



ESCOLA POLITÉCNICA SUPERIOR  
CAMPUS UNIVERSITARIO DE ESTEIRO  
MENDIZÁBAL S/N, 15403 FERROL - ESPAÑA

# Estudio del golpe de ariete en bombas volumétricas

Análisis fluidodinámico de los transitorios de presión

*Trabajo Fin de Grado*

*Tutores: Marcos Lema Rodríguez y Anne Gosset*

David Adrián Rodríguez García

Junio de 2016



ESCOLA POLITÉCNICA SUPERIOR



# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Introducción	7
1.2. Objetivos	9
1.3. Estructura del TFG	9
<b>2. Recursos literarios</b>	<b>11</b>
2.1. Introducción	11
2.2. Golpe de ariete	11
2.2.1. Propagación de las ondas de presión	11
2.3. Interacción fluido estructura	12
2.3.1. Acoplamiento de fricción	13
2.3.2. Acoplamiento de Poisson	14
2.3.3. Acoplamiento de las uniones	14
2.4. Cierre de válvulas esféricas	15
2.5. Dinámica de fluidos computacional (CFD)	16
2.5.1. Pre-proceso	16
2.5.2. Solver o resolución	16
2.5.3. Post-proceso	17
2.6. Fatiga	17
2.6.1. Cálculo de la resistencia a fatiga a partir de la tensión	17
2.6.2. Cálculo de la resistencia a fatiga a partir de la deformación	18
2.6.3. Fatiga por crecimiento de grieta	19
<b>3. Golpe de ariete</b>	<b>21</b>
3.1. Introducción al problema	21
3.1.1. Definición de la geometría	21

3.1.2. Hipótesis sobre el problema . . . . .	22
3.2. Estudio analítico . . . . .	22
3.2.1. Cálculo del área de paso en relación al giro de la válvula . . . . .	23
3.2.2. Conclusión . . . . .	24
<b>4. Sobrepresión interna</b>	<b>27</b>
4.1. Planteamiento de la hipótesis . . . . .	27
4.2. Elección del modelo de compresibilidad . . . . .	27
<b>5. Simulación numérica computacional</b>	<b>31</b>
5.1. Estudio previo . . . . .	31
5.1.1. Geometría . . . . .	31
5.1.2. Simulación . . . . .	32
5.1.2.1. Implementación de la malla dinámica . . . . .	32
5.1.2.2. Resolución numérica . . . . .	34
5.1.2.3. Esquemas numéricos aplicados . . . . .	34
5.1.3. Estudio comparativo de las condiciones de contorno . . . . .	35
5.2. Estudio del caso real . . . . .	36
5.2.1. Geometría . . . . .	37
5.2.2. Simulación . . . . .	37
5.2.2.1. Implementación del solver . . . . .	37
5.2.2.2. Implementación de las propiedades físicas del fluido de trabajo . . . . .	38
5.2.2.3. Esquemas numéricos . . . . .	38
5.2.3. Verificación de los resultados simulados con los calculados analíticamente . . . . .	39
5.2.4. Estudio de la sensibilidad de malla . . . . .	39
5.2.5. Conclusión del estudio . . . . .	40
5.3. Simulación con simetría 3D . . . . .	42
5.4. Análisis de los resultados . . . . .	43
<b>6. Conclusiones y propuesta de soluciones</b>	<b>45</b>
6.1. Conclusiones . . . . .	45
6.2. Planteamiento de soluciones al problema . . . . .	46
<b>A. <i>movingConeTopoFVMesh_v2</i></b>	<b>51</b>

# Índice de figuras

1.1. Secuencia de movimiento de la bomba volumétrica . . . . .	8
2.1. Propagación de las ondas de presión $F$ y $f$ . . . . .	12
2.2. Ciclo de propagación de las ondas de presión . . . . .	13
2.3. Acoplamiento de Poisson. . . . .	14
2.4. Acoplamiento de las uniones. . . . .	15
2.5. Variación del área de paso de una válvula esférica . . . . .	15
2.6. Diagrama S-N . . . . .	18
2.7. Ciclo de control de deformación de un material elastoplástico . . . . .	18
3.1. Esquema de la geometría de la bomba . . . . .	21
3.2. Modelo de variación del área de paso . . . . .	23
3.3. Vista simplificada en planta de la válvula giratoria . . . . .	24
3.4. Ángulo de apertura y cierre . . . . .	24
3.5. Variación del área de paso de fluido respecto al tiempo . . . . .	25
3.6. Variación del área de paso de fluido respecto al ángulo girado por la válvula . . . . .	25
4.1. Etapa de compresión estática del fluido . . . . .	29
4.2. Comparativa de las funciones barotrópicas . . . . .	29
5.1. Geometría del estudio previo $R = 15mm$ . . . . .	32
5.2. Tutorial de la utilidad <i>movingConeTopoFvMesh</i> de <i>OpenFOAM</i> . . . . .	33
5.3. Estudio de las condiciones de contorno . . . . .	37
5.4. Comparativa de los resultados analíticos y simulados . . . . .	40
5.5. Estudio de la sensibilidad de malla . . . . .	41
5.6. Fuerzas sobre la cabeza del pistón . . . . .	42
5.7. Simulación 3D con simetría de la bomba volumétrica . . . . .	44

---

5.8. Comparativa de los resultados analíticos y simulados con el nuevo volumen . . . .	44
6.1. Solución mediante bypass . . . . .	47
6.2. Solución por modificación del área de paso . . . . .	47

# Capítulo 1

## Introducción

### 1.1. Introducción

Las máquinas hidráulicas son máquinas cuya misión es transformar la energía, por lo general mecánica, que las acciona en energía aportada al fluido incompresible (o de compresibilidad despreciable) en forma de presión, velocidad o altura. Existen dos grandes grupos de máquinas hidráulicas, las máquinas dinámicas y las máquinas de desplazamiento positivo o volumétricas, este informe se centra en este último tipo, más concretamente se estudiará una bomba volumétrica de tipo pistón.

