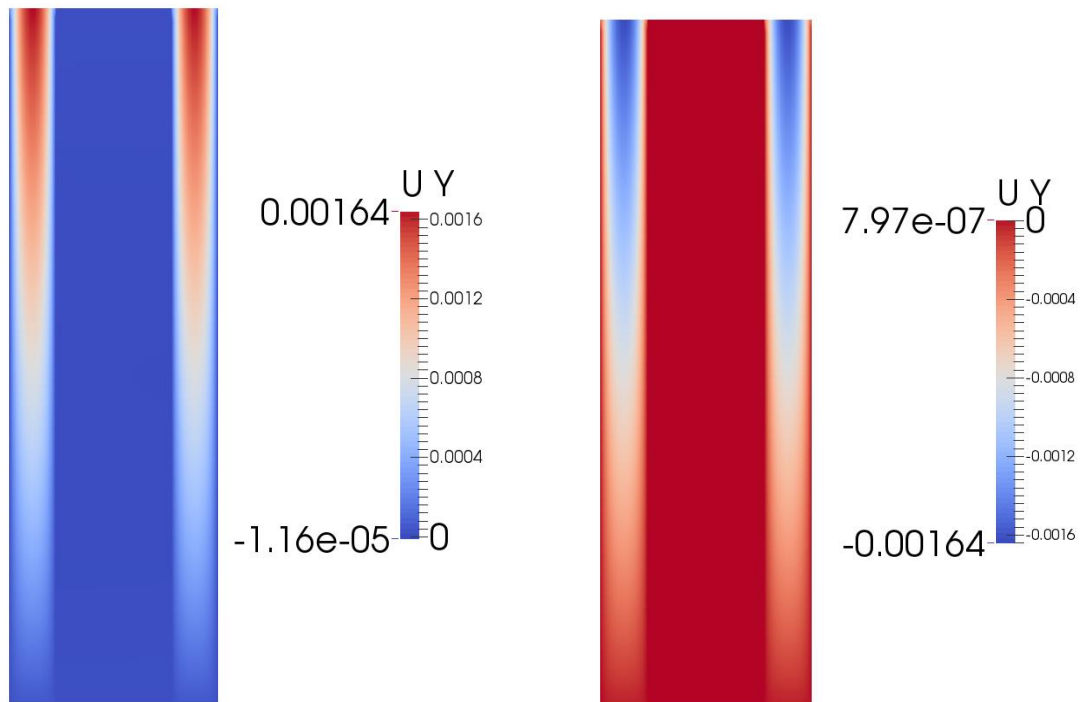


Figura 12.5.a) Distribución de la velocidad en el instante de tiempo $t=8.25s$ b) Distribución de la velocidad para el instante de tiempo $t=8.75s$

En estos instantes de tiempo tenemos un perfil de velocidades que presente en un flujo laminar en tubo. Se observa de forma definida en las figuras 9.6.b y d.

Ahora estudiamos las formas de los perfiles de velocidad tanto a la entrada ('inlet') como a la salida ('outlet') de nuestro caso.

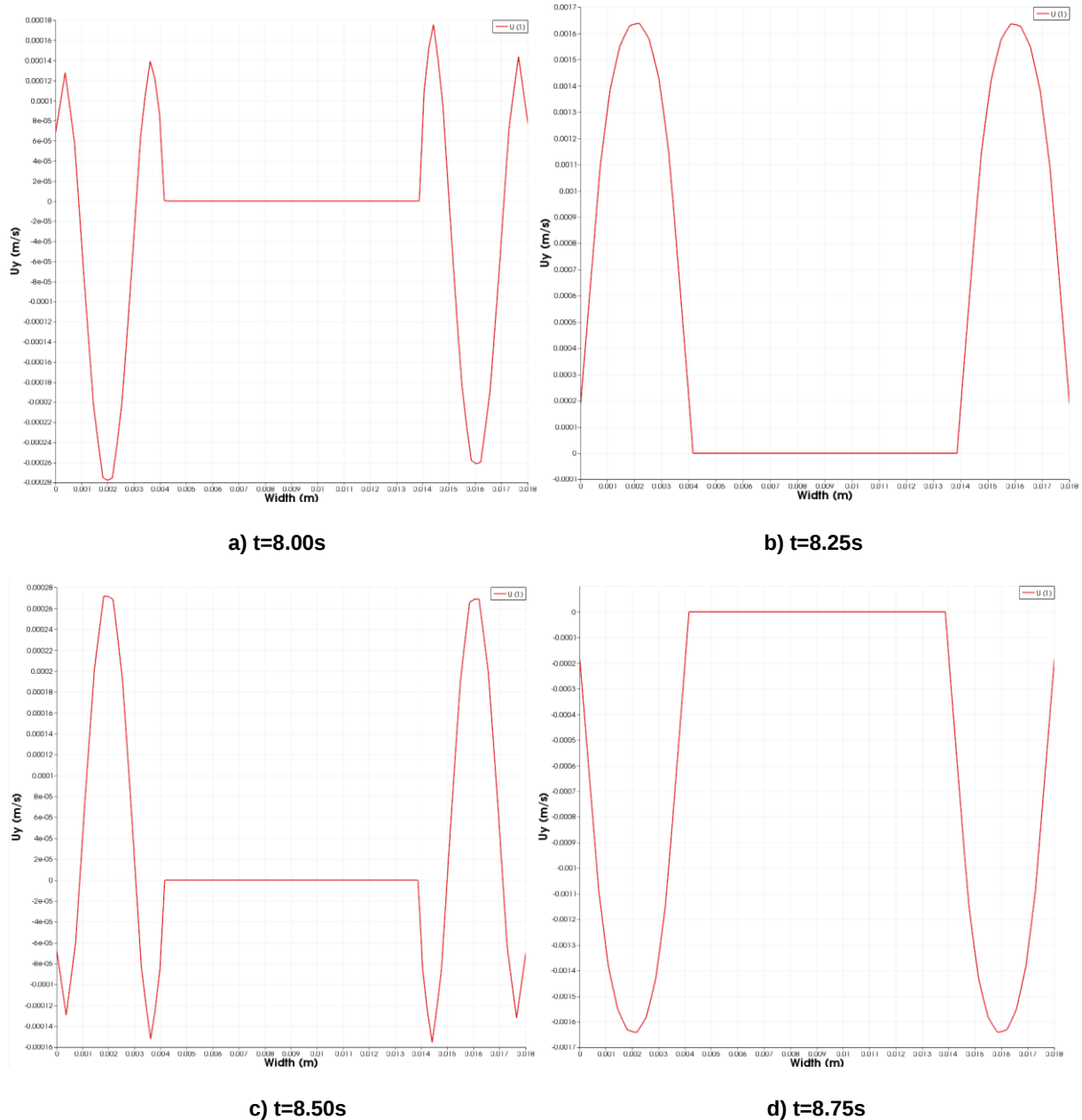


Figura 12.6. Perfil de la velocidad de entrada en distintos instantes de tiempo

Si observamos las figuras 12.6.a., b., c., y d. comprobamos el perfil de velocidades que habíamos descrito al observar las figuras 12.4.a. y b, 12.5.a. y b. Vemos la forma del perfil del flujo inverso.

Por otro lado, tenemos los perfiles de velocidad a la salida, los cuáles se muestran en la figura 12.7.a. , b., c., y d.

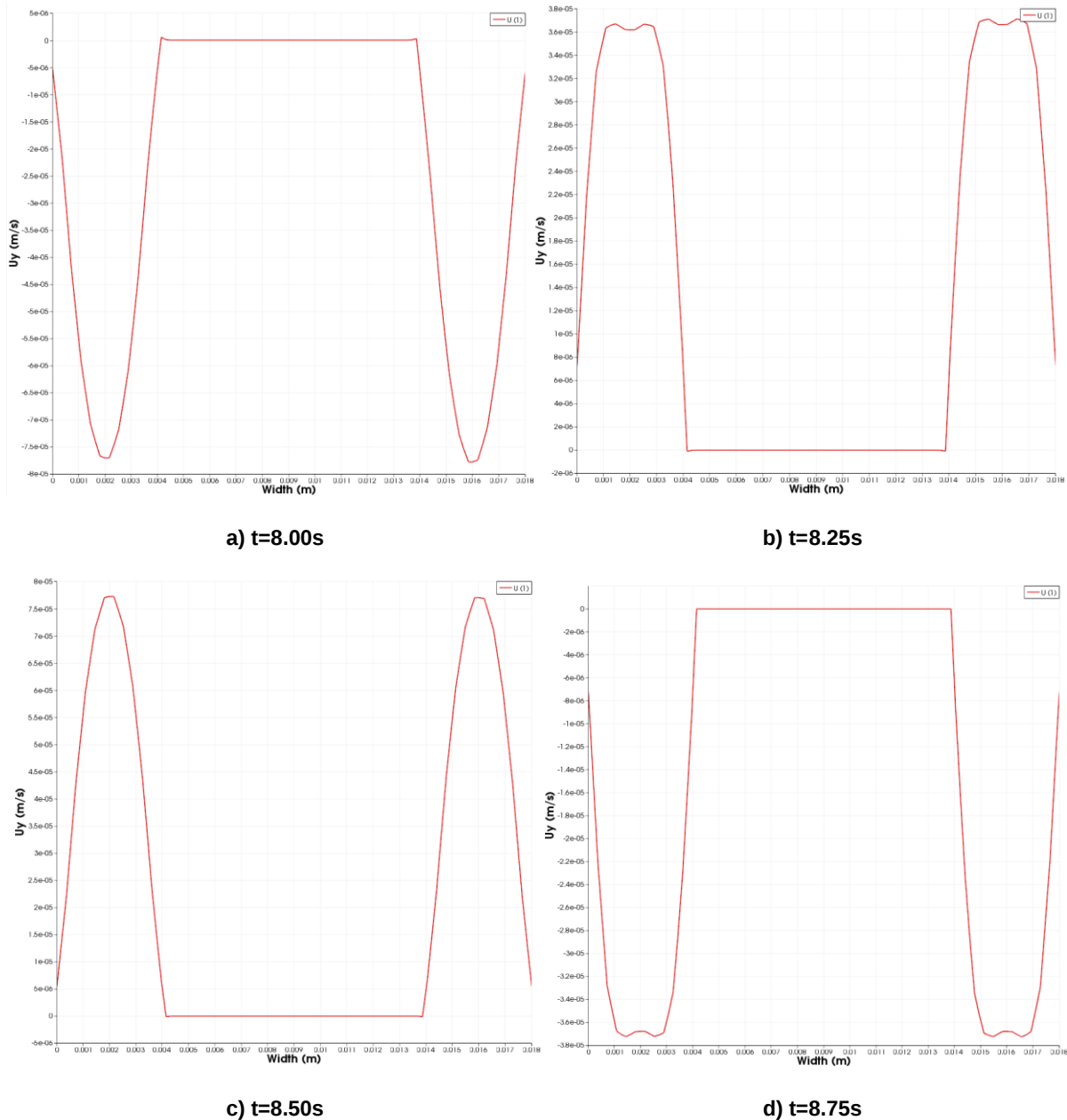


Figura 12.7. Perfil de la velocidad de salida en distintos instantes de tiempo

El flujo a la salida es negativo a lo largo de todo el periodo, es decir, que solo sale masa.

A lo largo de todas las imágenes y gráficas comentadas se ha observado una asimetría entre los dos flujos a pesar de que las condiciones de entrada son idénticas. Esta asimetría puede ser producidas o bien por las interferencias que se producen en el cálculo, o bien por las vibraciones que puede haber podido experimentar el sólido.

Por último, estudiamos los desplazamientos que se producen. El valor máximo de desplazamiento y deformación se alcanzará, como es obvio, cuando estamos en

un máximo o un mínimo de la onda de presión. La figura 9.8 nos muestra un instante donde se alcanza un pico de presión (las unidades son en metros).

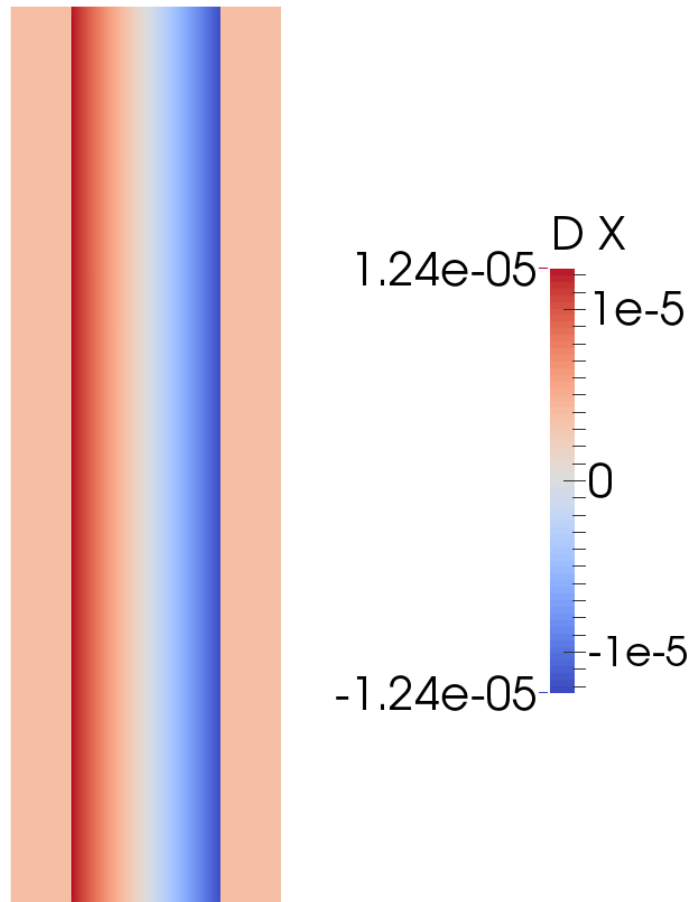


Figura 12.8. Valores de desplazamiento en el eje x para $t=6.00s$

El desplazamiento máximo es en el eje x es de $0.0124mm$. Como es de esperar para esta disposición, el desplazamiento principal se presentará en esta dirección.

12.1.1. Solución promedio – ‘Steady streaming’

Estos son los puntos que hemos definido en el directorio ‘system’ en los que se recogen los datos de presión y temperatura para dichos puntos. La distribución de los puntos esta esquematizada en la figura 12.9. OpenFOAM genera dos archivos, uno para la presión y otro para la velocidad que tratamos posteriormente en MatLab.

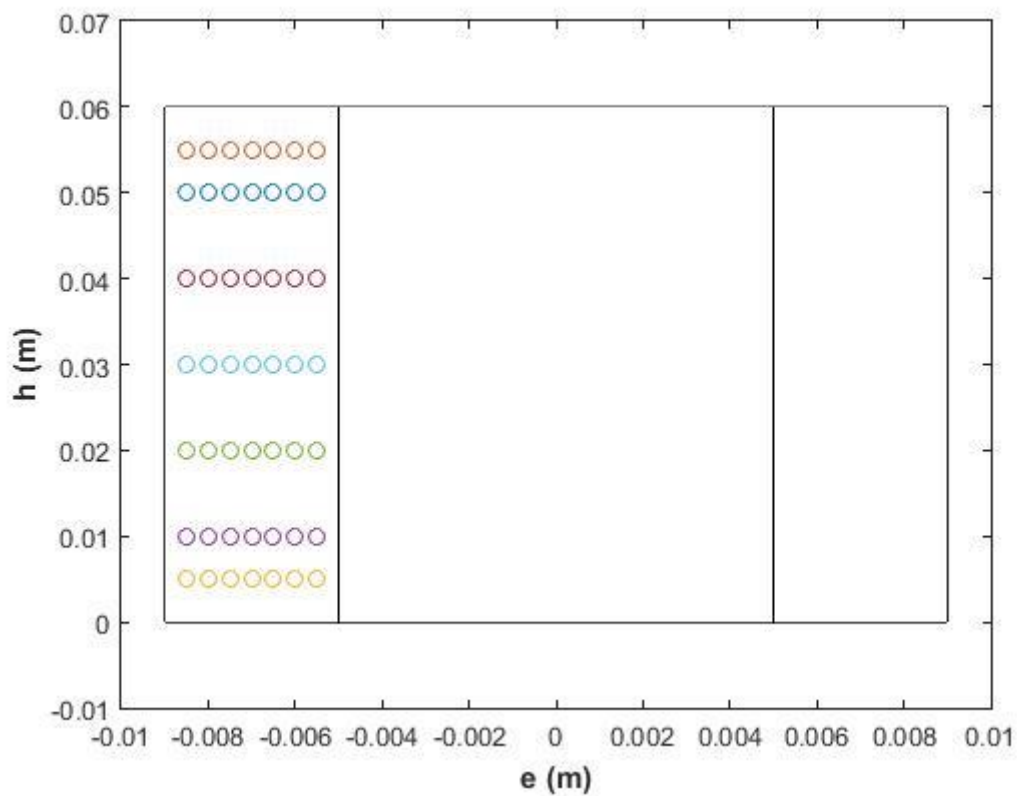


Figura 12.9. Esquema de la distribución de los puntos ('probes')

Primero, comenzamos leyendo el archivo de presión para observar cómo cambia la presión frente al tiempo (figura 12.10).

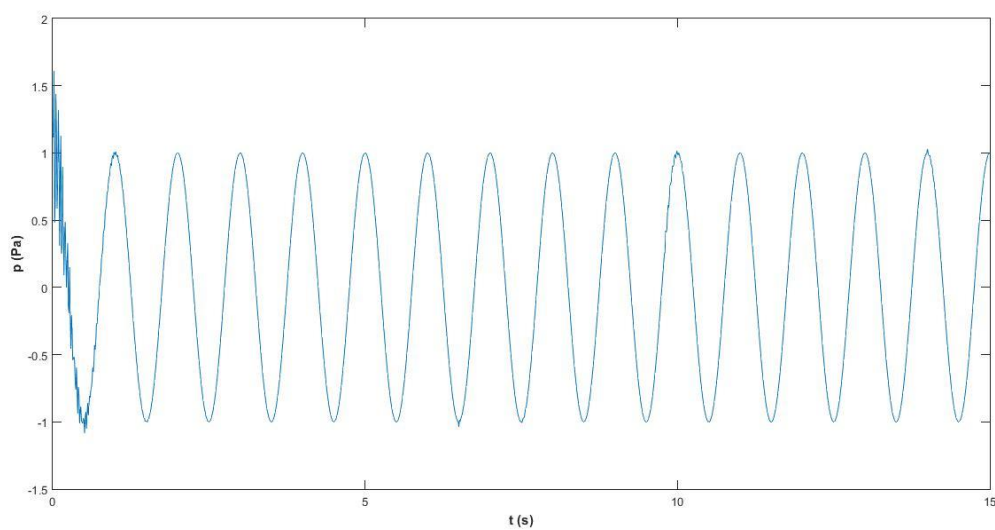


Figura 12.10. Variación de la presión en el punto ('probe') central

En la figura 12.10 se ha mostrado la variación de presión respecto al tiempo en un punto intermedio de la geometría. De esta gráfica extraemos las interferencias que existen.