Una empresa del sector alimentario ha sufrido diversos problemas en sus líneas de producción debido a la rotura de las bombas que utilizan en sus cadenas de procesado de alimentos. Estas bombas se encargan de bombear productos como mayonesa, mermelada, yogur, etc. . La razón de sus percances se debe a que las bombas volumétricas que utilizan en sus procesos sufren rotura por fatiga al cabo de cierto tiempo. A lo largo de este informe se tratarán diferentes aspectos del por qué ocurre este fenómeno y se plantearán alternativas para su solución. Comenzando por la rama de investigación del golpe de ariete, puesto que es la principal sospecha de problemas para la empresa que utiliza este tipo de bombas.

Las bombas que se utilizan en los procesos son de tipo volumétrico, están diseñadas como un pistón de cabeza cónica redondeada en la punta. El proceso de entrada y salida de fluido viene regulado por una válvula giratoria con un orificio que conecta, las tuberías de aspiración e impulsión de fluido, separadas una da la otra  $180^\circ$ , con la cámara de la bomba.

Cuando el desplazamiento del pistón es descendente, la cámara de la bomba se llena de fluido, y cuando éste alcanza el punto muerto inferior la válvula se encuentra girando hacia la posición de impulsión. Acto seguido el pistón comienza a realizar el movimiento de ascensión que termina con la impulsión del fluido a través de la tubería de salida.

Se puede observar con más claridad el movimiento descrito en el vídeo que se adjunta a continuación: **Movimiento de la bomba volumétrica**. La figura 1.1 muestra una secuencia del movimiento (de arriba a abajo) de la bomba, extraída del vídeo referenciado.

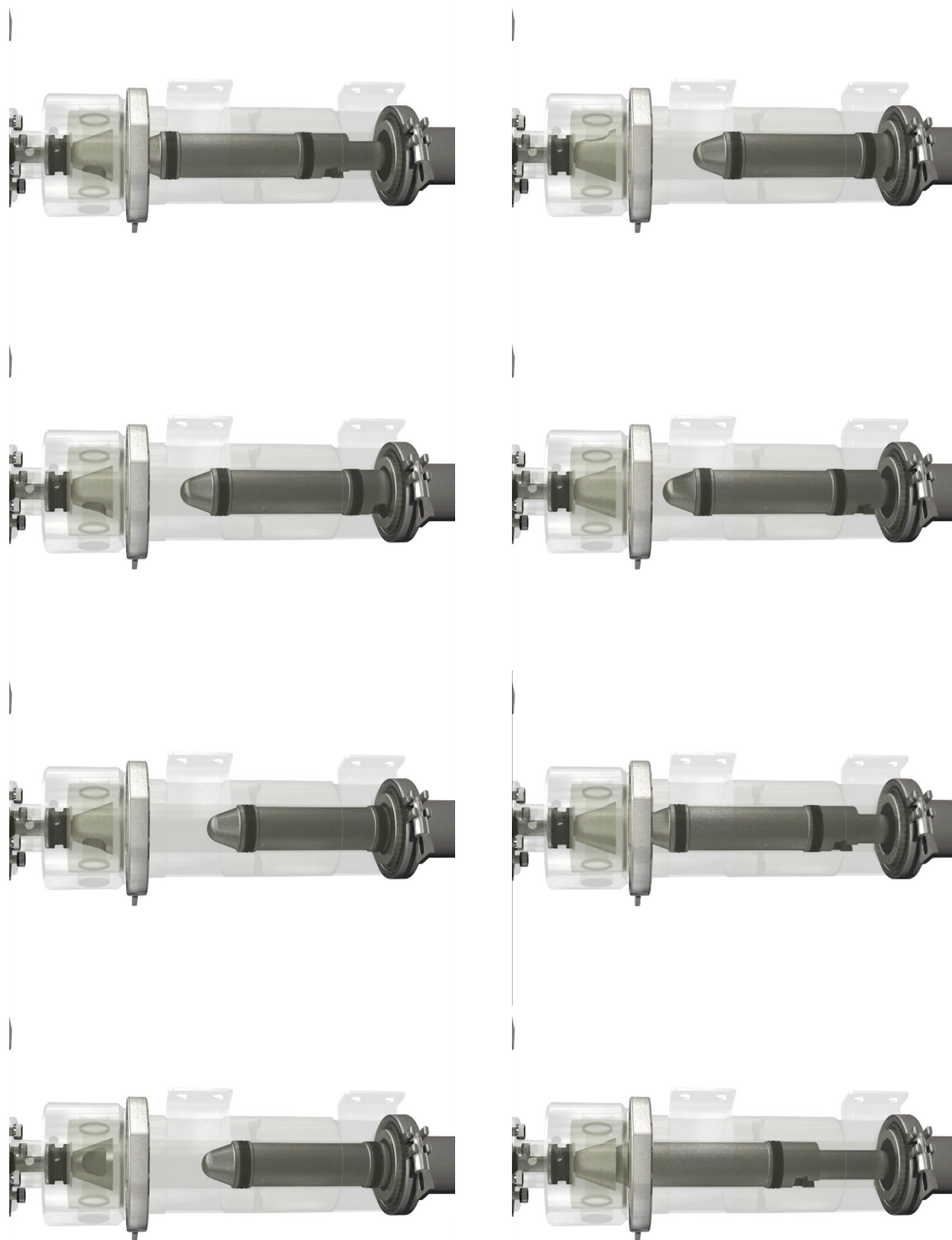


Figura 1.1: Secuencia de movimiento de la bomba volumétrica



## 1.2. Objetivos

Los objetivos que se persiguen en este informe son los de detectar qué problemas atañen al proceso de fatiga de la bomba volumétrica. También su grado de importancia a la hora de provocar la ruptura de la misma, tanto de forma cualitativa como cuantitativa mediante el apoyo de software de simulación computacional CFD. Por último como objetivo final se buscarán soluciones al problema y se le propondrán a la empresa para que decidan ellos cual es la que mejor se adapta a su utilización en los procesos.

## 1.3. Estructura del TFG

La estructura se divide en capítulos para tratar de alcanzar en ellos los objetivos descritos previamente.