Para la velocidad tendremos interferencias similares por lo que deberemos coger un tramo con pequeñas interferencias y filtrarlo para trabajar con él. Hemos tomado un tramo de tiempo de la simulación de 2.5 a 6.5 segundos para ajustar la onda con interferencias a una función senoidal de la forma:

$$f(t) = a_1 \cdot \sin(2\pi \cdot t + b_1)$$

(Fórmula 12.1)

Una vez que tenemos la función, como en la figura 12.11. calculamos la media para obtener la velocidad en dirección de eje 'y' para cada probe.

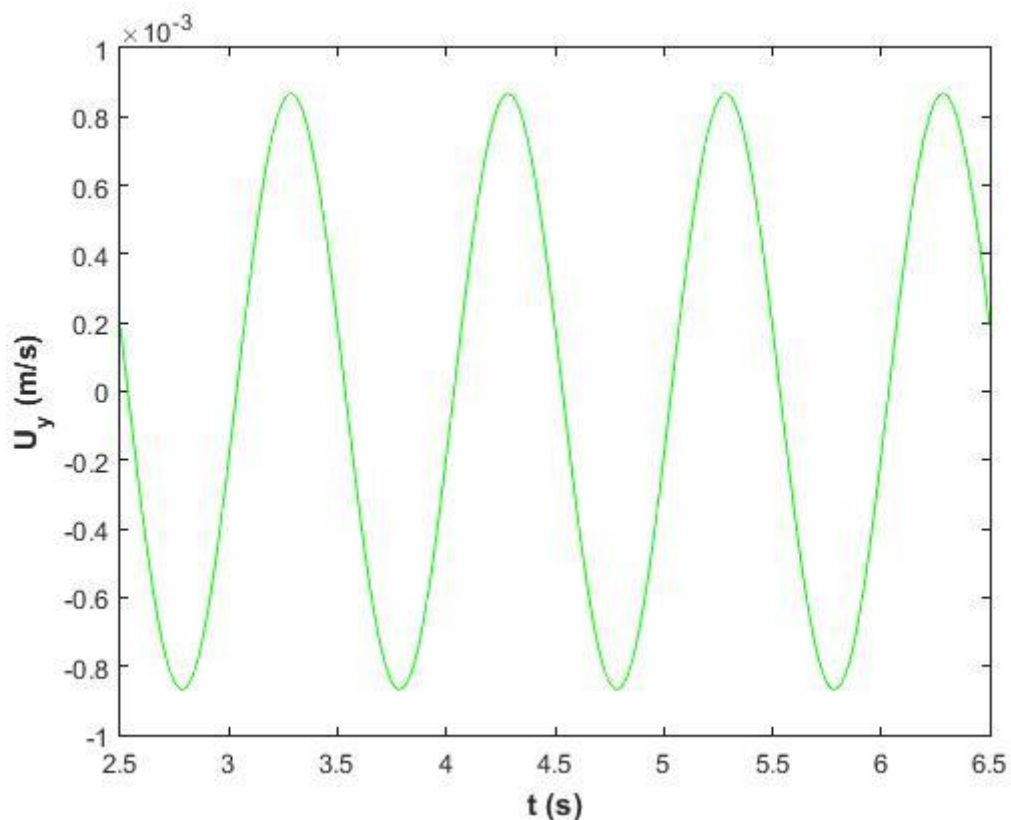


Figura 12.11. Ajuste de la onda de velocidad para un probe

Si tenemos la altura adimensional $y^+ = \frac{y}{L}$ (no confundir con la distancia adimensional a una pared), escogemos tres alturas distintas. Calculamos la media para cada punto y las representamos en las figuras 12.12.a., b. y c. Y los organizamos en la tabla 12.1.

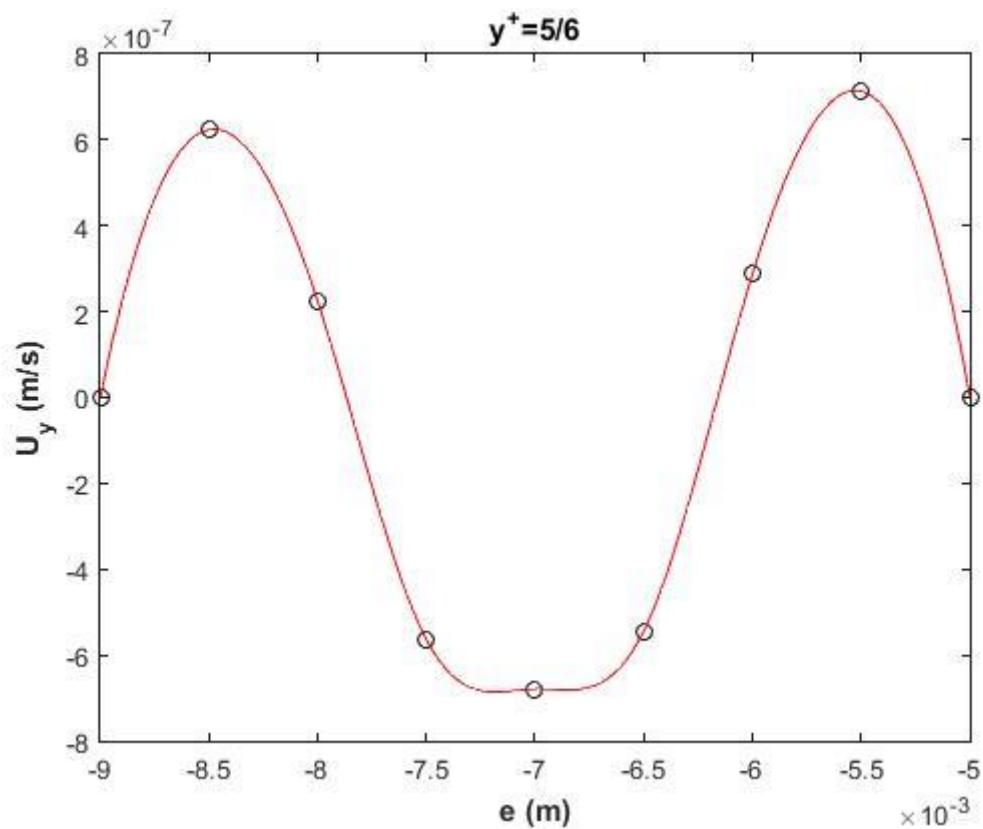


Figura 12.12.a. Velocidades medias de los probes en $y^+ = \frac{5}{6}$

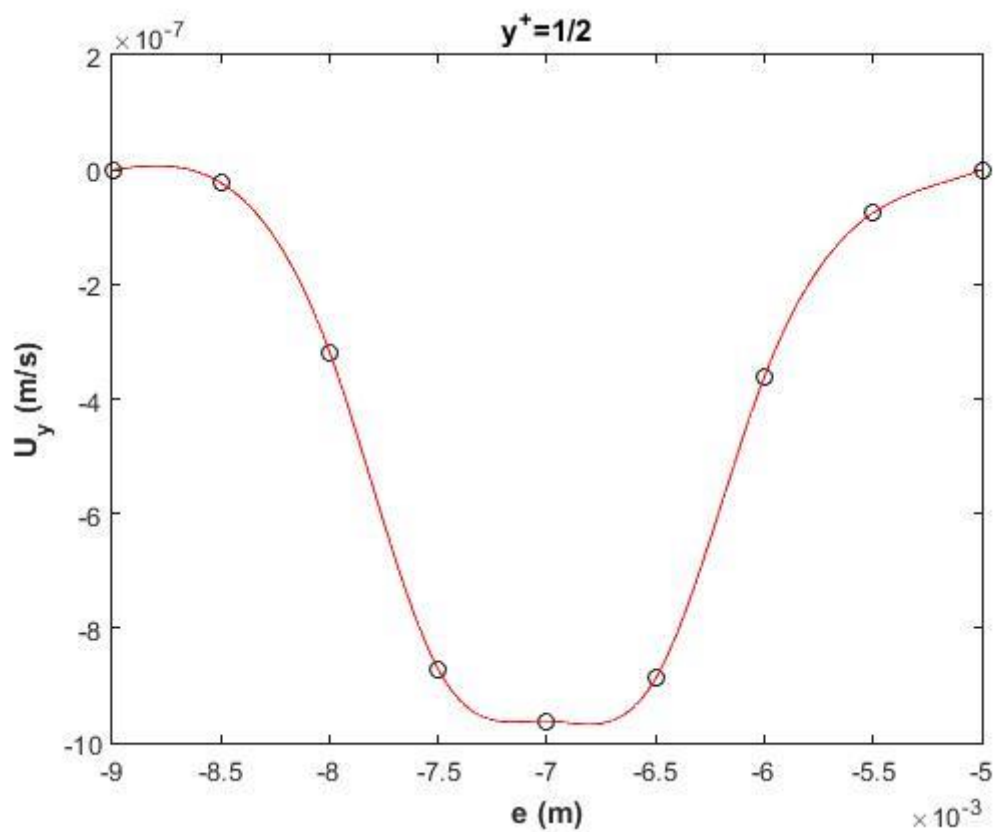


Figura 12.12.b. Velocidades medias de los probes en $y^+ = \frac{1}{2}$

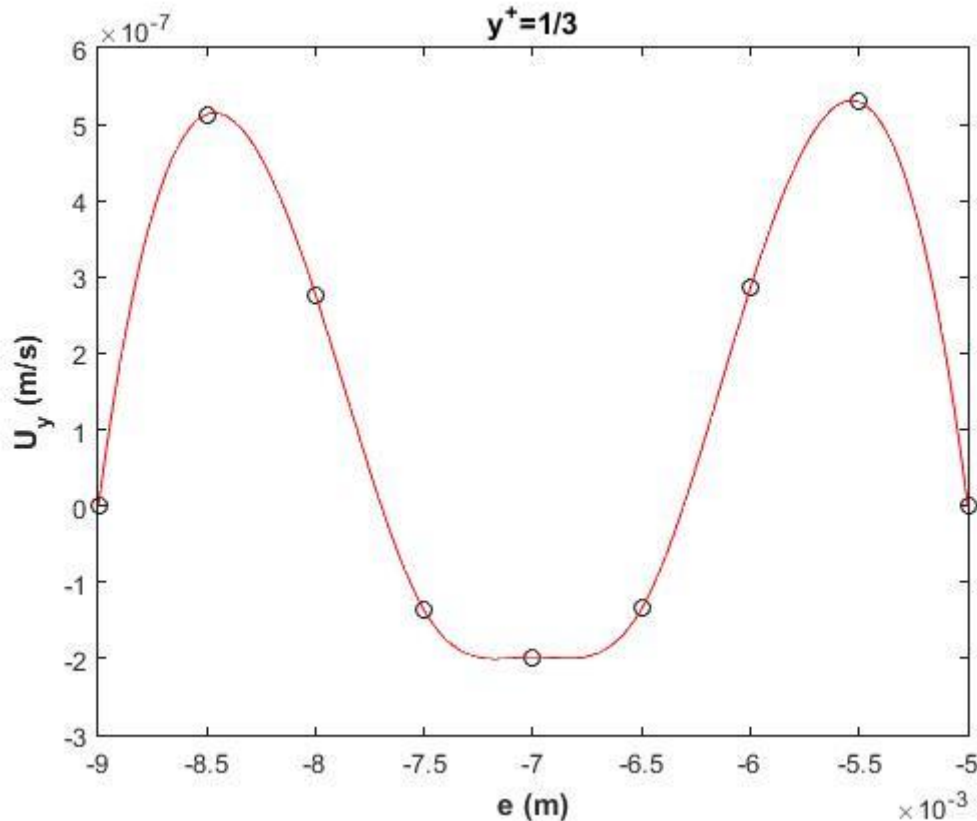


Figura 12.12.c. Velocidades medias de los probes en $y^+ = \frac{1}{3}$

Puesto que, al fin y al cabo, lo que tenemos es un conducto con una onda de presión que varía con el tiempo a la entrada y que su valor medio es nulo, el flujo medio tiene que ser cero también. Sabiendo esto, la integral del perfil de velocidades medio tiene que ser aproximadamente cero. Como vemos esto no se cumple totalmente como se ve claramente en la figura 12.12.b. Esto se debe a que hemos supuesto cero la velocidad en los extremos, pero dado que la pared de la derecha (en este caso) es móvil puede que esto no sea así.

Teniendo en cuenta esto, el resultado que extraemos es que el líquido de estudio tiende a bajar por el centro y a subir por los lados.

	$U_y \cdot 10^7 \text{ (m/s)}$						
	Probe 1	Probe 2	Probe 3	Probe 4	Probe 5	Probe 6	Probe 7
$y^+ = \frac{5}{6}$	6,2381	2,2381	-5,6398	-6,7876	-5,4363	2,8821	7,1357
$y^+ = \frac{1}{2}$	-0,2274	-3,1901	-8,7223	-9,6168	-8,8701	-3,6073	-0,7535
$y^+ = \frac{1}{3}$	5,1316	2,7568	-1,3712	-1,9768	-1,3403	2,8643	5,2938

Tabla 12.1. Velocidades medias en el eje 'y' de los probes estudiados

Observamos que tenemos un orden de magnitud de 10^7 para la velocidad. Si sustituimos en la fórmula 5.3, comprobamos que efectivamente ϵ es mucho menor que uno.

$$\epsilon = \frac{U}{\omega e} \sim 10^{-4}$$

(Fórmula 12.2)

12.2. Configuración B

Continuamos con la segunda configuración en la que la parte líquida circula entre dos paredes sólidas pero flexibles. El canal central líquido mantiene el espesor de 4mm y los anchos de las partes sólidas son 5mm cada uno, la altura es 60mm también. El módulo de Young es el mismo que el de la configuración A, $E=3\text{MPa}$.

Se observan para concretos instantes de tiempo como se distribuyen las presiones y velocidades en ParaView (figura 12.13.a., Figura 12.13.b, Figura 12.14.a., Figura 12.14.b).

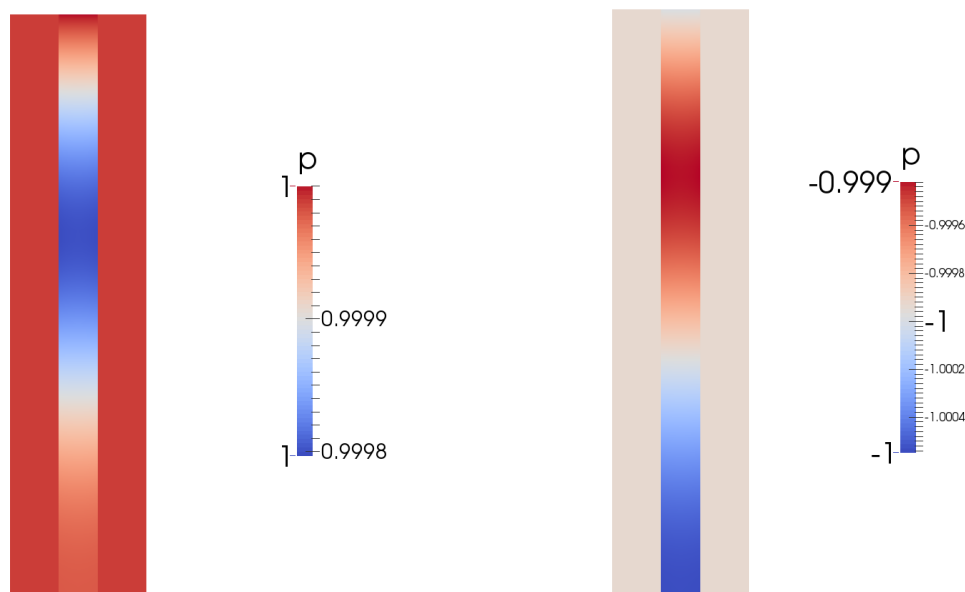


Figura 12.13. a) Distribución de la presión en el instante de tiempo $t=4\text{s}$ (izquierda) b) Distribución de la presión para el instante de tiempo $t=4.5\text{s}$ (derecha)

Al igual que en la configuración anterior se mantiene aproximadamente la misma presión a lo largo de la longitud de la médula. Esto es distinto entre estos instantes de tiempo. Para segundos y cuarto; y segundo y tres cuartos (figura 12.2.a. y b.) se alcanzan la mayor variación de presión en la longitud.

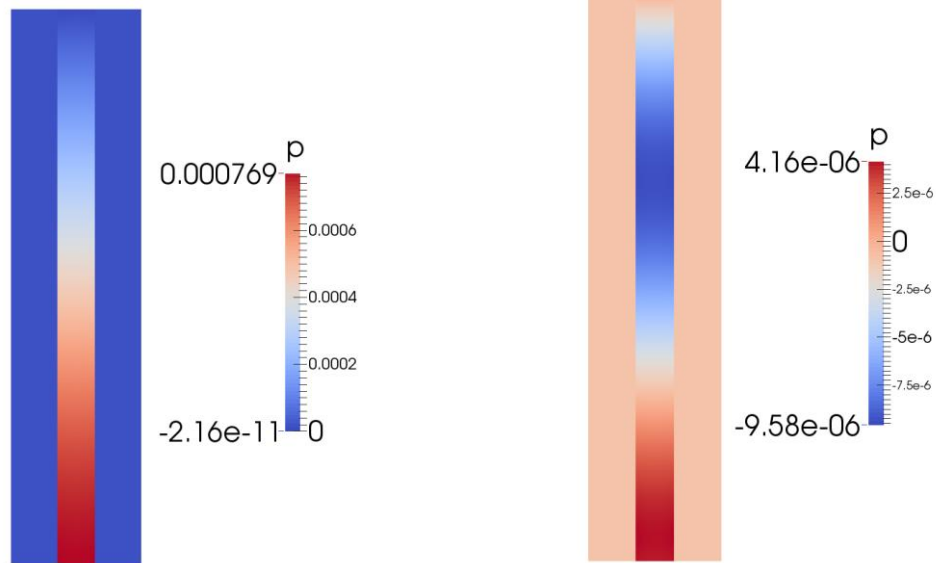


Figura 12.14.a) Distribución de la presión en el instante de tiempo $t=4.25s$ b) Distribución de la presión para el instante de tiempo $t=4.75s$

Ahora, representamos estas variaciones de presión a lo largo de la configuración en una línea.

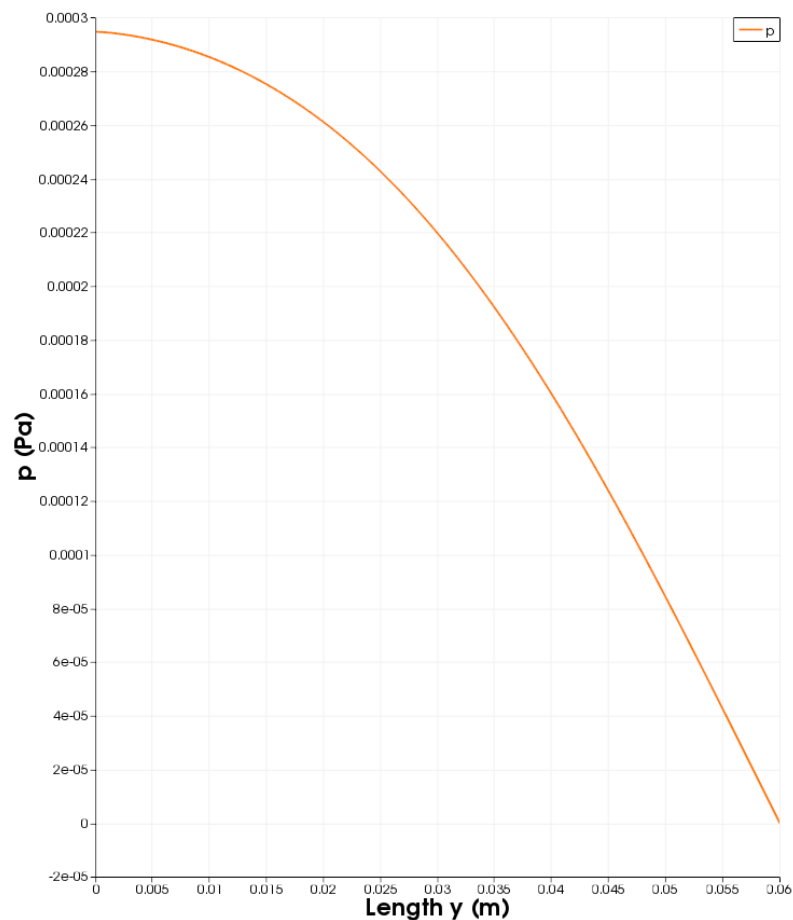


Figura 12.15.a. Variación de presión a lo largo de una línea vertical en $t=3.25s$

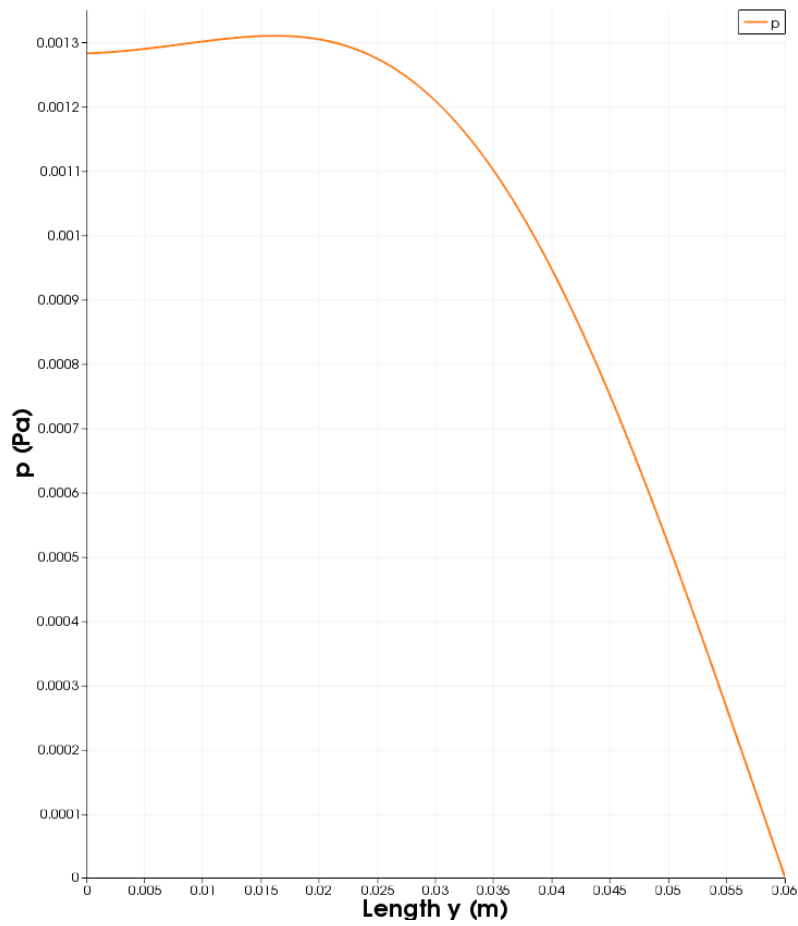
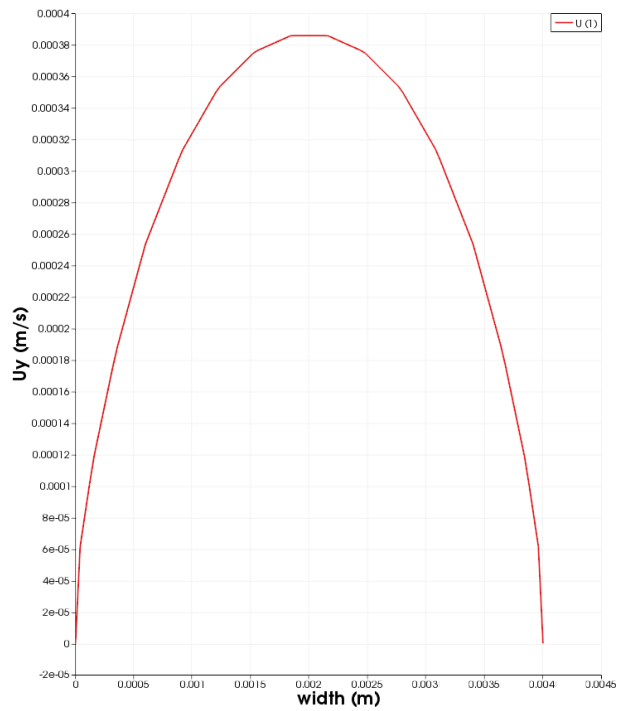


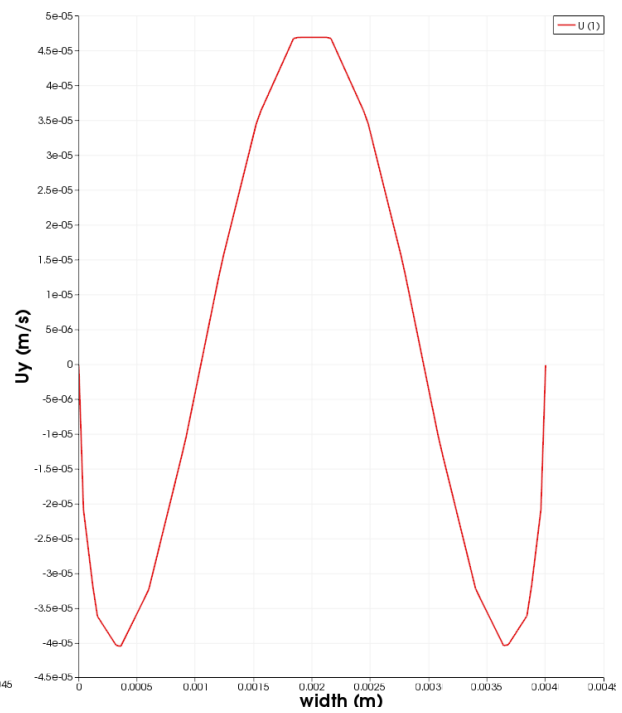
Figura 12.15.a. Variación de presión a lo largo de una línea vertical en $t=3.75s$

Salvo por la perturbación de salida que se produce por la condición de contorno la variación de presión tiende a ser lineal.

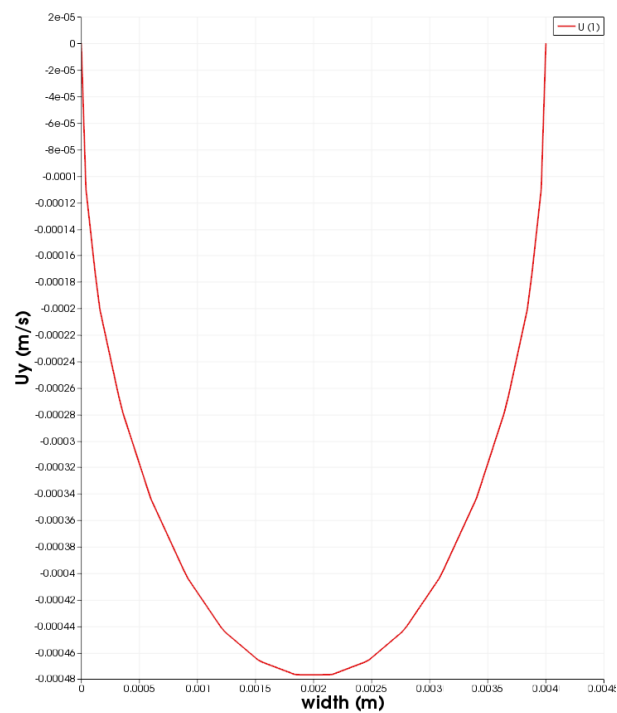
Veamos ahora los perfiles de velocidad que se forman a la entrada y a la salida del canal.



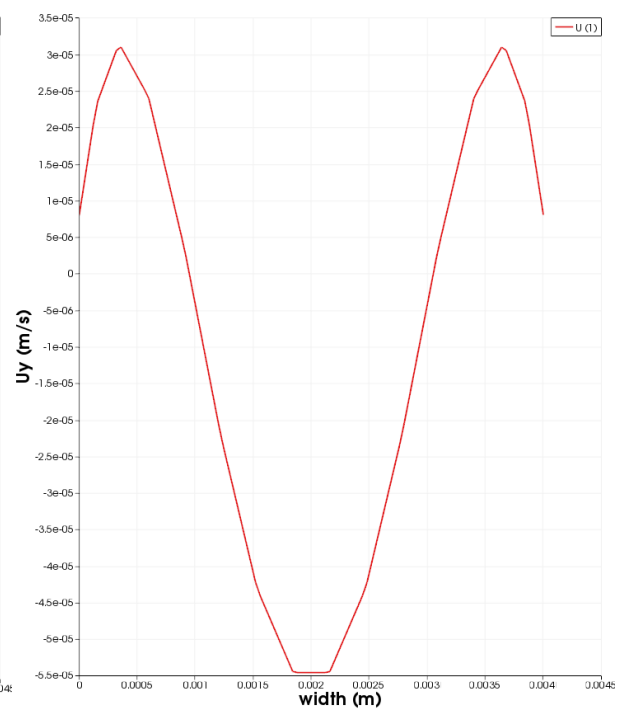
a) $t=3.25$ s



b) $t=3.50$ s



c) $t=3.75$ s



d) $t=4.00$ s

Figura 12.16. Perfil de la velocidad de entrada en distintos instantes de tiempo

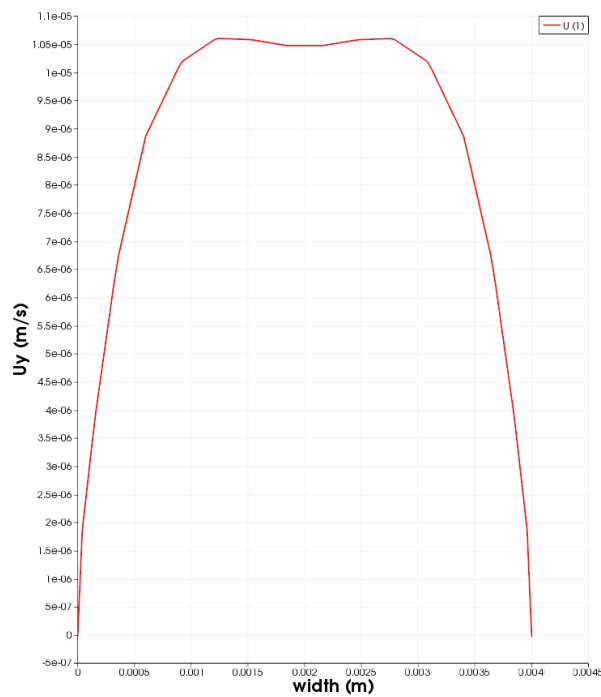
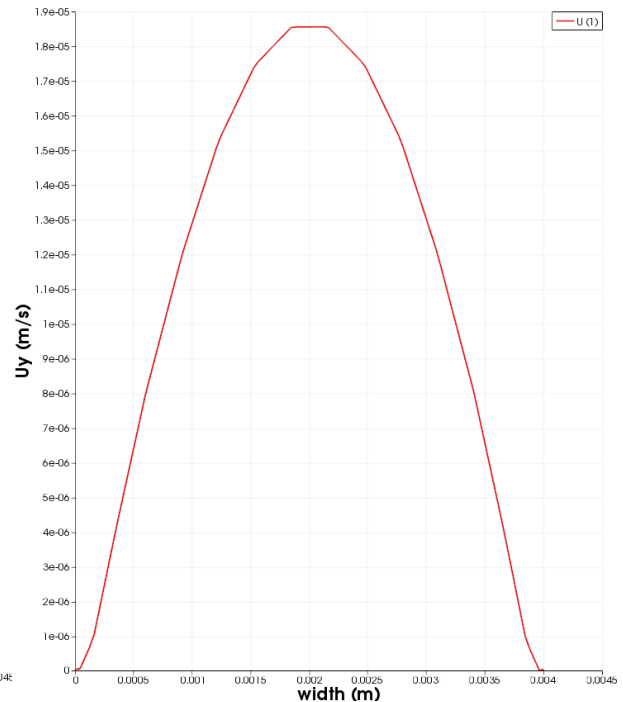
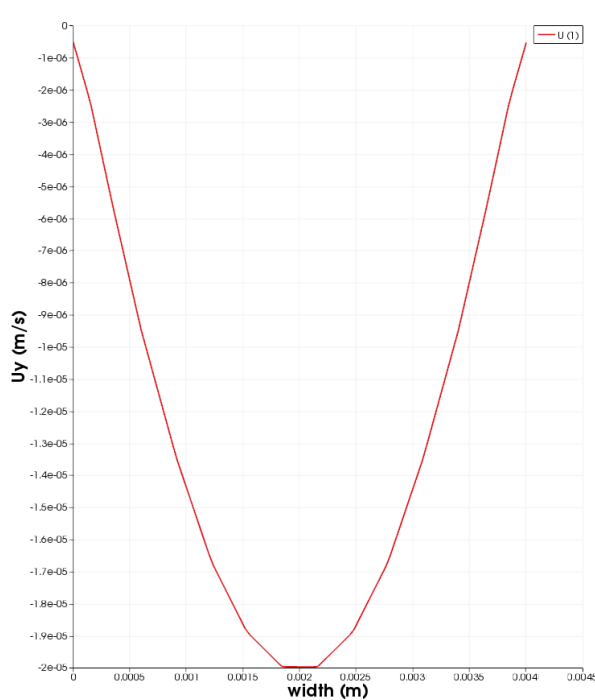
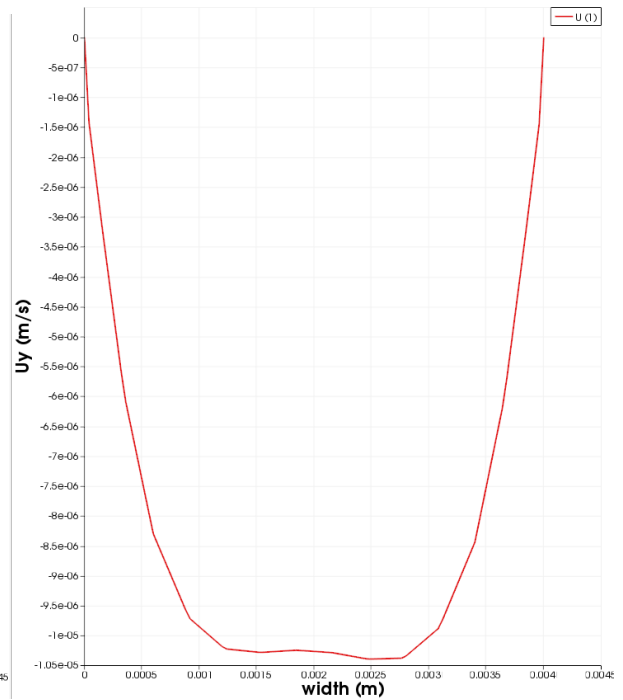
a) $t=3.25s$ b) $t=3.50s$ c) $t=4.00s$ d) $t=4.75s$

Figura 12.17. Perfil de la velocidad de salida en distintos instantes de tiempo

No observamos diferencias importantes con la configuración anterior en cuanto a forma. Pero hay que destacar la simetría en estos resultados y que los valores son menores que en la configuración A, ya que al tener dos paredes flexibles el ancho del

canal podrá ser mayor y, por lo tanto, la velocidad menor. Tomemos como ejemplo la velocidad en 3.25s en la figura 12.18.

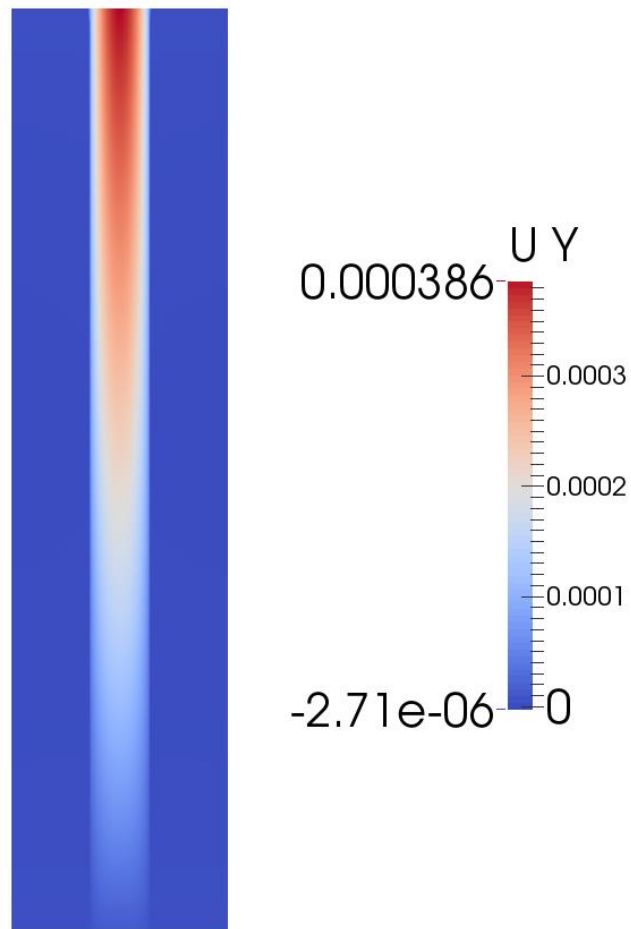


Figura 12.18. Distribución de la velocidad en el instante de tiempo $t=3.25s$

En este instante de tiempo se alcanza la velocidad máxima que, si comparamos con la configuración A, vemos que la velocidad en esta configuración es aproximadamente un cuarto de la velocidad en la configuración A.

12.2.1. Solución promedio – ‘Steady streaming’

Al igual que en la configuración anterior estudiamos algunos puntos de la configuración. Dichos puntos se observan en la figura 12.20.