En el capítulo 2 se presentan los recursos literarios empleados. Se han estudiado diferentes artículos y libros acerca de temas que atañen al problema en cuestión, como son el fenómeno de golpe de ariete, el cierre de válvulas esféricas, la interacción fluido estructura, la fatiga como causa de la rotura, y simulación CFD.

En el capítulo 3 se analiza de forma teórica la influencia que tiene el golpe de ariete en el caso de la rotura de las bombas, atendiendo a diferentes fenómenos como la rotura de columna líquida.

En el capítulo 4 se aborda el problema desde un nuevo enfoque relacionado con la sincronización entre la válvula de entrada y salida de fluido y el movimiento del pistón, analizando cualitativamente el aumento de la presión interna dentro de la cámara del pistón.

En el capítulo 5 se pone en práctica lo supuesto en el capítulo 4 mediante una simulación de dinámica computacional de fluidos bidimensional que permita obtener valores cuantitativos de esa sobrepresión interna en la cámara del cilindro.

En el capítulo 6 se comentan los resultados obtenidos a lo largo de todo el estudio realizado, se proponen a la empresa una serie de medidas a adoptar que solucionen el problema de la rotura por fatiga y se resumen las conclusiones más importantes obtenidas de este estudio.



# Capítulo 2

## Recursos literarios

### 2.1. Introducción

Este capítulo presenta los recursos literarios que se han utilizado para llevar a cabo el estudio sobre los posibles problemas que ocurren en este tipo de bombas volumétricas, tanto analíticamente como numéricamente. Las referencias más relevantes han sido agrupadas en cinco categorías, listadas a continuación:

- Golpe de ariete
- Interacción fluido estructura
- Cierre de válvulas esféricas
- Simulación computacional de dinámica de fluidos, CFD
- Estudio de la rotura por fatiga

### 2.2. Golpe de ariete

#### 2.2.1. Propagación de las ondas de presión

El fenómeno de golpe de ariete principalmente está originado por variaciones bruscas de velocidad en la circulación de un fluido, como las que son producidas al cerrar repentinamente una válvula que impide el paso de fluido por una tubería. Cuando esto ocurre las partículas próximas a la válvula se detienen y las que se encuentran aguas arriba todavía siguen viajando hacia la válvula con la velocidad de antes de cerrarla, estas partículas al llegar a la válvula chocan con las que allí se encontraban ya detenidas y generan una fuerte sobrepresión que puede llegar a ser de un orden de magnitud superior a la presión de operación de la tubería, esta onda de presión viaja aguas arriba a la velocidad del sonido en el seno del fluido. A continuación se muestra una figura que ilustra mejor las ondas que se propagan durante el fenómeno descrito.

La figura 2.1 nos muestra la propagación de las ondas de golpe de ariete a través del fluido, refiriéndose la onda F a la originaria que se produce con el cierre de la válvula de control y la onda f es la generada como consecuencia del reflejo de las condiciones de contorno del depósito situado aguas arriba. Son ondas de igual magnitud pero de sentidos contrarios, esta forma de

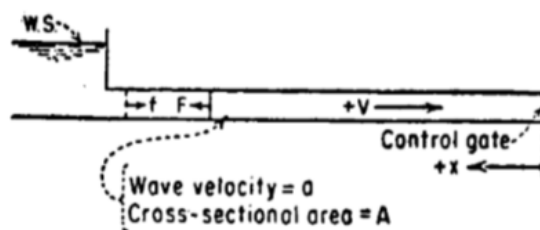


Figura 2.1: Propagación de las ondas de presión  $F$  y  $f$

explicar la propagación de ondas de presión en el seno de una tubería llena de fluido podemos encontrarlo en [6]

En la figura 2.2 se explica con más detalle pudiendo observar que el proceso de golpe de ariete es un proceso cíclico amortiguado por la fricción viscosa que presenta el fluido. En este proceso podemos observar en primer lugar la propagación aguas arriba de la onda de sobrepresión generada por el cierre de la válvula de paso, durante un tiempo  $t = \frac{L}{a}$  donde  $a$  es la velocidad del sonido en dicho fluido. Al alcanzar la onda de presión el depósito, como el volumen es muy grande en comparación con el de fluido en la tubería, las condiciones de contorno del tanque son impuestas a la tubería, y por tanto se genera una onda reflejada con las condiciones del depósito, es decir su presión interior, esta nueva onda se propaga aguas abajo y genera una diferencia de presión entre las partículas fluidas que están aguas arriba de la onda y las que están aguas abajo provocando que estas últimas se muevan con velocidad en sentido hacia el depósito, la onda alcanza la válvula cerrada en el tiempo  $t = \frac{2L}{a}$ . Cuando esto ocurre todo el fluido de la tubería está moviéndose aguas arriba lo que provoca que se genere en la zona de la válvula una depresión que se propaga de nuevo en forma de onda aguas arriba y que alcanza el depósito en  $t = \frac{3L}{a}$ . En este momento se genera la condición de presión del depósito y por tanto aparece una nueva diferencia de presiones que provoca que el fluido se desplace aguas abajo llegando a impactar contra la válvula cerrada en  $t = \frac{4L}{a}$  generando un nuevo golpe de ariete y por tanto repitiéndose todo el ciclo.

### 2.3. Interacción fluido estructura

La interacción fluido estructura se presenta como una extensión de la teoría convencional del golpe de ariete dado que los sistemas de tuberías sufren severas fuerzas dinámicas a causa del golpe de ariete, cuando estas fuerzas hacen al sistema moverse ocurren significativas interacciones fluido estructura y el fluido y el sistema de tuberías no puede ser tratado por separado, es necesario tener en cuenta los mecanismos de interacción que intervienen.

En el análisis de golpe de ariete convencional, la elasticidad de la tubería se incorpora en la velocidad de propagación de las ondas de presión. La inercia de la tubería y el movimiento axial de la tubería no se tienen en cuenta. Esto es aceptable para los sistemas de tuberías rígidamente anclados. Para los sistemas menos restringidos, la interacción fluido estructura puede llegar a ser de importancia. En ese caso, el comportamiento dinámico del sistema de fluido y tubería debe ser tratado simultáneamente.