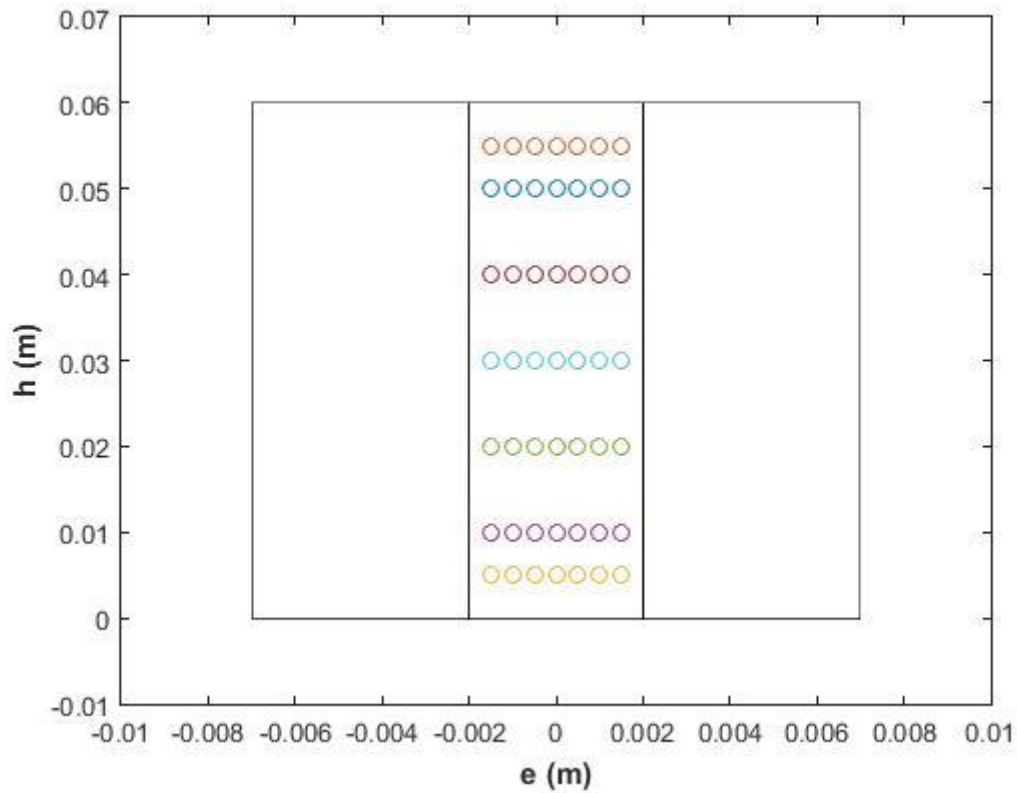


Figura 12.20. Esquema de distribución de los puntos ('probes') de la configuración 'B'

Para la altura adimensional $y^+ = 2/3$, tenemos el siguiente perfil de velocidades medio en la figura 12.21.

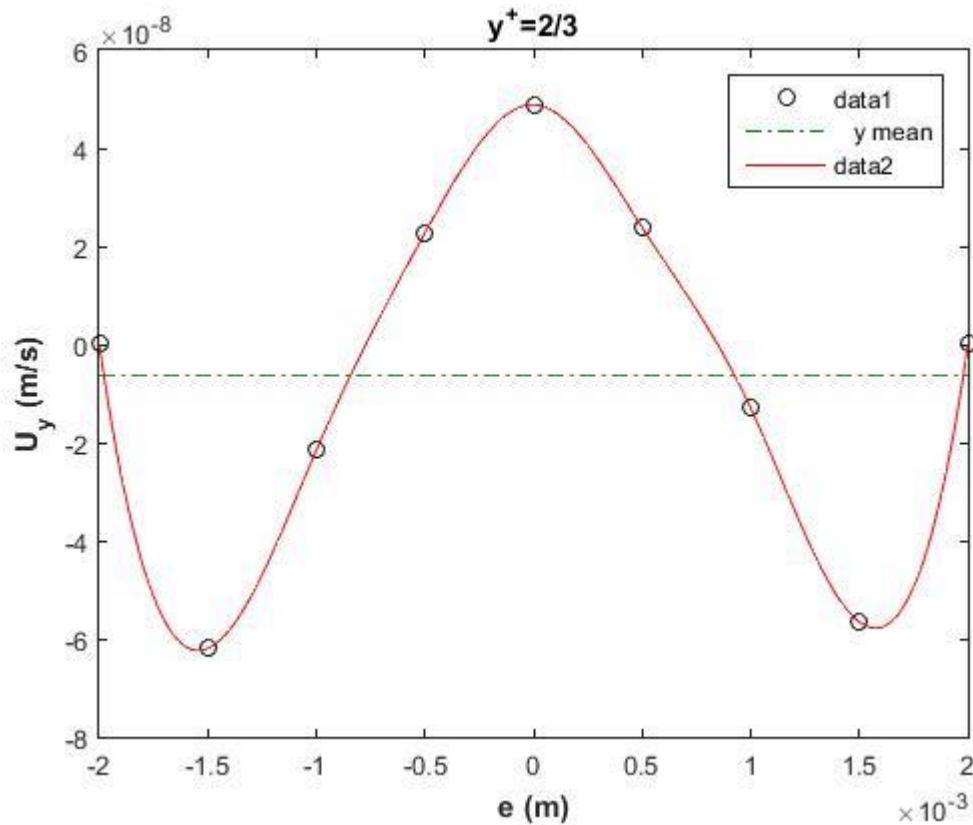


Figura 9.21. Perfil de velocidad media de los probes en $y^+ = \frac{2}{3}$

La línea verde que se muestra es la velocidad media del perfil que como vemos no es cero. Por lo tanto, obtenemos para esta configuración un flujo fijo descendente de $6.383 \cdot 10^{-9} \text{ m/s}$ en valor absoluto.

12.3. Configuración C

Esta última configuración consiste en una parte líquida con una onda de presión de entrada con una amplitud de valor igual a la unidad y flujo de salida, rodeada de un sólido con forma de 'u' como se aprecia en las siguientes imágenes. Las figuras 12.22. y 12.23. muestra la distribución de presiones para diferentes instantes de tiempo. El espesor de este canal fluido también será de 4mm y ancho del sólido 5mm. Como especificación adicional de esta configuración tenemos un módulo de Young de $E=3 \cdot 10^4$ Pa.

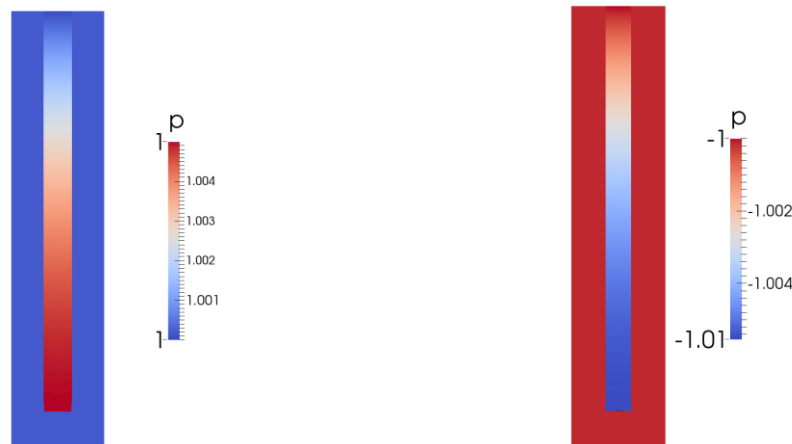


Figura 12.22.a) Distribución de la presión en el instante de tiempo $t=7s$ b) Distribución de la presión para el instante de tiempo $t=7.5s$

Observando estas imágenes podemos extraer como se distribuye la presión a lo largo del tiempo. Como hemos introducido una onda de presión que siguen una función coseno, tenemos la amplitud total de la onda en los segundos enteros (valor positivo de la amplitud); y segundos y medio en valor negativo (figura 12.22.).

En estos instantes de tiempo la presión se mantiene a lo largo del eje 'y' sin apenas variaciones, desde la entrada del fluido hasta el fondo.

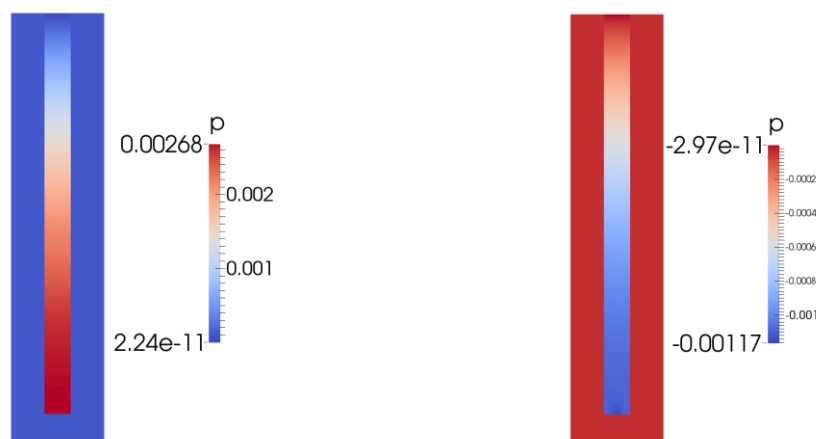


Figura 9.23.a) Distribución de la presión en el instante de tiempo $t=7.25s$ b) Distribución de la presión para el instante de tiempo $t=7.75s$

En el intervalo entre los segundos enteros y medios segundos, ocurre una situación diferente. En los instantes de tiempo que se encuentran en el medio de estos intervalos, es decir, $t=n.25$ y $t=n.75$ siendo 'n' un número entero, se da el máximo salto de presiones a lo largo de la configuración. Como veremos posteriormente, este salto coincide con el máximo y mínimo de la velocidad. Las ondas de presión y velocidad se solapan de forma que cuando la presión es máxima la velocidad es cero y viceversa.

A continuación, mostramos esta variación de presión en las figuras 12.24.a. y 12.24.b para los instantes de tiempo de las imágenes anteriores. Esto se hace tomando una línea vertical y representando los valores de presión a lo largo de esta. Obtenemos como resultado unas gráficas de presión frente a la longitud de la línea descrita.

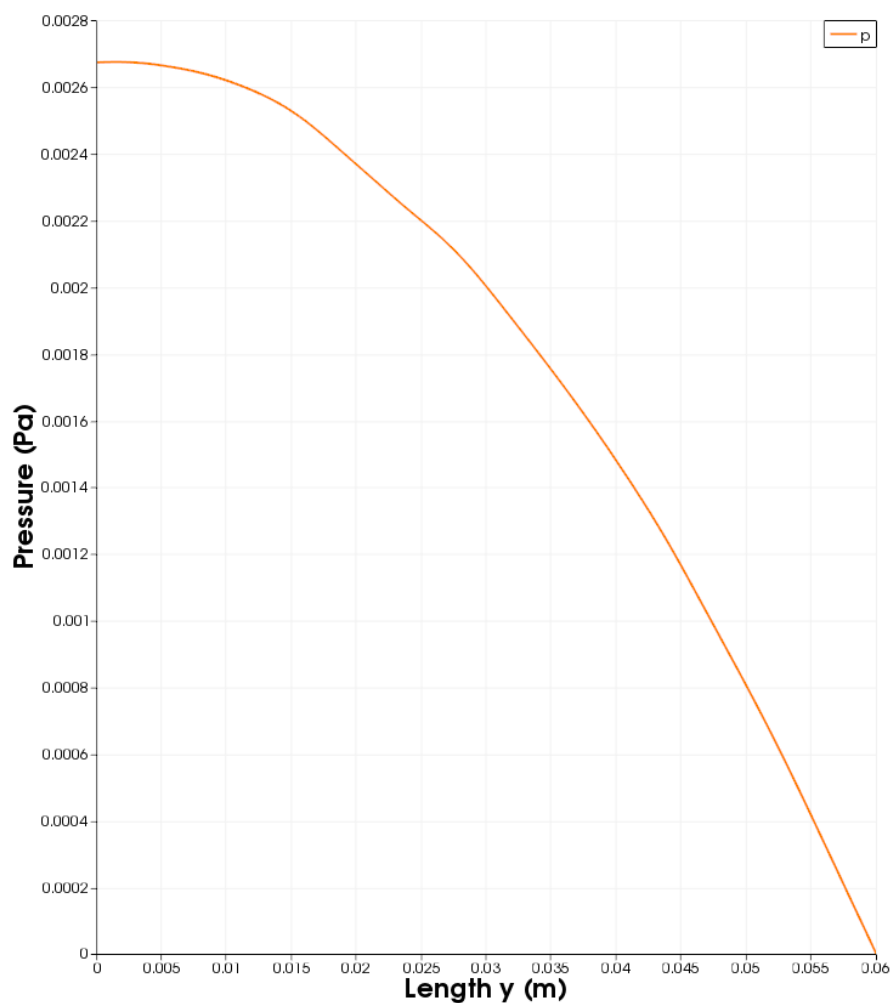


Figura 12.24.a. Variación de presión a lo largo de una línea vertical en $t=7.25s$

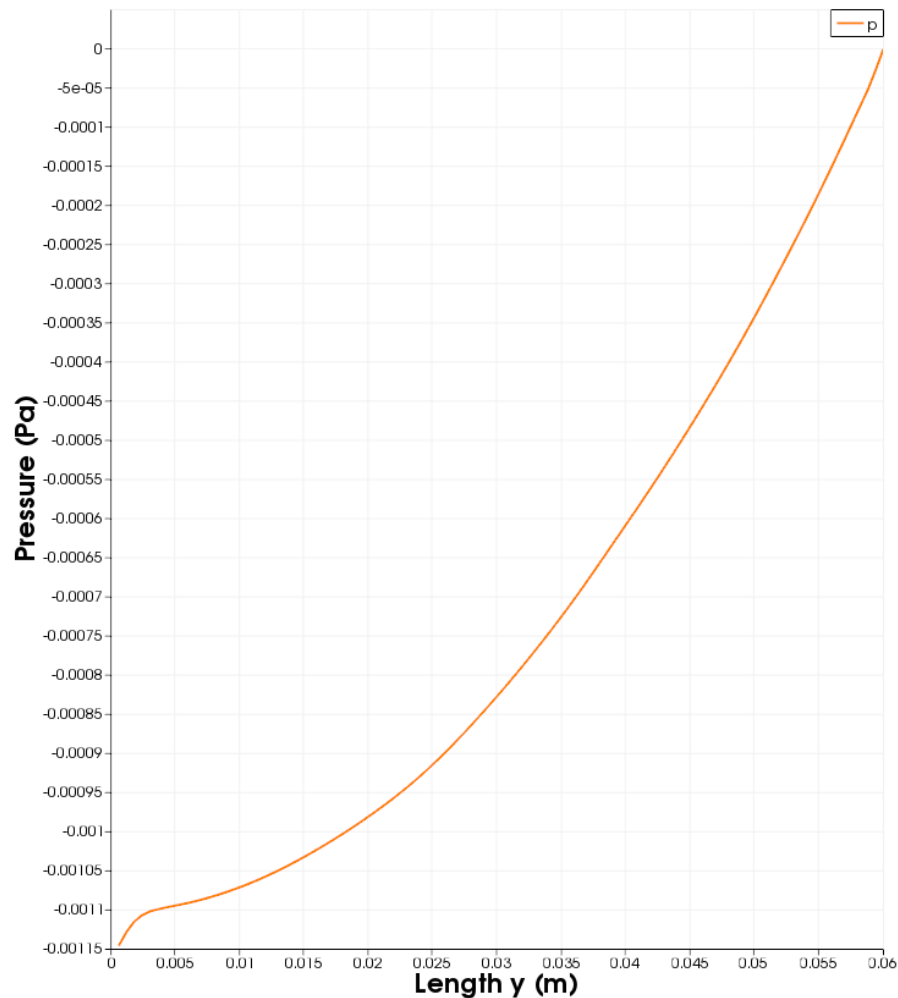


Figura 12.24.b. Variación de presión a lo largo de una línea vertical en $t=7.75s$

Lo más destacable que observamos es que se sigue una función parabólica al contrario que con la tesis doctoral de Vegard, en la cual se representa una línea recta. Esto es porque esta configuración en cerrada rodeada de sólido.

A continuación, estudiaremos la velocidad en la dirección del eje 'y', seguiremos el mismo procedimiento de estudio que para la presión. Se muestra la distribución de velocidades en las figuras 12.25 y 12.26.

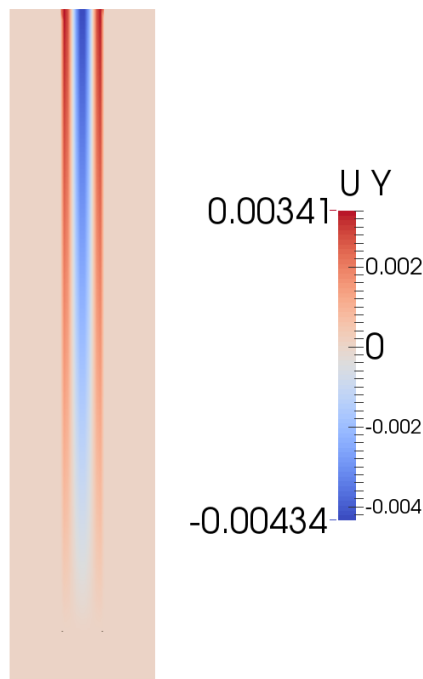


Figura 12.25.a. Distribución de la presión en el instante de tiempo $t=7s$

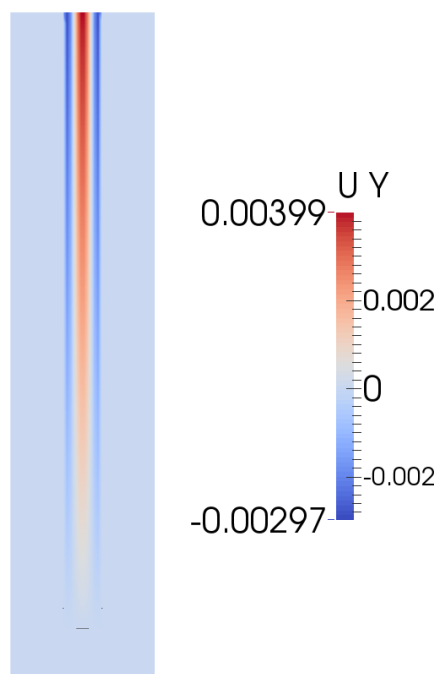


Figura 12.25.b. Distribución de la velocidad en el instante de tiempo $t=7.5s$

La presión en los segundos enteros es máxima en el 'inlet' por lo que el flujo en el centro del líquido tiene un sentido descendente mientras por el exterior asciende.

Para la figura 12.25.b tenemos el caso contrario: flujo hacia el 'inlet' en la parte más central del líquido y descendente en el exterior.

Para los instantes donde la velocidad en el eje 'y' son máximas y mínimas obtenemos las configuraciones de la figura 12.26.