Tres mecanismos de interacción fluido estructura serán tratados a continuación [8]:

- Acoplamiento de fricción
- Acoplamiento de Poisson

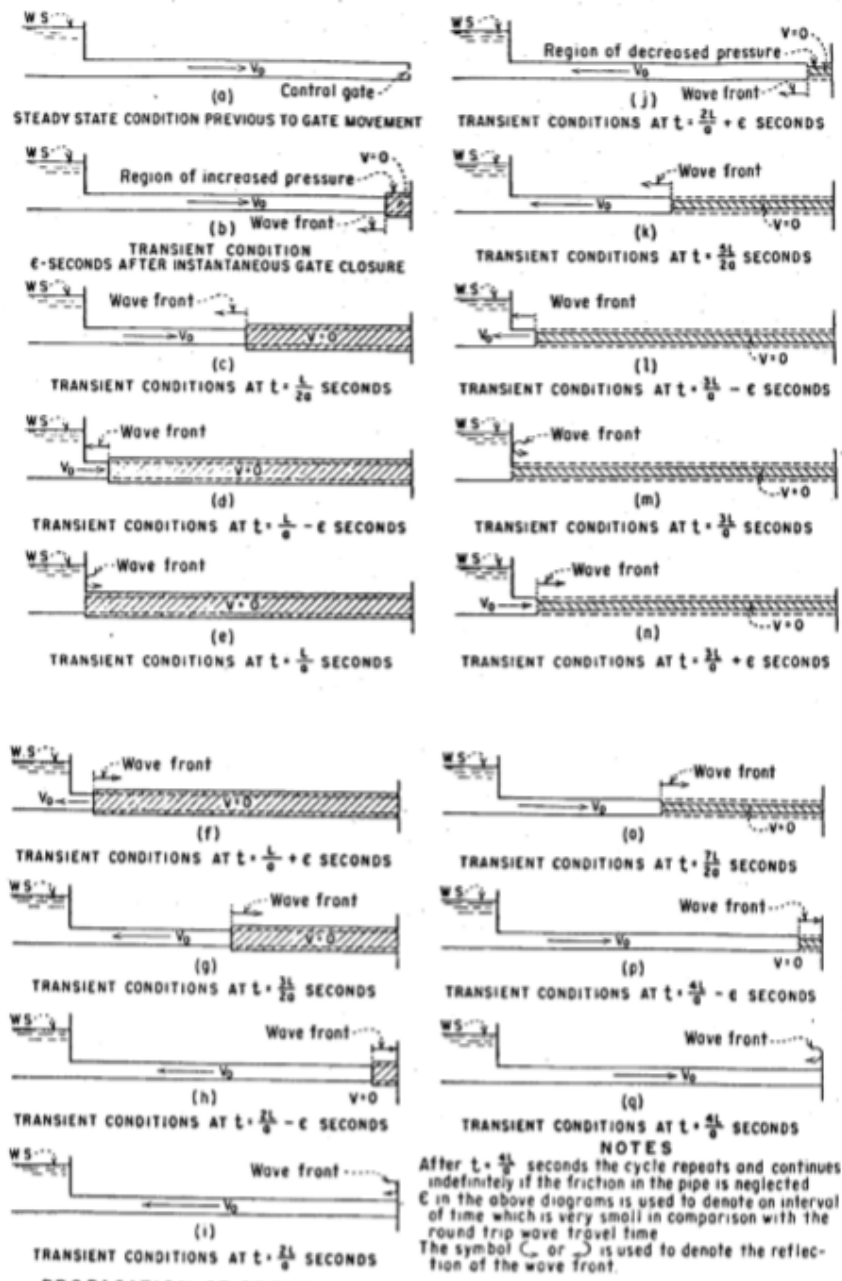


Figura 2.2: Ciclo de propagación de las ondas de presión

- Acoplamiento de las uniones

### 2.3.1. Acoplamiento de fricción

El acoplamiento de fricción simplemente representa la fricción mutua existente entre el fluido del interior de la tubería y las paredes de la misma.

### 2.3.2. Acoplamiento de Poisson

El acoplamiento de Poisson relaciona las presiones en el líquido con el estrés axial (longitudinal) en la tubería a través de la contracción radial o expansión de las paredes de la misma. Lleva el nombre de Poisson en relación con su coeficiente de contracción. El acoplamiento de Poisson lleva a la generación de las ondas precursoras que se propagan a través del material de la tubería a mayor velocidad  $a$  (velocidad del sonido) que las ondas de golpe de ariete clásicas, por lo que van por delante de estas en su propagación. En la figura 2.3 se muestra un ejemplo del acoplamiento de Poisson.

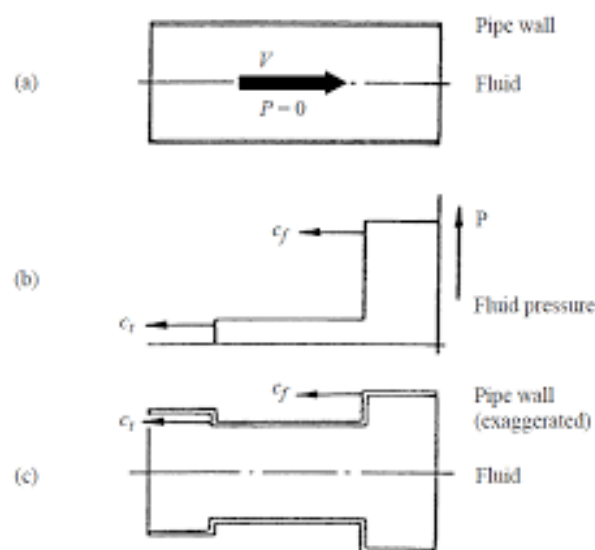


Figura 2.3: Acoplamiento de Poisson.