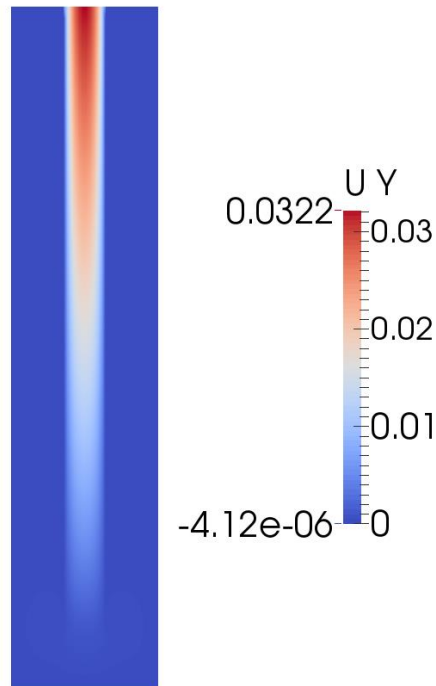


Figura 12.26.a. Distribución de la presión en el instante de tiempo $t=7.25s$

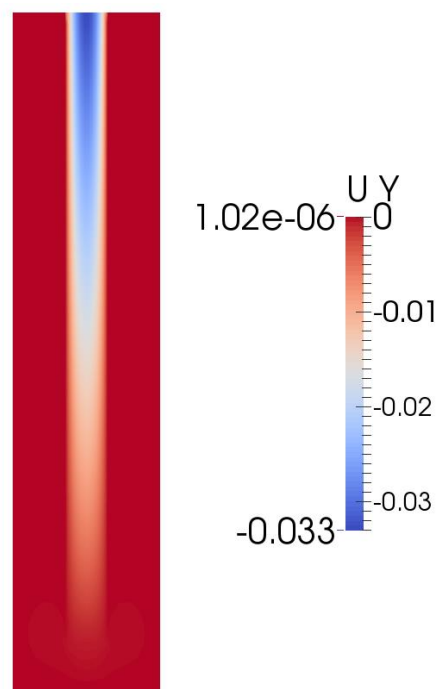


Figura 12.26.a. Distribución de la presión en el instante de tiempo $t=7.75s$

En estos casos no se da flujo en doble sentido. Se obtiene únicamente un perfil de velocidad convencional de un tubo en el sentido que marca el signo de la velocidad en función de si se encuentra en un máximo o un mínimo de la onda que se describe.

La velocidad a lo largo del eje 'y' sigue una función lineal para todos los instantes de tiempo.

Por otro lado, representamos para cuatro instantes de tiempo los perfiles de velocidad que aparecen en la entrada del fluido (figura 12.27).

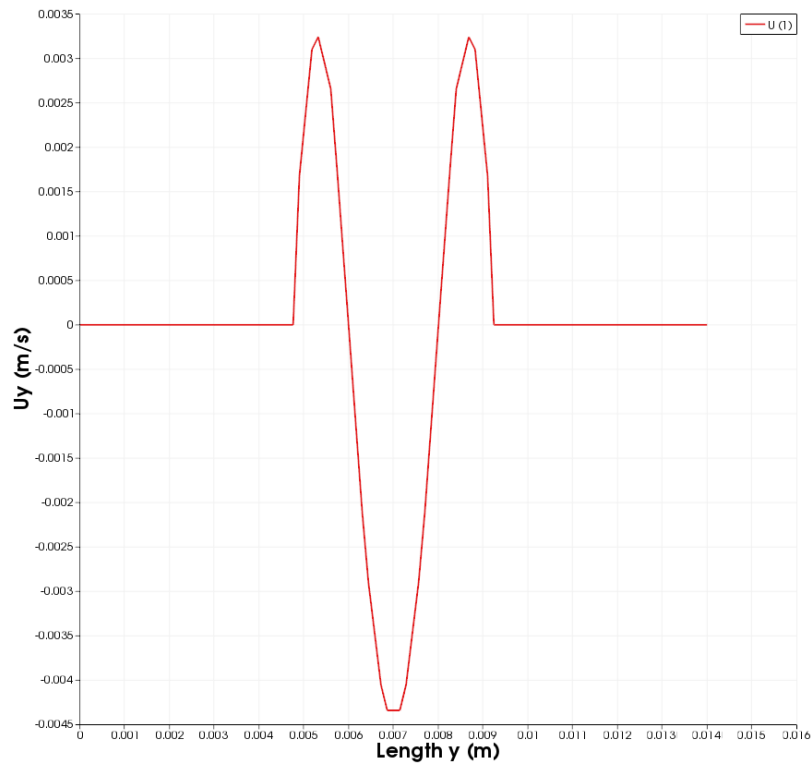


Figura 12.27.a. Perfil de velocidad a la entrada $t=7.00$ s

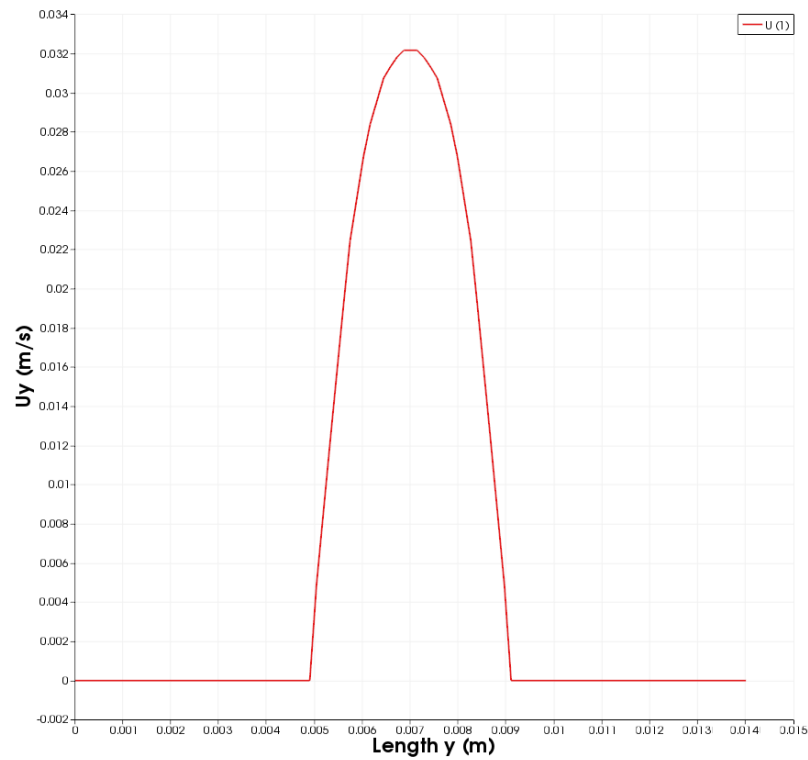
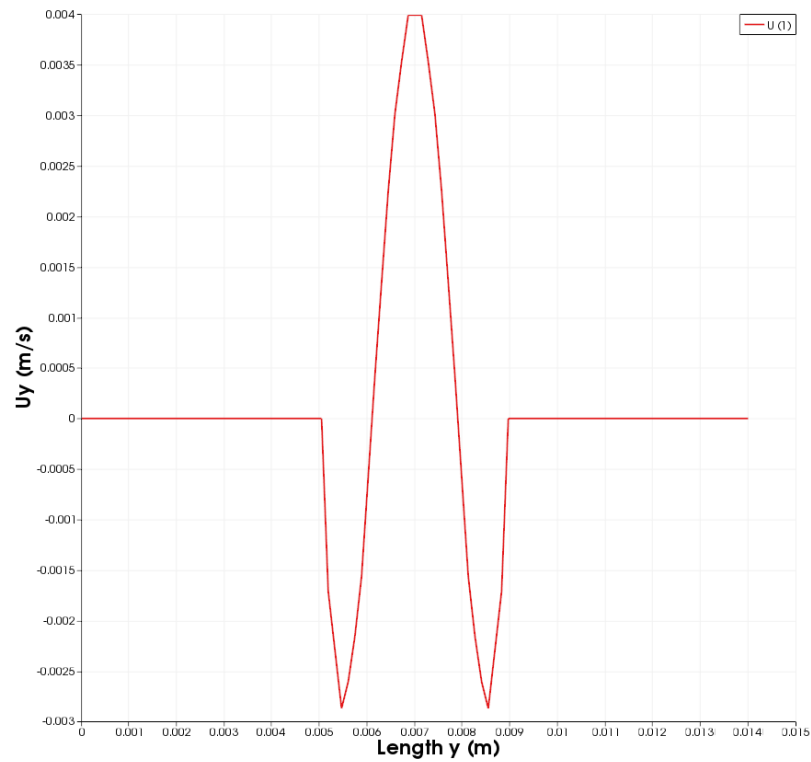
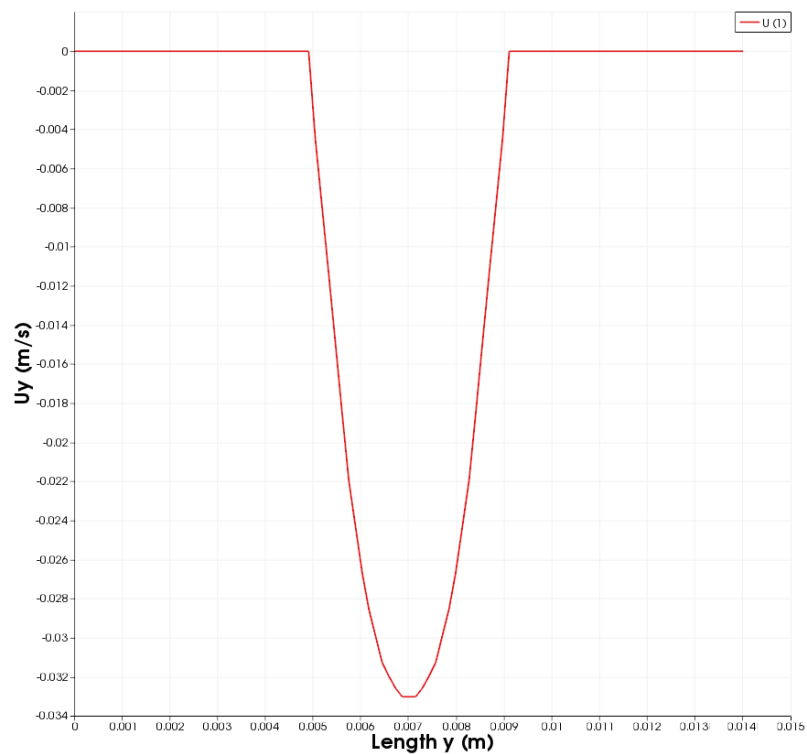


Figura 12.27.b. Perfil de velocidad a la entrada $t=7.50$ s

Para estos instantes de tiempo se representa el flujo en doble sentido que habíamos predicho con las figura 12.25.a. y b.

**Figura 12.27.c. Perfil de velocidad a la entrada $t=7.25$ s****Figura 12.27.d. Perfil de velocidad a la entrada $t=7.75$ s**

En estos instantes de tiempo, tenemos un perfil de velocidad convencional en un solo sentido.

12.3.1. Solución promedio – ‘Steady streaming’

Tenemos para esta configuración la distribución de los ‘probes’ que se observan en la figura 12.28.

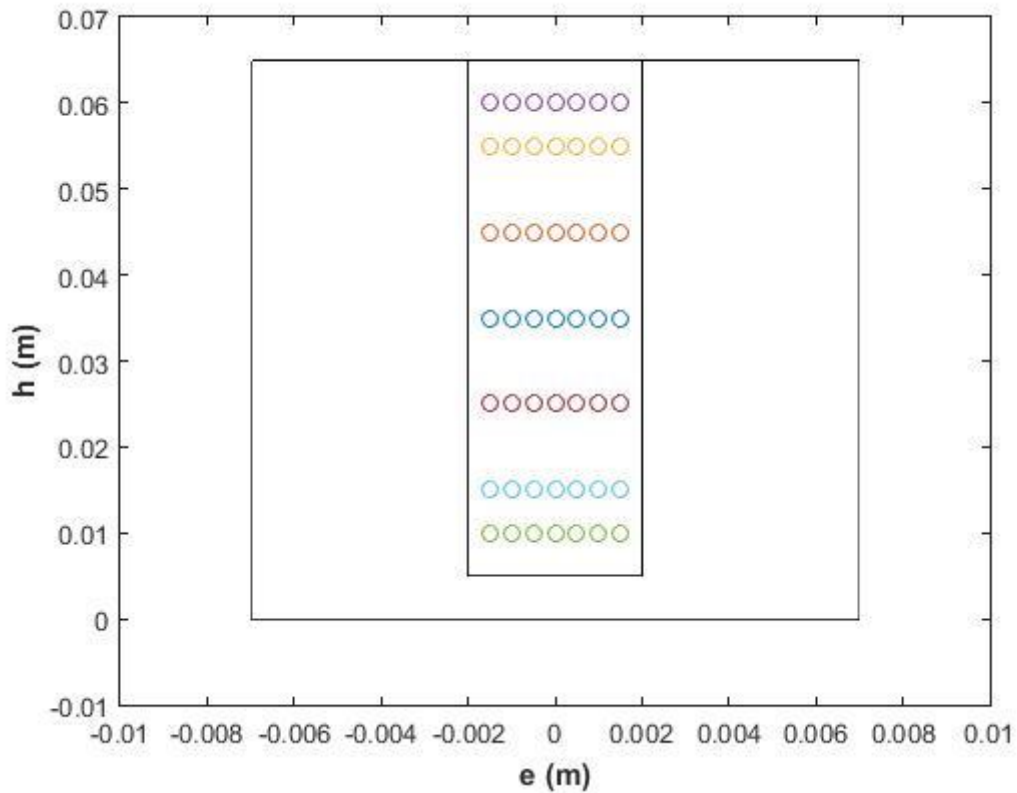


Figura 12.28. Esquema de distribución de los puntos (‘probes’) de la configuración ‘C’

Como en los casos anteriores, mostramos primero la onda de presión que se forma en un ‘probe’ intermedio con el paso de tiempo (figura 12.29).

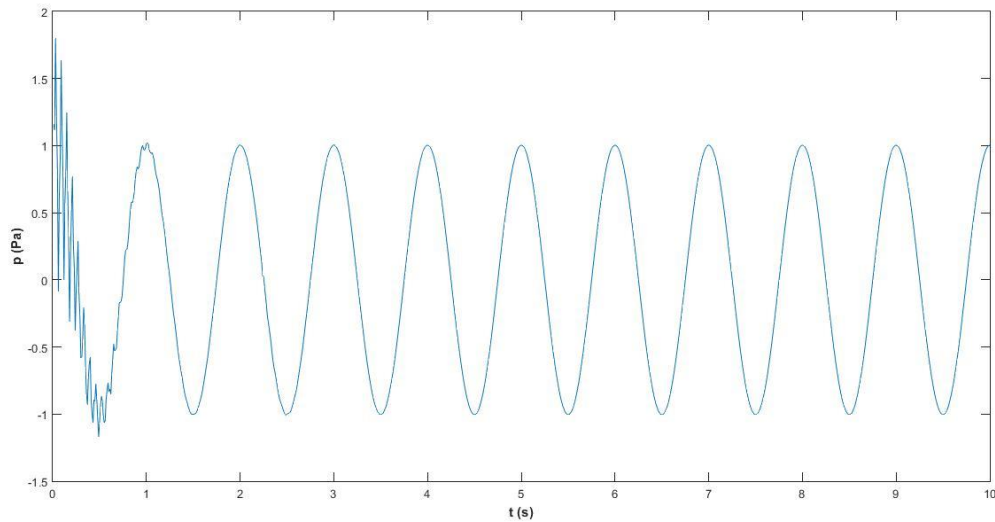


Figura 12.29. Onda de presión del 'probe' central

Volvemos a tener aquí interferencias en los primeros segundos hasta que se consigue la convergencia y estabilidad de la solución. Trabajaremos entonces, con el intervalo de tiempo limpio que tenemos de seis a diez segundos. Así, obtenemos una onda de velocidad representada en la figura 12.30.

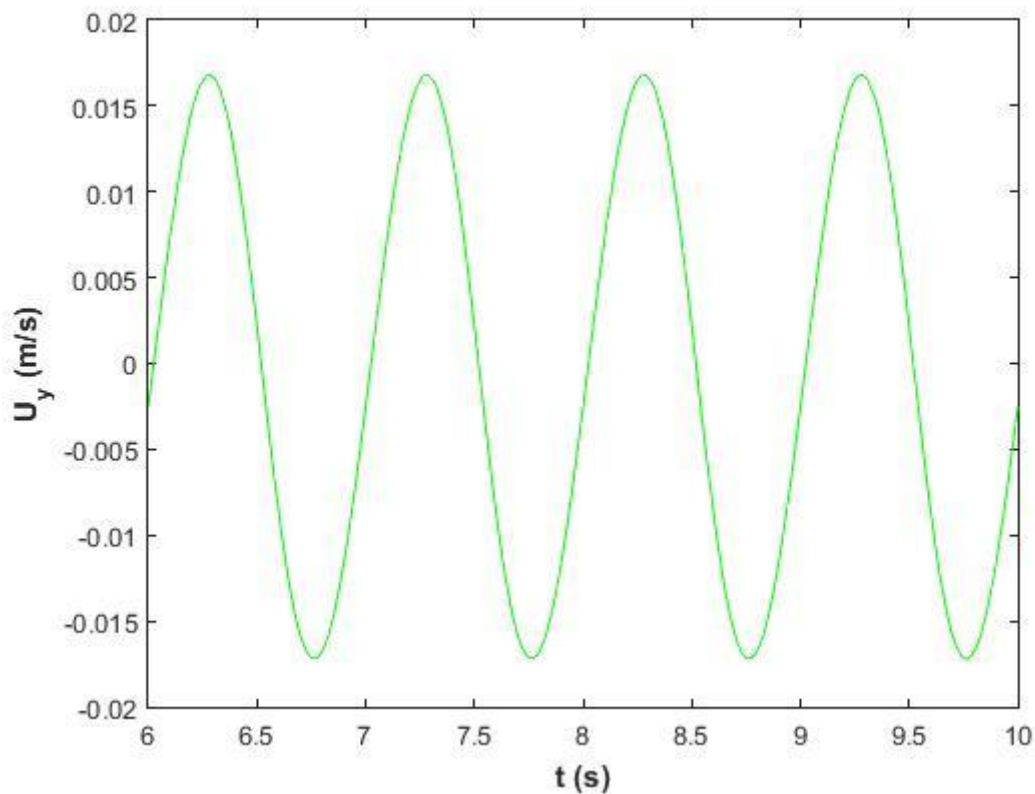


Figura 12.30. Onda de velocidad del 'probe' central

En este caso, al obtener una señal libre de interferencias, podemos calcular el valor de la velocidad media en el eje 'y' de un punto usando la función de MatLab 'mean' directamente.

Para la obtención de las velocidades medias nos ayudamos de un programa creado para este motivo, el cuál lee la matriz de velocidades en los instantes de tiempo elegidos. En resultado del programa es una matriz con dichas velocidades que a continuación representamos.

Estas velocidades medias a lo largo del tiempo son representadas frente al espesor a tres alturas distintas para obtener tres perfiles de velocidades. Interpolamos los puntos de estas gráficas con MatLab para obtener una curva del perfil de velocidad en el eje 'y' medio y observar la forma aproximada del perfil de velocidades completo (figura 12.31.a., b. y c.).

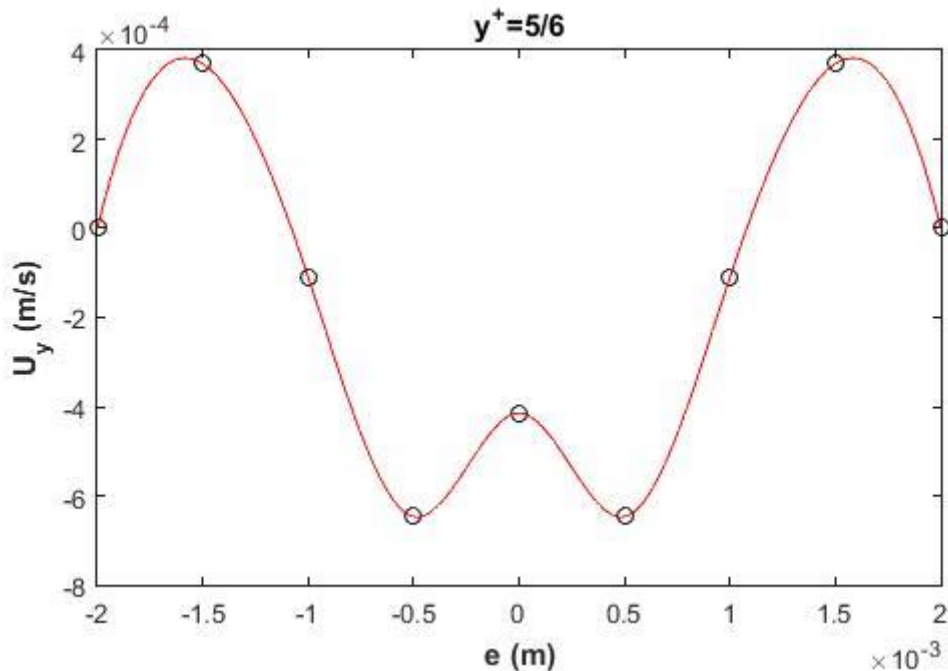


Figura 12.31.a. Perfil de velocidad media de los probes en $y^+ = \frac{5}{6}$

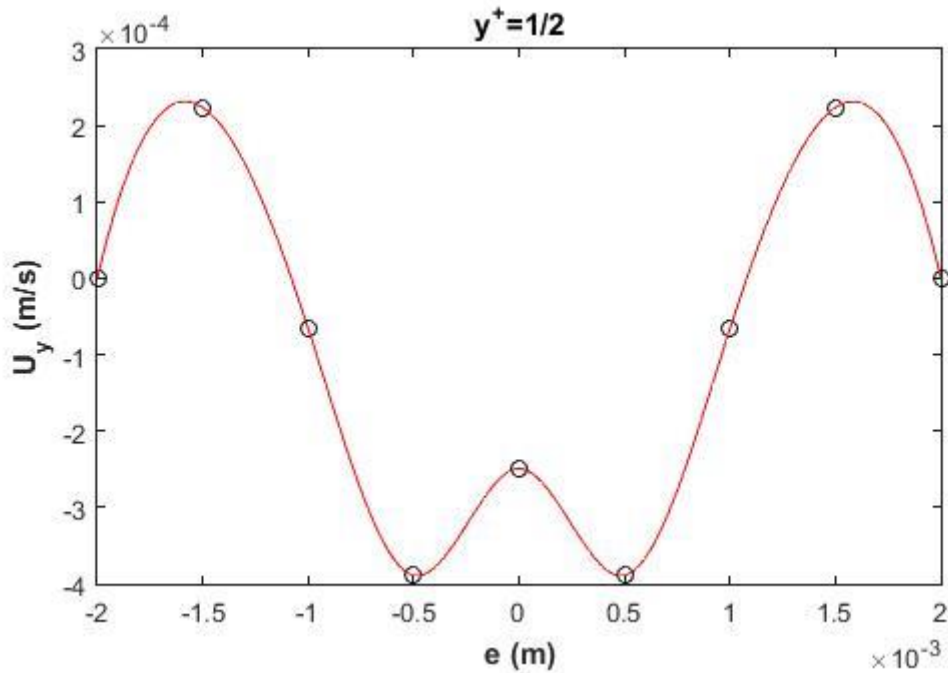


Figura 12.31.b. Perfil de velocidad media de los probes en $y^+ = \frac{1}{2}$

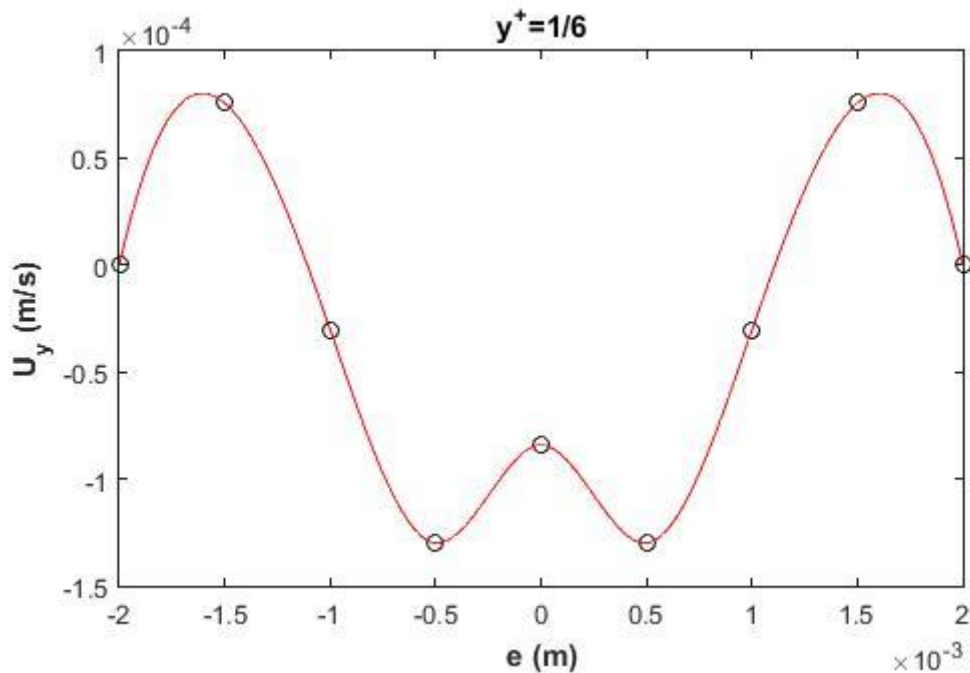


Figura 12.31.c. Perfil de velocidad media de los probes en $y^+ = \frac{1}{6}$

Estas y^+ son tomando como referencia $y = 0$ en la frontera entre el líquido y el sólido en la parte baja.

Dado que solo hay una entrada/salida, el flujo de entrada tiene que ser igual al flujo de salida, y la integral de la velocidad media a lo largo del espesor debe de ser cero. No obstante, al tener una interfase móvil, nuestro volumen de control va a variar

y, por lo tanto, el espesor variará por lo que al representar un espesor fijo y estableciendo una velocidad nula en los extremos cometemos un error a tener en cuenta.

	$U_y \cdot 10^4 \text{ (m/s)}$						
	Probe 1	Probe 2	Probe 3	Probe 4	Probe 5	Probe 6	Probe 7
$y^+ = \frac{5}{6}$	3,6930	-1,0939	-6,4686	-4,1436	-6,4682	-1,0935	3,6931
$y^+ = \frac{1}{2}$	2,2346	-0,6575	-3,8890	-2,4911	-3,8888	-0,6574	2,2346
$y^+ = \frac{1}{6}$	0,7569	-0,3025	-1,2992	-0,8421	-1,2992	-0,3025	0,7569

Tabla 12.2. Velocidades medias en el eje 'y' de los probes en la configuración 'C'

Por último, estudiamos los desplazamientos que se producen, el valor máximo, como se aprecia en la figura 12.32, es de $0.128mm$.

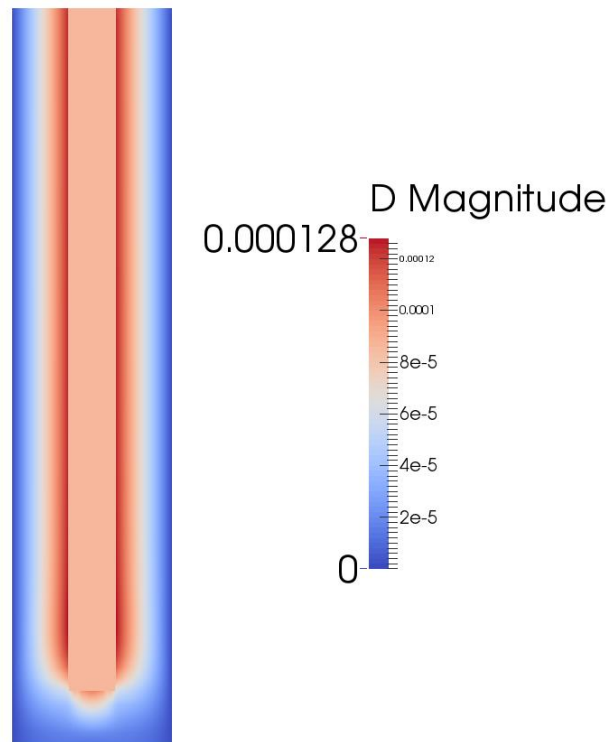


Figura 12.32. Valores absolutos de desplazamientos en $t=7.00s$

13. Estudio paramétrico

Con el estudio de las configuraciones prestando atención a los parámetros establecidos en diversas simulaciones hemos extraído que el módulo de Young es uno de los parámetros que más influyen en el flujo que estudiamos.

Por lo tanto, el estudio paramétrico que se presenta se centrará en como varían los resultados obtenidos con distintos módulos de Young para la configuración cerrada 'C'. Compararemos la configuración 'C' del apartado anterior con $E = 2 \cdot 10^4 Pa$, $E = 3 \cdot 10^4 Pa$ y $E = 6 \cdot 10^4 Pa$ ante una misma onda de presión con amplitud igual a la unidad.

Comenzamos representando las ondas de velocidad que se forman en un intervalo de tiempo en el punto central de la configuración en la figura 13.1.

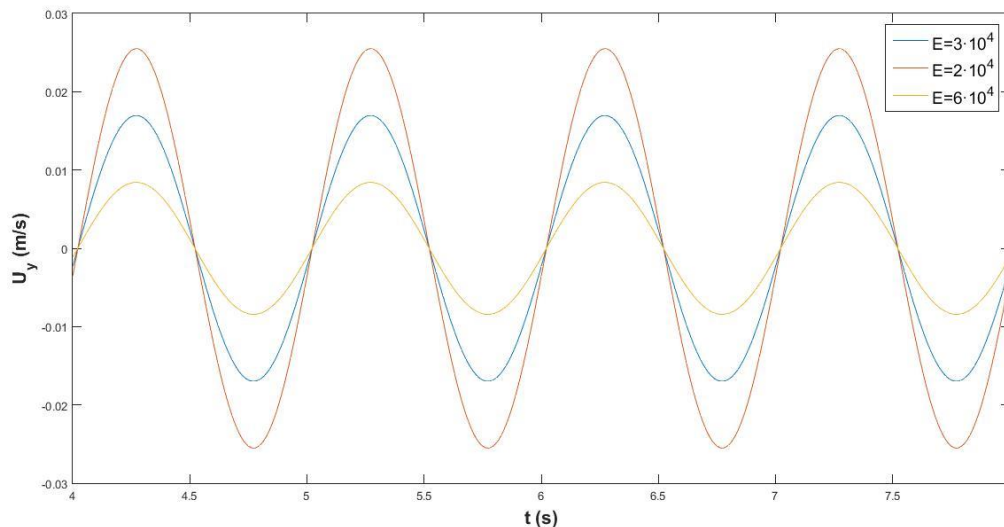


Figura 13.1. Variación de la velocidad respecto al tiempo para tres módulos de Young

Observamos que las ondas están en fase y que a mayor módulo de Young menor es la amplitud de la onda. Las velocidades máximas que se alcanzan son del orden de centímetros por segundo.

A continuación, en la figura 13.2, se muestran los perfiles de velocidad medios que se obtienen a distintas alturas para los tres módulos de Young.

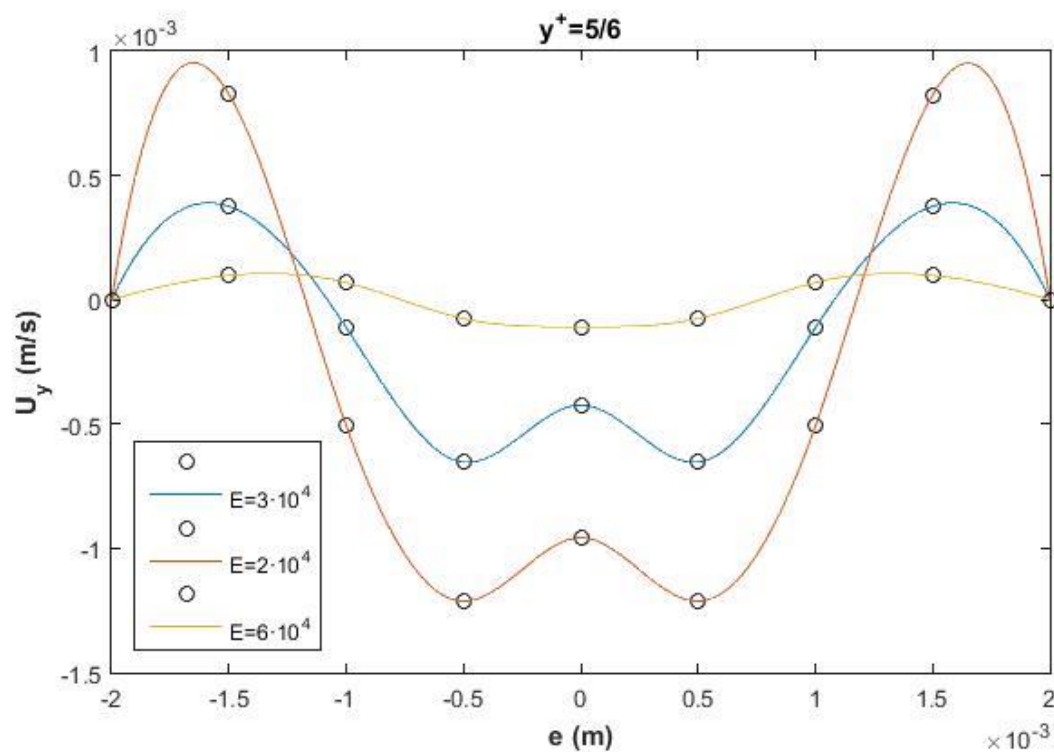


Figura 13.2.a. Perfil de velocidad media de los probes en $y^+ = \frac{5}{6}$

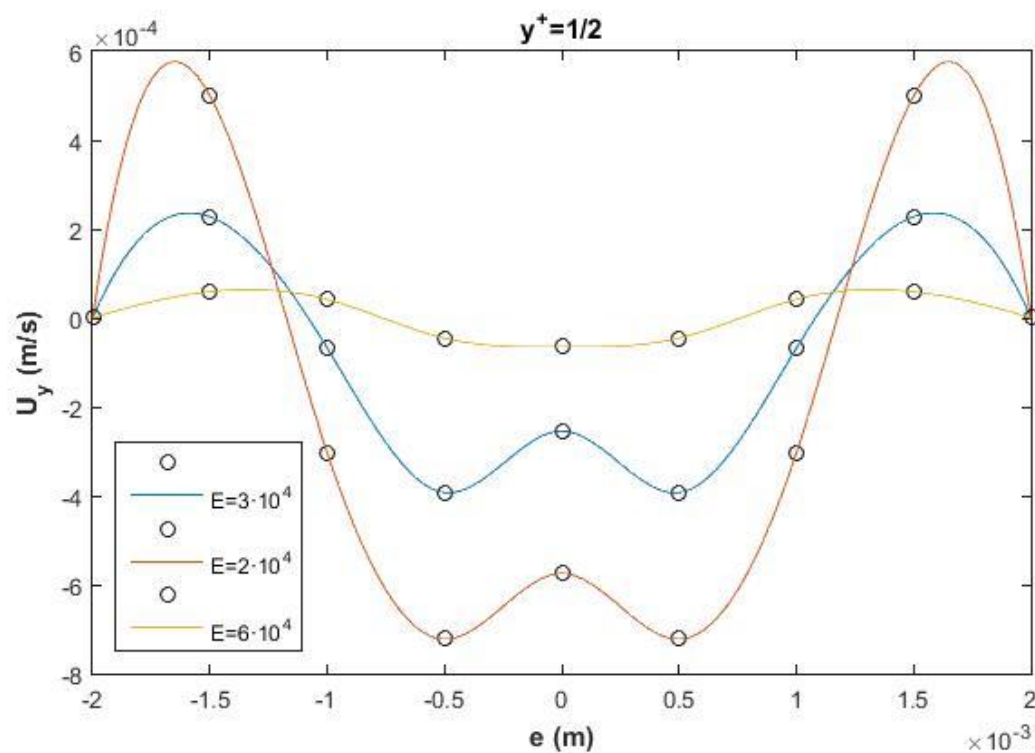


Figura 13.2.b. Perfil de velocidad media de los probes en $y^+ = \frac{1}{2}$

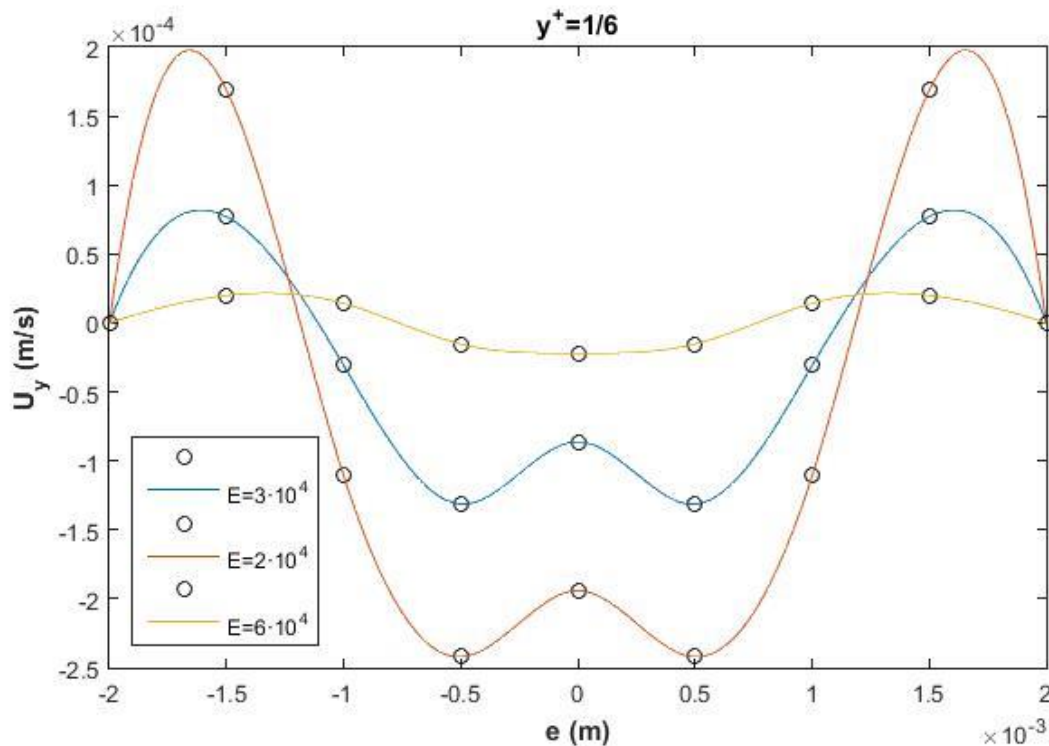


Figura 13.2.c. Perfil de velocidad media de los probes en $y^+ = \frac{1}{6}$

Como vimos en el apartado anterior, existe un flujo neto ascendente por el exterior y descendente por el interior. Estos flujos netos tienden a ser menores cuanto mayor es el módulo de Young. Los máximos y mínimos de los perfiles se dan en la parte superior de la configuración ($y^+ = 5/6$) como se podría prever.