En (a) el fluido está fluyendo a velocidad  $V$  y presión de referencia  $P = 0$ . En (b) el fluido se detiene instantáneamente, resultando en un aumento de presión que se propaga a la velocidad  $c_f$  a través del fluido. El incremento de presión es acompañado por una expansión radial de la pared de la tubería como se muestra en (c). Debido a esta expansión radial, la tubería se acorta detrás y se elonga en frente del incremento de presión. La elongación se revela como una onda de estrés axial. Ésta se propaga a la velocidad  $c_t$  a través de la tubería y causa una contracción radial de las paredes de la misma como en (c). La contracción radial causa un segundo incremento de presión en el fluido como se observa en (b). Este secundario aumento de presión es generalmente debido a una onda precursora que se propaga a la velocidad  $c_t$ , que es generalmente superior a la velocidad de propagación  $c_f$  de la primera onda de incremento de presión.

### 2.3.3. Acoplamiento de las uniones

El acoplamiento de las uniones actúa en puntos específicos en un sistema de tuberías, tales como válvulas sin restricciones, curvas y tes. Un ejemplo típico es el codo de vibración, el cual induce ondas de presión en el líquido a través de una acción de bombeo combinado (compresión) y de almacenamiento (descompresión). Otro ejemplo es el puente de tuberías que se muestra en la figura 2.4.

Cuando una onda de presión ha pasado el codo derecho en (a), la diferencia neta de presiones entre los dos codos causa al puente de tuberías moverse como en (b). Debido al movimiento la presión desciende en el derecho y aumenta en el codo izquierdo, como se muestra en (b). El

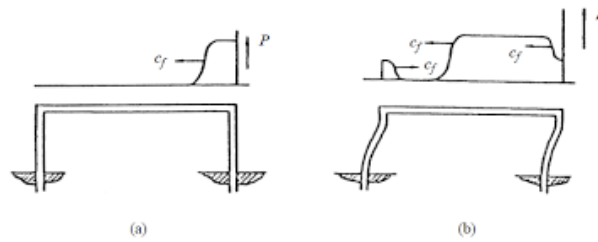


Figura 2.4: Acoplamiento de las uniones.

movimiento del puente induce ondas de presión en el fluido, que en el regreso influyen también en el movimiento del puente.

## 2.4. Cierre de válvulas esféricas

Las válvulas esféricas son usadas en multitud de aplicaciones, para nuestro caso en concreto se ha profundizado en como es el cierre de las mismas por similitud con el que presenta la válvula giratoria de la bomba volumétrica que se está estudiando.

Según el artículo [3] el área de paso de las válvulas esféricas puede ser determinada por el área de la línea sólida que se muestra en la figura 2.5. Es sencillo usar una aproximación geométrica para encontrar las coordenadas  $x'$  e  $y'$  que indican el camino de la línea solida de manera que satisfaga las ecuaciones siguientes:

$$x' = r' [2\cos^2(2\delta) - 1] \cos(\theta) - h\sin(\theta) \quad (2.1)$$

$$y' = r' \sin(2\delta) \quad (2.2)$$

$$(x')^2 + (y')^2 \leq (r')^2 \quad (2.3)$$

Donde  $\theta$  es el ángulo de giro de la válvula esférica desde la posición de apertura total,  $r'$  es el radio del agujero circular,  $h$  es la distancia desde el centro de la válvula de bola hasta la superficie del agujero y  $\delta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$

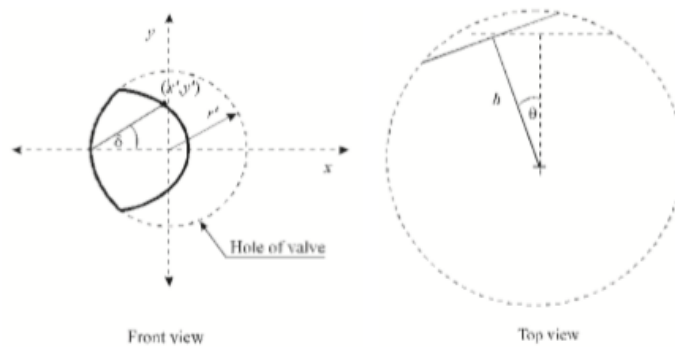


Figura 2.5: Variación del área de paso de una válvula esférica

## 2.5. Dinámica de fluidos computacional (CFD)

CFD es el análisis de sistemas que implican flujo de fluido, transferencia de calor y fenómenos asociados tales como reacciones químicas por medio de la simulación por ordenador [9].

En cualquier programa de CFD se distinguen siempre 3 elementos principales:

- Pre-proceso
- Solver o resolución
- Post-proceso

### 2.5.1. Pre-proceso

El pre-proceso consiste en la introducción de un problema de flujo en un programa de CFD por medio de una interfaz de usuario y la posterior transformación de esta entrada en una forma adecuada para uso por el solver. Los pasos a seguir por el usuario para establecer las condiciones de su problema a resolver son las siguientes:

1. Definición de la geometría de la región de interés: el dominio computacional
2. Generación de la malla, la subdivisión del dominio en un número de pequeños y no solapados dominios denominados celdas o volúmenes de control.
3. Selección de los fenómenos físicos y químicos que necesitan ser modelados, por ejemplo una combustión, un fenómeno fluido laminar o turbulento, etc.
4. Definición de las propiedades del fluido, de las cuales dependerá el tipo de solver a utilizar, como por ejemplo si el fluido es compresible o incompresible o si se tienen en cuenta los efectos de transmisión de calor o no, etc.
5. Especificación de las condiciones de contorno apropiadas con las celdas que coinciden con o tocan los límites del dominio, es de gran importancia este aspecto del pre-proceso para que la simulación alcance un rigor y comportamiento físico adecuado al del caso que se trata de simular.

### 2.5.2. Solver o resolución

Existen tres corrientes distintas de técnicas de solución numérica: diferencias finitas, elementos finitos y el método espectral. En este trabajo se hablará solo del método de volúmenes finitos, una formulación especial del método de diferencias finitas, que es el que usaremos en la resolución de nuestro caso. De forma esquemática el algoritmo numérico de resolución consiste en los siguientes pasos:

1. Integración de las ecuaciones que gobiernan el flujo del fluido en todos los finitos volúmenes de control del dominio.
2. Discretización, conversión de las ecuaciones integrales resultantes en un sistema algebraico de ecuaciones.



3. Solución de las ecuaciones algebraicas mediante un método iterativo.

Dos de los algoritmos más conocidos y utilizados son los algoritmos SIMPLE y PISO.

### 2.5.3. Post-proceso

Los softwares actuales incluyen una gran variedad de herramientas de visualización de los resultados obtenidos como son:

- Visualización de la geometría de dominio y de la malla
- Gráficos de vectores
- Línea y gráficos de contorno sombreados
- Superficies 2D Y 3D
- Visualización del movimiento de las partículas fluidas
- Manipulación de la vista (rotación, traslación, escalado, etc)
- Animación del resultado dinámico obtenido

Sin duda una gran herramienta para obtener una mejor interpretación de los resultados obtenidos y de su rigor físico.

## 2.6. Fatiga

Se entiende por fatiga de materiales a un fenómeno por el cual la rotura de los materiales bajo cargas dinámicas cíclicas se produce más fácilmente que con cargas estáticas.

En este caso y por simplificación de la complejidad del fenómeno se hablará de los casos de fatiga producidos bajo una carga uniaxial, como es el movimiento oscilatorio del pistón de la bomba volumétrica. Existen tres métodos más habituales para el cálculo de la resistencia a fatiga [7]:

- Cálculo de la resistencia a fatiga a partir de la tensión
- Cálculo de la resistencia a fatiga a partir de la deformación
- Fatiga por crecimiento de grieta

### 2.6.1. Cálculo de la resistencia a fatiga a partir de la tensión

Este tipo de cálculo se aplica para el estudio de la fatiga bajo un número alto de ciclos  $> 10^4$ . En estas situaciones la tensión es baja y no se aprecia deformación plástica. Se observa además, que los resultados de este tipo de fracturas por fatiga apenas dependen de la velocidad de aplicación de las cargas.

La herramienta fundamental en este tipo de análisis son los llamados diagramas de Wöhler, o diagramas S-N, diagramas cartesianos semilogarítmicos en los que se representa en el eje de ordenadas la tensión nominal S y en el de abscisas el número de ciclos N en el que se alcanza el fallo.

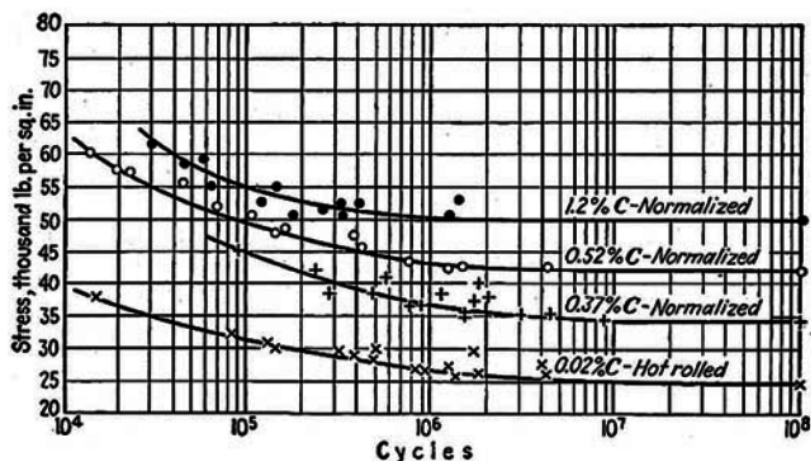


Figura 2.6: Diagrama S-N

### 2.6.2. Cálculo de la resistencia a fatiga a partir de la deformación

Este análisis se hace para la rotura en un número bajo de ciclos  $< 10^3$ , que es cuando las deformaciones plásticas son importantes. La hipótesis fundamental de este tipo de análisis es que la rotura en una situación de fatiga en bajo número de ciclos se debe a la acumulación de deformación plástica.

Para estudiar la fatiga en ciclos de este estilo se realizan ensayos con control de desplazamiento como el mostrado en la figura 2.7. Si sobre una probeta sometida a tracción se imponen desplazamientos de rango  $\Delta\epsilon$ , después de una fase transitoria en la que la deformación plástica crece, se alcanza un régimen permanente en el que se aprecia claramente una respuesta con histéresis, donde la tensión abarca un rango  $\Delta\sigma$ .

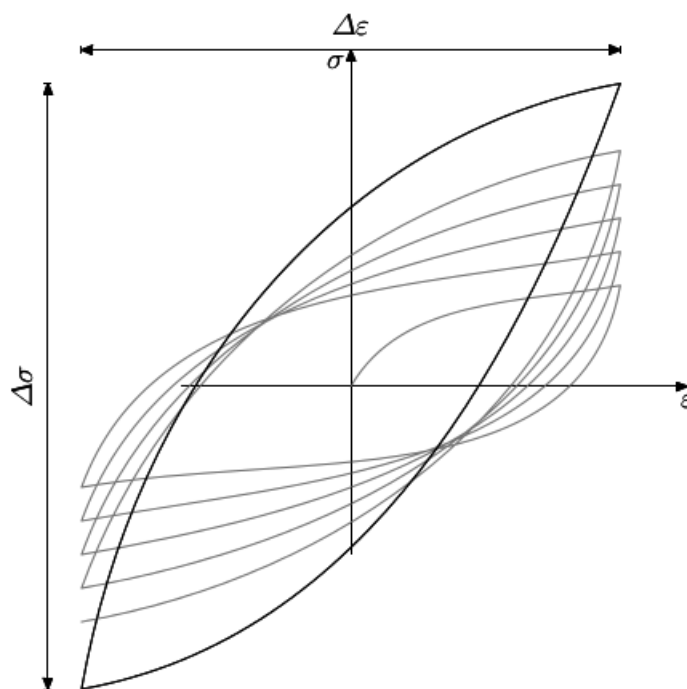


Figura 2.7: Ciclo de control de deformación de un material elastoplástico

Experimentalmente se ha observado que en estos ciclos de deformación, la amplitud de la deformación plástica  $\frac{\Delta\varepsilon^p}{2}$  está relacionada con el número de ciclos hasta el fallo  $N$  mediante una ecuación de la forma:

$$\frac{\Delta\varepsilon^p}{2} = \varepsilon'_f(2N)^c \quad (2.4)$$

Siendo  $\varepsilon'_f$  el coeficiente de ductilidad a fatiga y  $c$  el exponente de ductilidad a fatiga. La primera de estas constantes mide la deformación plástica que llevaría al fallo en medio ciclo de carga (un cambio de signo en la deformación). La segunda de estas constantes tiene un valor entre  $-0,5$  y  $-0,7$  para metales.