Una vez que hemos estudiado como varía la velocidad y sabiendo que la onda de presión es la misma, tenemos que estudiar las deformaciones que se producen.

Se ha mencionado anteriormente que los picos de presión (positiva y negativa) se alcanzan en un número de segundos entero; y segundo y medio. Lógicamente, en estos instantes se producirá las mayores deformaciones que serán iguales en valor absoluto. En la figura 13.3, observan las deformaciones en siete segundos para los tres módulos de Young distintos.

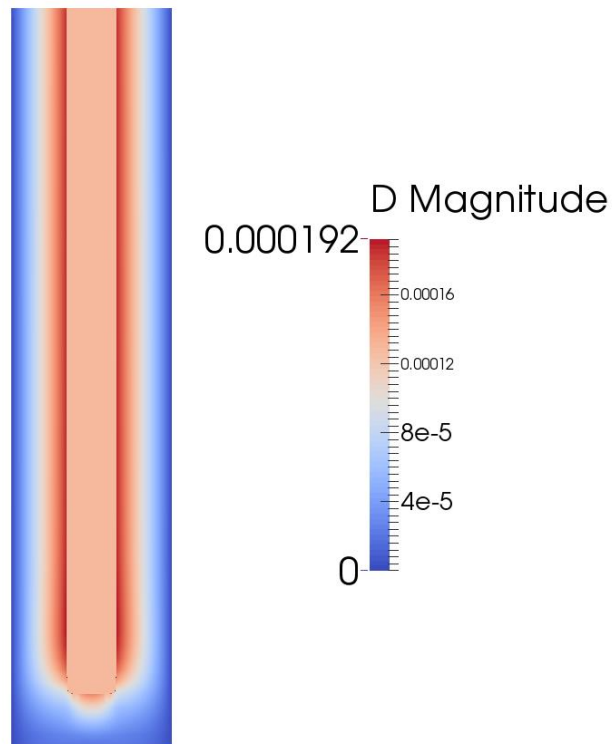


Figura 13.3.a. Valores absolutos de desplazamientos en $t=7.00s$ para $E=2 \cdot 10^4$ Pa

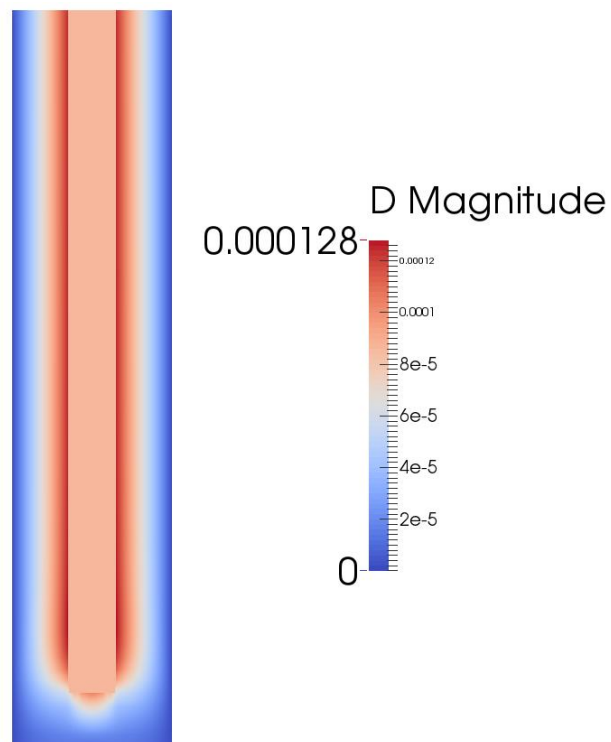


Figura 13.3.b. Valores absolutos de desplazamientos en $t=7.00s$ para $E=3 \cdot 10^4$ Pa

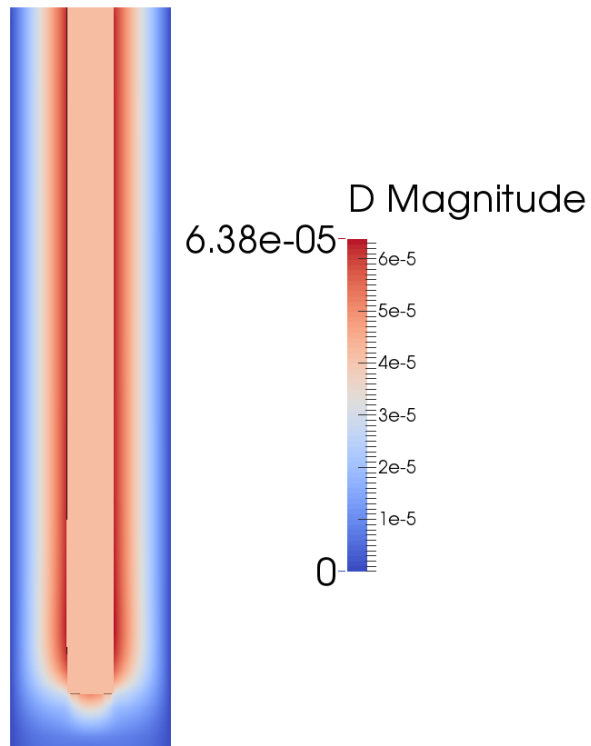


Figura 13.3.c. Valores absolutos de desplazamientos en $t=7.00s$ para $E=6 \cdot 10^4$ Pa

La distribución de las deformaciones a lo largo de la configuración es igual, pero se observan diferentes máximos de desplazamiento de la pared móvil. Las unidades siguen en el sistema internacional (metros).

14. Conclusiones

Para todos los casos, la onda de presión de frecuencia $1/2\pi$ ha generado una onda de velocidad con el mismo periodo, pero desfasada $\pi/2$. ¿POR QUÉ EL DESFASE? Pero la frecuencia de las ondas, en cualquier caso, habría sido impuesta por nosotros. Sería interesante estudiar la frecuencia real, aunque a simple vista puede que no sea relevante para los resultados del problema

Centrémonos ahora en el estudio paramétrico. Debido a la inesperada complejidad de este trabajo el estudio paramétrico solo se ha centrado en el módulo de Young. Sin embargo, existen otros parámetros que influyen ampliamente en los resultados del problema y que deberían de estudiarse. Estos son: el coeficiente de Poisson del sólido, densidad del sólido, presión de entrada, etc.

Una de las dificultades que hemos encontrado era la lentitud de las simulaciones, lo cual se debe a la complejidad de los cálculos y a que la aplicación fsiFoam aún no permite simular en paralelo. Además, el código de este 'solver' nos ha dado en general muchos problemas de funcionamiento.

Como hemos visto anteriormente en este pequeño estudio paramétrico, las velocidades que se alcanzan son mayores cuanto menor es el número de Young. Sabiendo esto, se podría deducir que si se produce esta circulación en un canal totalmente rígido sería menor que en cualquiera de los casos estudiados. Esto significa que la flexibilidad de las paredes influye ampliamente en el movimiento del líquido cerebroespinal a lo largo de la columna.

Esta hipótesis, la hemos querido comprobar tanto para el caso cerrado como para el caso abierto. Nos centraremos en el caso cerrado ya que es donde hemos realizado el estudio paramétrico. El procedimiento que hemos llevado a cabo ha sido realizar una malla mediante el mallador de OpenFOAM con las mismas medidas de la parte fluida de la configuración C, hemos establecido las mismas condiciones de contorno y hemos ejecutado la simulación con ' pisoFoam ' que es el ' solver ' que incorpora ' fsiFoam ' para la parte del fluido.

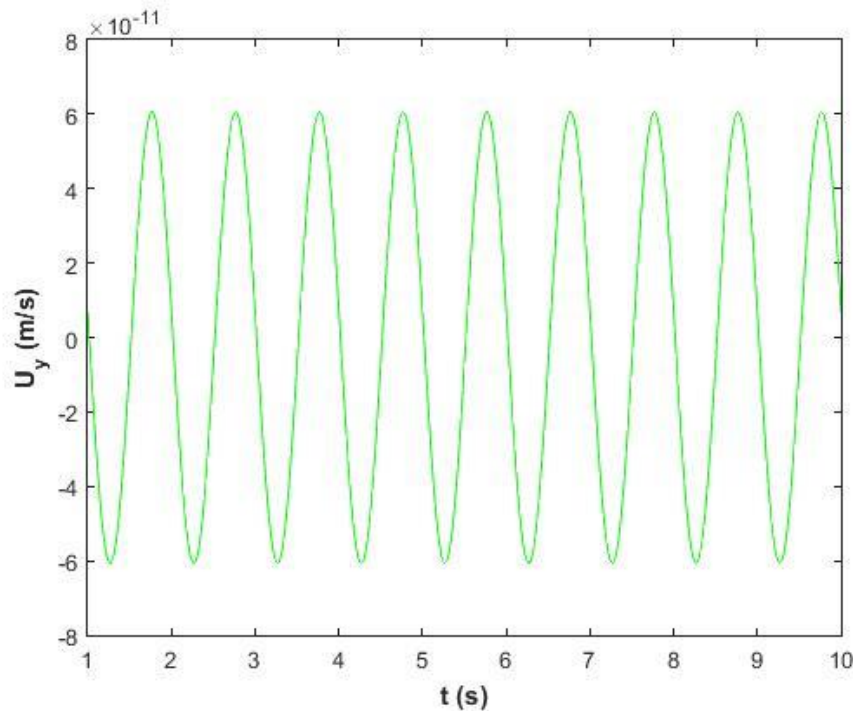


Figura 14.1 Onda de velocidad para el flujo en un conducto rígido cerrado

Si observamos la figura 14.1 podremos confirmar nuestra hipótesis ya que obtenemos una orden de magnitud de 10^{-11} mientras que para la configuración con sólido deformable tendremos un orden de 10^{-2} .

14.1. Estudios futuros

Cuando estudiamos las propiedades de la médula, los estudios hasta ahora no han distinguido entre materia gris y materia blanca. Si existiera una diferencia considerable en cuanto a características, esta consideración debería ser aplicada. Aunque, algunos estudios apuntan a que esta diferencia en cuanto a elasticidad y porosidad es insignificante.

Según la revisión anual de este tema de 2016 [7], una de las direcciones que tenía que seguir el modelo del sistema nervioso central era un modelo FSI con una deformación de malla Euler-Lagrangiana que se aborda en este trabajo de fin de grado.

Por otro lado, se sabe que existe un filtrado del LCR por lo que el sólido debería de caracterizarse usando poroelasticidad. Esto ya ha sido presentado en algunos estudios, pero se ha dejado fuera del objetivo de éste.

Por último, para completar el estudio, sería conveniente hacer un modelo completo del sistema nervioso central y en tres dimensiones con el que obtendríamos resultados directamente extrapolables a los datos 'in vivo'.

Para resumir, el futuro de este trabajo sería obtener un modelo real y completo de la médula espinal mediante técnicas CAD, realizar mediciones 'in vivo' para obtener los datos reales de los parámetros que aparecen en este problema (incluyendo una onda de presión real que sabemos que no es senoidal), tener en cuenta la poroelasticidad de los sólidos deformables y, por último, mejorar el código de programación que implementa OpenFOAM para las funciones que hemos usado o incluso crear uno propio. Esto nos permitirá obtener datos más fiables que podamos validar.

Referencias

- [1] www.wikipedia.org <https://es.wikipedia.org/wiki/Meninges#/media/File:Meninges-es.svg>
- [2] <https://www.slideshare.net/chucky01/spinal-cord-and-spinal-nerves-lab>
- [3] <https://es.wikipedia.org/wiki/Meninges#/media/File:Gray770-es.svg>
- [4] <http://www.monografias.com/trabajos93/meninges/meninges.shtml>
- [5] Ghaffari M, Hsu C, Linninger A. (2015) *Automatic reconstruction and generation of structured hexahedral mesh for non-planar bifurcations in vascular networks*.
- [6] Hsu C-Y, Schneller B, Ghaffari M, Alaraj A, Linninger A. (2015) *Medical image processing for fully integrated subjectspecific whole brain mesh generation*.
- [7] Andreas A. Linninger, Kevin Tangen, Chih-Yang Hsu, David Frim (2016). *Cerebrospinal Fluid Mechanics and Its Coupling to Cerebrovascular Dynamics*. Annual Reviews
- [8] Loth et al. (2000). *Studied flow in short segments of the spine and impact of annular geometric variation*
- [9] Tangen et al. (2015). *Studied impact of spinal microanatomy on complex flow profiles and on drug dispersion*
- [10] Amin-Hanjani S, Shin JH, Zhao M, Du X, Charbel FT (2007). *Evaluation of automatic measurement of the intracranial volume based on quantitative MR imaging*.
- [11] Kuttler A, Dimke T, Kern S, Helmlinger G, Stanski D, Finelli LA (2010). *Understanding pharmacokinetics using realistic computational models of fluid dynamics: biosimulation of drug distribution within the CSF space for intrahecal drugs*.
- [12] Bazilevs, Y.; Takizawa, K.; Tezduyar, T. E. (2013). *Computational Fluid-Structure Interaction*. Editorial Wiley
- [13] Vera, M.; Iglesias, I.; Sánchez, A. L.; Martínez, C.(2012). *Ingeniería fluidomecánica*. Editorial Paraninfo.
- [14] Ortiz Berrocal, L. (1998). *Elasticidad*. Editorial McGraw-Hill.
- [15] J.I. Jiménez González (Marzo, 2017). *Modelado matemático de problemas de mecánica de sólidos elásticos*
- [16] Hua-Dong Yao (Sept,2014). *Simulation of Fluid-Structural Interaction using OpenFOAM*
- [17] Vyzikas, Thomas (2015). *The implementation of InterFoam solver as a flow model of the fsiFoam solver for strong fluid-structure interaction*, Project work
- [18] A. L. Sánchez, C. Martínez-Bazán, C. Gutiérrez-Montes, E. Criado-Hidalgo, G. Pawlak, J. C. Lasheras (2017). *On the bulk motion of the cerebrospinal fluid in the spinal canal*.
- [19] ANSYS ICEM CFD Users' Manual

[20] <https://openfoamwiki.net/index.php/Installation/Linux/foam-extend-3.1>

[21] The Open Source CFD Toolbox. User Guide. 2016

[22] Vinje, Vegard (2016). *Fluid Flow and Spinal Cord Movement associated with Syringomyelia*, Master's Thesis

[23] <https://openfoamwiki.net/>

[24] <https://www.cfd-online.com/>

[25] <https://cpp.openfoam.org/v3/index.htm>

15. Anexo A – Archivos OpenFoam

15.1. Condiciones de contorno – Líquido

En este apartado se observan las condiciones de contorno de la parte líquida concretamente para la configuración A. Para la configuración B, tendríamos las mismas condiciones de contorno, pero sin la frontera llamada 'outer_wall'. Por último, para la configuración C no tendríamos ni la pared 'outer_wall' ni la salida 'outlet'.

15.1.1.P

Introducimos una onda de presión a la entrada mediante la función groovyBC e imponemos una condición de 'zeroGradient' a la salida. Esto es una condición de pseudo-tracción que implica una continuación del canal. La figura 16.1 nos indica cómo será el flujo desarrollado a la salida.

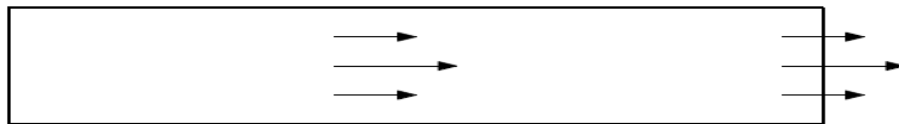


Figura 15.1 Flujo en el conducto [22]

```

/*-----* C++ -*-----*/
| ===== |
| \ \ / /   | F ield           | foam-extend: Open Source CFD |
| \ \ / /   | O peration        | Version:      3.0           |
| \ \ / /   | A nd              | Web:         http://www.extend-project.de |
| \ \ / /   | M anipulation     | |
|-----*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       p;
}
// *****

dimensions      [0 2 -2 0 0 0];

internalField    uniform 0;

boundaryField
{
    outlet
    {
        //type fixedValue;
        //value uniform 0;
        type zeroGradient;
    }
    inlet
    {

```

```

        type          groovyBC;
        valueExpression "1*cos(2*pi*time())";
        value uniform 0;
    }
    interface
    {
        type extrapolatedPressure;
        value uniform 0;
    }
    outer_wall
    {
        type extrapolatedPressure;
        value uniform 0;
    }
    frontAndBackPlanes
    {
        type          empty;
    }
}

// *****

```

15.1.2. pointMotionU

En este archivo cambiamos las condiciones de nodos fijos en el 'inlet' y en el 'outlet' (si existe) por una condición de deslizamiento ('slip'). Esta condición permite que la entrada y la salida aumente de tamaño transversalmente al igual que en todo el conducto.

```

/*----- C++ -----*/
|=====|
| \ \ / / F i e l d | foam-extend: Open Source CFD |
| \ \ / / O p e r a t i o n | Version: 3.0 |
| \ \ / / A n d | Web: http://www.extend-project.de |
| \ \ / / M a n i p u l a t i o n | |
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        pointVectorField;
    object        pointMotionU;
}
// *****

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

internalField    uniform (0 0 0);

boundaryField
{
    outlet
    {
        type slip;
    }
    inlet
    {
        type slip;
    }
    interface

```

```

{
    type fixedValue;
    value uniform (0 0 0);
}
outer_wall
{
    type slip;
}
frontAndBackPlanes
{
    type          empty;
}
}

// *****

```

15.1.3.U

En la velocidad también imponemos las condiciones de contorno de 'zeroGradient' a la entrada y a la salida. Y, en la interfase, establecemos una condición de pared móvil.

```

/*-----*- C++ -*-----*/
| ===== |
| \\ / F ield | foam-extend: Open Source CFD |
| \\ / O peration | Version: 3.0 |
| \\ / A nd | Web: http://www.extend-project.de |
| \\ / M anipulation |
/*-----*- C++ -*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0 0];
internalField    uniform (0 0 0);

boundaryField
{
    outlet
    {
        type zeroGradient;
    }
    inlet
    {
        type zeroGradient;
    }
    interface
    {
        type movingWallVelocity;
        value uniform (0 0 0);
    }
    outer_wall

```

```

{
    type fixedValue;
    value uniform (0 0 0);
}
frontAndBackPlanes
{
    type          empty;
}
}

// *****

```

15.2. Condiciones de contorno – Sólido

Al igual que en el apartado anterior, las condiciones de contorno que se muestran son las de la configuración A. Las fronteras altas y bajas del sólido son llamadas 'inlet_wall' y 'outlet_wall'. A estas fronteras les imponemos una condición de pared deslizante ('slip') que permite encogerse o estirarse con el fluido en la entrada y en la salida. Por otro lado, la frontera del sólido en la interfase posee una condición de desplazamiento.