Teniendo en cuenta también la deformación elástica podemos escribir la ecuación de Basquin como:

$$\frac{\Delta\varepsilon}{2} = \frac{\sigma'_f}{E}(2N)^b + \varepsilon'_f(2N)^c \quad (2.5)$$

El primer sumando del término de la derecha cuantifica la contribución elástica al fallo por fatiga y el segundo sumando mide la contribución de la deformación plástica al fallo.

### 2.6.3. Fatiga por crecimiento de grieta

Se basa principalmente en que cuando una pieza o estructura está sometida a cargas cíclicas y además tiene una grieta, puede ocurrir que ésta crezca hasta alcanzar un tamaño tan grande que la pieza se parta. Es importante tener en cuenta que en las cargas que afectan a este fenómeno no se incluyen las de compresión pues estas no propician el crecimiento de grietas.



## Capítulo 3

# Golpe de ariete

### 3.1. Introducción al problema

#### 3.1.1. Definición de la geometría

Para el estudio del golpe de ariete que puede tener lugar en la bomba volumétrica es importante tener en cuenta la geometría de la misma y el ciclo de funcionamiento que sigue, por ello a continuación se muestra una representación de su geometría en la figura 3.1.

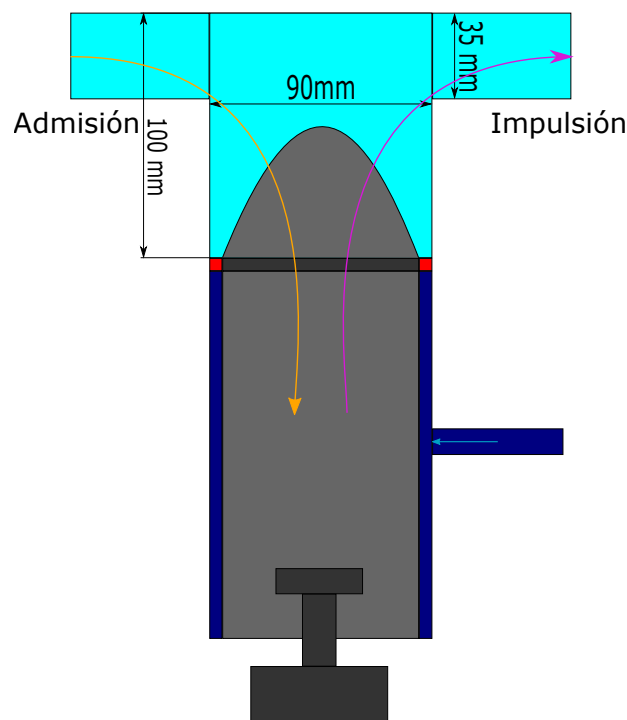


Figura 3.1: Esquema de la geometría de la bomba

Como se puede observar durante la etapa de admisión se produce el proceso de aspiración y el sentido de movimiento del pistón en esta etapa es descendente. Cuando la válvula giratoria, que no ha sido representada para una mejor visualización de la geometría, comienza a rotar, el pistón inicia su movimiento ascendente y en el momento que la válvula pone en contacto su orificio con la tubería de impulsión, el fluido sale bombeado.

Alrededor de las paredes del pistón se introduce agua para utilizarla como barrera hidráulica con una presión igual o ligeramente superior a la del fluido que se va a bombear para así evitar fugas del mismo, y elevados gradientes de presión que provoquen que sufra la membrana aislante, que separa la barrera de agua del fluido de trabajo.

Puede observarse una parte del anclaje que tiene el pistón con el brazo motor que genera el movimiento, destacando la presencia de la zona con menor sección que será la que sufra en mayor medida los efectos de las sobrepresiones que tienen lugar en el proceso.

### 3.1.2. Hipótesis sobre el problema

En el funcionamiento de la bomba volumétrica tienen lugar dos problemas relacionados con el golpe de ariete:

En primer lugar, durante la aspiración el fluido viaja a la velocidad lineal del pistón  $V_0$  y a una velocidad en la tubería que obtiene por conservación de la masa y que designaremos como  $V_T$ . Cuando el pistón llega al final de su carrera la velocidad pasa a ser nula, el fluido sufre por tanto una  $\Delta V = V_0 - 0$ , lo que genera un golpe de ariete, una sobrepresión.

Durante la impulsión del fluido, éste vuelve a moverse a la misma velocidad que el pistón cuya variación es igual que en la etapa de aspiración. Cuando el pistón termina su carrera y se detiene, transmite las nuevas condiciones de contorno al fluido ( $V = 0$ ) el cual como lleva la inercia del movimiento generado durante la impulsión, tiende a alejarse de la bomba y a arrastrar las partículas más cercanas a ella, lo que origina una depresión. Se cree que esta depresión es tan importante que provoca que el fluido en las zonas cercanas a la bomba volumétrica alcance la presión de vapor, generándose una burbuja de vapor de fluido y produciéndose la rotura de columna líquida. Las partículas que permanecen todavía en estado líquido se ven arrastradas por la fuerza que se origina a causa de la depresión, provocando que la burbuja de vapor colapse generando un nuevo golpe de ariete.

## 3.2. Estudio analítico

Para una primera aproximación del problema trataremos de integrar la ecuación diferencial que rige el desplazamiento del frente líquido en la interfase vapor-líquido tratando de obtener las tres soluciones posición, velocidad y aceleración del mismo con el motivo de averiguar en que momento colapsa la burbuja, y si cabe la posibilidad de que este momento ocurra cuando la válvula se ha abierto de nuevo hacia la impulsión pudiendo ocasionar el colapso de dicha burbuja en el interior de la bomba en lugar de la tubería, lo que provocaría un fenómeno de interacción fluido estructura con la transmisión de ondas precursoras a lo largo del pistón, afectando a la zona más débil antes descrita y provocando su rotura por fatiga a lo largo del tiempo.