Todas estas condiciones de contorno se mantienen en los cuatro archivos: 'D', 'DD', 'PointD' y 'PointDD'

Para la configuración B, tendremos que añadir aquí la frontera 'outer_wall' con una condición de pared rígida. Al igual tendríamos que hacer para la configuración C, pero también suprimiríamos la pared 'outlet_wall' y esta pasaría a formar parte de 'outer_wall'.

15.2.1.D

```

/*----- C++ -----*/
|=====|
|  \ \  /  | F i e l d      | foam-extend: Open Source CFD |
|  \ \  /  | O p e r a t i o n | Version:      3.0          |
|  \ \  /  | A n d             | Web:         http://www.extend-project.de |
|  \ \  /  | M a n i p u l a t i o n |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       D;
}
// *****

dimensions      [0 1 0 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField

```

```

{
    outlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }
    inlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }
    interface
    {
        type            tractionDisplacement;
        traction        uniform ( 0 0 0 );
        pressure        uniform 0;
        value           uniform (0 0 0);
    }
    frontAndBackPlanes
    {
        type            empty;
    }
}

// ***** //

```

15.2.2.DD

```

/*-----*- C++ -*-----*/
| ===== |
| \ \ / F i e l d | foam-extend: Open Source CFD |
| \ \ / O p e r a t i o n | Version: 3.0 |
| \ \ / A n d | Web: http://www.extend-project.de |
| \ \ / M a n i p u l a t i o n | |
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       DD;
}
// ***** //

dimensions      [0 1 0 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    outlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }
    inlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }
}

```

```

interface
{
    type          tractionDisplacementIncrement;
    traction      uniform ( 0 0 0 );
    pressure      uniform 0;
    value         uniform (0 0 0);
}
frontAndBackPlanes
{
    type          empty;
}
}

// ***** //

```

15.2.3.pointD

```

/*-----*- C++ -*-----*/
| ===== |
| \ \ / F i e l d | foam-extend: Open Source CFD |
| \ \ / O p e r a t i o n | Version: 3.0 |
| \ \ / A n d | Web: http://www.extend-project.de |
| \ \ / M a n i p u l a t i o n | |
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        pointVectorField;
    object       pointD;
}
// ***** //

dimensions      [0 1 0 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    outlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }
    inlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }
    interface
    {
        type          calculated;
        value         uniform (0 0 0);
    }
    frontAndBackPlanes
    {
        type          empty;
    }
}

// ***** //

```

15.2.4. pointDD

```

/*-----* C++ -*-----*/
|=====|
|  \ \  /  F i e l d      | foam-extend: Open Source CFD
|  \ \  /  O p e r a t i o n | Version:      3.0
|  \ \  /  A n d           | Web:          http://www.extend-project.de
|  \ \  /  M a n i p u l a t i o n |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        pointVectorField;
    object       pointDD;
}
// *****

dimensions      [0 1 0 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    outlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }

    inlet_wall
    {
        //type fixedValue;
        //value uniform (0 0 0);
        type slip;
    }

    interface
    {
        type          calculated;
        value          uniform (0 0 0);
    }

    frontAndBackPlanes
    {
        type          empty;
    }
}

// *****

```

15.3. Archivos de la carpeta ‘constant’ - Líquido

Estos archivos permanecen constantes para las tres configuraciones que han sido estudiadas ya que son independientes de la geometría con la que se está trabajando.

15.3.1.dynamicMeshDict

```

/*-----*- C++ -*-----*/
| =====
| \ \ / / F i e l d | foam-extend: Open Source CFD
| \ \ / / O p e r a t i o n | Version: 3.0
| \ \ / / A n d | Web: http://www.extend-project.de
| \ \ / / M a n i p u l a t i o n |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       dynamicMeshDict;
}
// *****

dynamicFvMesh    dynamicMotionSolverFvMesh;

solver velocityLaplacian;

diffusivity quadratic inverseDistance 1(interface);
diffusivity quadratic inverseDistance 1(outlet);

// *****

```

15.3.2.flowProperties

```

/*-----*- C++ -*-----*/
| =====
| \ \ / / F i e l d | foam-extend: Open Source CFD
| \ \ / / O p e r a t i o n | Version: 3.0
| \ \ / / A n d | Web: http://www.extend-project.de
| \ \ / / M a n i p u l a t i o n |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       flowProperties;
}
// *****

flowModel consistentIcoFlow;

icoFlowCoeffs
{
    nCorrectors 3;
    nNonOrthogonalCorrectors 1;
}

consistentIcoFlowCoeffs
{
    nCorrectors 3;
    nNonOrthogonalCorrectors 1;
}

```

```
// ***** //
```

15.3.3.fsiProperties

```
/*-----*- C++ -*-----*\
| =====|
| \ \ / F ield | foam-extend: Open Source CFD|
| \ \ / O peration | Version: 3.0|
| \ \ / A nd | Web: http://www.extend-project.de|
| \ \ / M anipulation |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       flowProperties;
}
// ***** //

solidPatch interface;
solidZone interface-zone;

fluidPatch interface;
fluidZone interface-zone;

relaxationFactor 0.05;

interfaceDeformationLimit 0;

outerCorrTolerance 1e-6;
nOuterCorr 30;

interpolatorUpdateFrequency 0;

//couplingScheme FixedRelaxation;
//couplingScheme Aitken;

couplingScheme IQN-ILS;
couplingReuse 1;

coupled yes;

// ***** //
```

15.3.4.transportProperties

```
/*-----*- C++ -*-----*\
| =====|
| \ \ / F ield | foam-extend: Open Source CFD|
| \ \ / O peration | Version: 3.0|
| \ \ / A nd | Web: http://www.extend-project.de|
| \ \ / M anipulation |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
```

```

class      dictionary;
object     flowProperties;
}
// * * * * *

nu         nu [0 2 -1 0 0 0] 3e-6;

rho        rho [1 -3 0 0 0 0] 1000;

// *****

```

15.4. Archivos de la carpeta ‘constant’ – Sólido

Los archivos que se describen a continuación, como en el apartado anterior, son independientes de la configuración que se está estudiando. No obstante, aquí se pueden modificar las propiedades del sólido y la localización de los puntos ‘probes’, lo que sí que cambiará.

15.4.1. rheologyProperties

```

/*-----*- C++ -*-----*\
| =====|
| \ \ / F i e l d | foam-extend: Open Source CFD|
| \ \ / O p e r a t i o n | Version: 3.0|
| \ \ / A n d | Web: http://www.extend-project.de|
| \ \ / M a n i p u l a t i o n |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       rheologyProperties;
}
// * * * * *

planeStress no;

rheology
{
    type         linearElastic;
    rho          rho [1 -3 0 0 0 0] 1200;
    E            E [1 -1 -2 0 0 0] 3e5; //2e4; 3e4; 6e4;
    nu           nu [0 0 0 0 0 0] 0.3;
}

// *****

```

15.4.2. stressProperties

```

/*-----*- C++ -*-----*\
| =====|
| \ \ / F i e l d | foam-extend: Open Source CFD|

```

```

|  \ \  /  O peration   | Version:    3.0   |
|  \ \  /  A nd         | Web:        http://www.extend-project.de |
|  \ \  /  M anipulation |              |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       stressProperties;
}
// *****

stressModel unsTotalLagrangianStress;

unsTotalLagrangianStressCoeffs
{
    nCorrectors 200;

    convergenceTolerance 1e-7;
    relConvergenceTolerance 1e-3;

    nonLinear no;
    debug no;
    moveMesh yes;
}

unsIncrTotalLagrangianStressCoeffs
{
    nCorrectors 200;

    convergenceTolerance 1e-7;
    relConvergenceTolerance 1e-3;

    nonLinear no;
    debug no;
    moveMesh yes;

    componentReference
    (
    );
}

// *****

```

15.5. Archivos de la carpeta 'system'

Los archivos que se presentan a continuación están presentes tanto en la carpeta del fluido como en la del sólido. Al estar la solución acoplada solo se requiere un archivo de cada uno, por defecto, se toman los archivos de la carpeta 'fluid'.

15.5.1.controlDict

```

/*-----*-- C++ --*-----*\
| =====
|  \ \  /  F ield       | foam-extend: Open Source CFD
|  \ \  /  O peration   | Version:    3.0
|  \ \  /  A nd         | Web:        http://www.extend-project.de

```

```

|      \\\      M anipulation      |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       controlDict;
}
// * * * * *

application      fsiFoam;

startFrom        startTime;

startTime        0;

stopAt           endTime;

endTime          20;//10;

deltaT           0.0001;

writeControl      timeStep;

writeInterval     100;

purgeWrite        0;

writeFormat       ascii;

writePrecision    10;

writeCompression  uncompressed;

timeFormat        general;

timePrecision     10;

runTimeModifiable yes;

adjustTimeStep    no;

maxCo             0.2;

libs (
    "libOpenFOAM.so"
    "libsimpleSwakFunctionObjects.so"
    "libswakFunctionObjects.so"
    "libgroovyBC.so"
);

functions
(
    pointHistory
    {
        type pointHistory;
        functionObjectLibs
        (
            "libpointHistory.so"
        );

        region solid;

        refHistoryPoint (0 0.005 0.025);
    }
    probes
    {

```

```
// Where to load it from
functionObjectLibs ( "libsampling.so" );

type          probes;

// Name of the directory for probe data
name          probes;

interpolationScheme cellPointFace;
// Fields to be probed
fields (
    p
    U
);

probeLocations
(
//      Coordinates with the form ( x y z=0 )

);

// Write at same frequency as fields

outputControl    timeStep;
outputInterval    100;

}

);

// ***** //
```

15.5.2.decomposeParDict

```
/*-----*- C++ -*-----*/
|=====|
| \\\ / | F ield | OpenFOAM: The Open Source CFD Toolbox |
| \\\ / | O peration | Version: 1.5 |
| \\\ / | A nd | Web: http://www.OpenFOAM.org |
| \\\ / | M anipulation |
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       decomposeParDict;
}
// ***** //

numberOfSubdomains 3;

method            simple;

globalFaceZones ( interface-zone );

simpleCoeffs
{
    n              (3 1 1);
    delta          0.001;
}

distributed       no;
```

```

roots
(
);

// *****

```

15.5.3. fvSchemes

```

/*-----*- C++ -*-----*/
|=====|
| \ \ / | F i e l d | foam-extend: Open Source CFD |
| \ \ / | O p e r a t i o n | Version: 3.0 |
| \ \ / | A n d | Web: http://www.extend-project.de |
| \ \ / | M a n i p u l a t i o n | |
|=====|
/*-----*- C++ -*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       fvSchemes;
}
// *****

ddtSchemes
{
    default backward;
}

gradSchemes
{
    default leastSquares;
}

divSchemes
{
    default none;
    div(phi,U) Gauss skewCorrected linear;
//    div(phi,U) Gauss linearUpwind cellLimited leastSquares 1;
}

laplacianSchemes
{
    default none;
    laplacian(nu,U) Gauss linear skewCorrected 1;
    laplacian((1|A(U)),p) Gauss linear skewCorrected 1;
    laplacian(diffusivity,cellMotionU) Gauss linear skewCorrected 1;
}

interpolationSchemes
{
    default linear;
    interpolate(U) skewCorrected linear;
}

snGradSchemes
{
    default skewCorrected 1;
}

fluxRequired
{
    default no;
}

```

```

    p;
}

// *****

```

15.5.4.fvSolution

```

/*-----*- C++ -*-----*/
|=====|
| \ \ / / F i e l d | foam-extend: Open Source CFD |
| \ \ / / O p e r a t i o n | Version: 3.0 |
| \ \ / / A n d | Web: http://www.extend-project.de |
| \ \ / / M a n i p u l a t i o n | |
/*-----*- C++ -*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       fvSolution;
}

// *****

solvers
{
    p
    {
        solver          GAMG;
        tolerance        1e-06;
        relTol           0;
        minIter          1;
        maxIter          1000;
        smoother         GaussSeidel;
        nPreSweeps        0;
        nPostSweeps       2;
        nFinestSweeps     2;
        scaleCorrection   true;
        directSolveCoarsest false;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 20;
        agglomerator      faceAreaPair;
        mergeLevels       1;
    }

    cellMotionU
    {
        solver          GAMG;
        tolerance        1e-06;
        relTol           0.001;
        minIter          1;
        maxIter          100;
        smoother         GaussSeidel;
        nPreSweeps        0;
        nPostSweeps       2;
        nFinestSweeps     2;
        scaleCorrection   true;
        directSolveCoarsest false;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 20;
        agglomerator      faceAreaPair;
        mergeLevels       1;
    }
}

```

```

U
{
    solver          PBiCG;
    preconditioner  DILU;
    tolerance       1e-06;
    relTol          0;
    minIter         1;
}

// *****

```

16. Anexo B – Archivos MatLab

Los programas que a continuación se describen se usan para las tres configuraciones, pero habrá que adaptarlos para cada una en función de los resultados y geometría del canal.

16.1. Lectura presión

Este programa se usa para leer el archivo 'p.txt' que genera OpenFoam que contiene la presión a lo largo del tiempo en los puntos de la configuración que hemos indicado y genera matriz 'p' con todos estos datos además de tres con las coordenadas de los puntos. Luego representa una gráfica de la presión respecto al tiempo del punto que indiquemos.

```

fileID = fopen('p.txt','r'); %Abrir archivo

%Matriz x
formatSpec = '%c %c';
sizeA = [1 2];
A=fscanf(fileID,formatSpec,sizeA);
x=fscanf(fileID,'%f',[1 Inf]);
%Matriz y
B=fscanf(fileID,formatSpec,sizeA);
y=fscanf(fileID,'%f',[1 Inf]);
%Matriz z
C=fscanf(fileID,formatSpec,sizeA);
z=fscanf(fileID,'%f',[1 Inf]);

%Time
formatSpec = '%c %c%c%c%c';
sizeA = [1 5];
T=fscanf(fileID,formatSpec,sizeA);

for i=1:1000 %Número de instantes de tiempo
    p(i,:)=fscanf(fileID,'%f',[1 50]);
end

pto=26; %punto a representar

```

```
plot(p(:,1),p(:,pto)); %presión respecto al tiempo
xlabel('t (s)', 'fontweight', 'bold'), ylabel('p (Pa)', 'fontweight', 'bold')
```

16.2. Lectura velocidad

Este programa realiza lo mismo que el anterior, pero para el caso de la velocidad. Se almacenan las velocidades en las direcciones principales, aunque solo nos interesa la velocidad en el eje 'y'.

```
fileID = fopen('U.txt', 'r');

%Matriz x
formatSpec = '%c %c';
sizeA = [1 2];
A=fscanf(fileID, formatSpec, sizeA);
x=fscanf(fileID, '%f', [1 Inf]);
%Matriz y
B=fscanf(fileID, formatSpec, sizeA);
y=fscanf(fileID, '%f', [1 Inf]);
%Matriz z
C=fscanf(fileID, formatSpec, sizeA);
z=fscanf(fileID, '%f', [1 Inf]);

%Time
formatSpec = '%c %c%c%c%c';
sizeB = [1 5];
T=fscanf(fileID, formatSpec, sizeB);
i=1; j=1;
for i=1:1000
    U(i,1)=fscanf(fileID, '%f', [1]);
    for j=1:3:145
        delete=fscanf(fileID, '%c', [1]);
        U(i,j+1)=fscanf(fileID, '%f', [1]);
        delete=fscanf(fileID, '%c', [1]);
        U(i,j+2)=fscanf(fileID, '%f', [1]);
        delete=fscanf(fileID, '%c', [1]);
        U(i,j+3)=fscanf(fileID, '%f', [1]);
        delete=fscanf(fileID, '%c', [1]);
    end
end

figure
pto=75; %punto a representar
plot(U(:,1),U(:,pto)); %velocidad en y
xlabel('t (s)', 'fontweight', 'bold'), ylabel('U_y (m/s)', 'fontweight', 'bold')
%plot(U(:,1),U(:,pto-1)); %velocidad en x
%plot(U(:,1),U(:,pto+1)); %velocidad en z
```

16.3. Perfiles de velocidad media

Por último, necesitamos un programa que lea la velocidad a lo largo del tiempo para los puntos 'probes', calcule la media y represente el perfil de velocidad media que se forma.

```
clear Upto      %Upto es un array con las velocidades en...
                %...un punto para un intervalo de tiempo
clear tiempo   %tiempo es un array con el intervalo de tiempo
i=1;
Umedia=zeros(7,9); %las velocidades medias se almacenan en Umedia
columna=2;
cont=1;
for n=1:49
    cU=3+3*(n-1) %columna posición en U
    for f=250:550 %intervalo de tiempo
        c=1; %columna tiempo
        tiempo(i,1)=U(f,c);
        Upto(i,1)=U(f,cU);
        i=i+1;
    end
    i=1;
    if cont<=7
        Umedia(cont,columna)=mean(Upto)
    else
        cont=1;
        columna=columna+1;
        Umedia(cont,columna)=mean(Upto)
    end
    cont=cont+1;
end

%espesor del canal por el que fluye el LCR
e=[-0.009 -0.0085 -0.008 -0.0075 -0.007 -0.0065 -0.006 -0.0055 -0.005];
xx=-0.009:0.00001:-0.005;

figure
%subplot(3,1,1);
yyal=spline(e,Umedia(1,:),xx); %genera una curva por los puntos calculados
plot(e,Umedia(1,:), 'ko',xx,yyal, 'r')
title('y^+=1/12');
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')

figure
%subplot(3,1,2);
yyal=spline(e,Umedia(2,:),xx);
plot(e,Umedia(2,:), 'ko',xx,yyal, 'r')
title('y^+=1/6');
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')

figure
%subplot(3,1,3);
yyal=spline(e,Umedia(3,:),xx);
plot(e,Umedia(3,:), 'ko',xx,yyal, 'r')
title('y^+=1/3');
```

```
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')

figure
%subplot(3,1,3);
yya1=spline(e,Umedia(4,:),xx);
plot(e,Umedia(4,:), 'ko',xx,yya1, 'r')
title('y^+=1/2');
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')

figure
%subplot(3,1,3);
yya1=spline(e,Umedia(5,:),xx);
plot(e,Umedia(5,:), 'ko',xx,yya1, 'r')
title('y^+=2/3');
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')

figure
%subplot(3,1,3);
yya1=spline(e,Umedia(6,:),xx);
plot(e,Umedia(6,:), 'ko',xx,yya1, 'r')
title('y^+=5/6');
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')

figure
%subplot(3,1,3);
yya1=spline(e,Umedia(7,:),xx);
plot(e,Umedia(7,:), 'ko',xx,yya1, 'r')
title('y^+=11/12');
xlabel('e (m)', 'fontweight', 'bold'), ylabel('U_{y} (m/s)', 'fontweight', 'bold')
```