La ecuación que nos permitiría obtener estos valores ha sido desarrollada en [4] y es la siguiente:

$$\ddot{x}x - \frac{f_r}{2D}(\dot{x})^2x = \frac{P_T - P}{\rho} \quad (3.1)$$

La resolución de la ecuación 3.1 no es tan sencilla de aplicar a nuestro caso en cuestión y después de analizarlo detenidamente, vemos que no disponemos de la suficiente información acerca de las condiciones de contorno para poder realizar la integración. Más concretamente no es posible evaluar la deceleración de las partículas fluidas en la zona del cierre de la válvula,

pues no conocemos la variación de la velocidad de dichas partículas. Esta velocidad depende del movimiento del pistón y de la velocidad de giro de la válvula.

Decidimos abordar por tanto la otra parte de nuestra hipótesis de la burbuja de vapor, buscando la respuesta a la pregunta de si la burbuja podría colapsar en el interior de la cámara de la bomba. Se trata de averiguar si en el supuesto momento del colapso de la burbuja, la válvula ya se ha vuelto a abrir, pudiendo colarse la burbuja al interior de la cámara. Por esta rama de estudio se decide, a partir de los datos que tenemos sobre la válvula giratoria, desarrollar un programa que determine la función que sigue el área de paso a lo largo del tiempo y del ángulo girado por la válvula, que tal y como esta diseñada la geometría corresponde con un giro de  $180^\circ$ . Con estos datos se pretende conocer el tiempo que tarda la válvula giratoria en volver a abrirse de nuevo.

### 3.2.1. Cálculo del área de paso en relación al giro de la válvula

Para conseguir obtener la variación del área de paso frente al tiempo y al ángulo haremos uso de la programación con Matlab basándonos en la siguiente suposición, el área de paso del fluido a través de la válvula giratoria se supone igual a la de la intersección de dos círculos de diámetro igual al diámetro de la salida, que a su vez es igual al de las tuberías de aspiración e impulsión como nos ha comunicado la empresa. Aunque varía un poco la curvatura si consideramos que nuestra válvula es similar a una válvula de bola, como el diámetro del cabezal de la válvula de giro es más del doble que el diámetro de salida del fluido, poco afecta por tanto en el resultado.

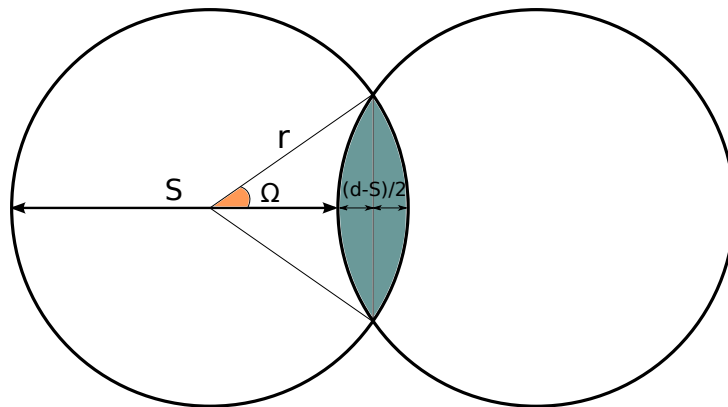


Figura 3.2: Modelo de variación del área de paso

Para calcular el área de intersección de las dos circunferencias, que será el área de paso de fluido a través de la válvula, observando la figura 3.2 podemos plantear la siguiente igualdad trigonométrica:

$$r \cos(\Omega) = r - \frac{d - S}{2} \quad (3.2)$$

Despejando el valor del ángulo  $\Omega$ .

$$\Omega = \arccos \left( 1 - \frac{d - S}{2r} \right) \quad (3.3)$$

Por lo tanto el área de paso, sombreada en la figura 3.2 se obtiene como:

$$A = 2 \left[ \pi r^2 \left( \frac{2\Omega}{2\pi} \right) - \left( r - \frac{d-S}{2} \right) r \sin(\Omega) \right] \quad (3.4)$$

Es necesario calcular tanto el valor de  $S$  para la ecuación 3.4, como el ángulo que recorre la válvula desde que está absorbiendo o expulsando fluido hasta que está cerrada completamente, denominado ángulo de apertura y cierre,  $\alpha_{io}$  para ello se recurre a la figura 3.3 y 3.4 respectivamente.

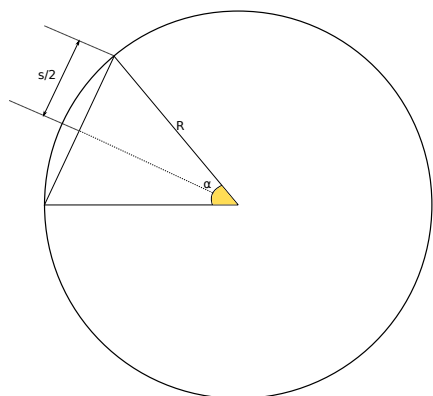


Figura 3.3: Vista simplificada en planta de la válvula giratoria

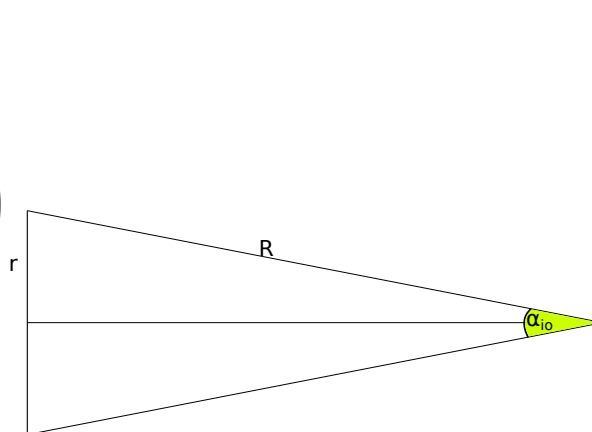


Figura 3.4: Ángulo de apertura y cierre

De estas figuras por geometría se puede obtener las siguientes relaciones:

$$\frac{S}{2} = R \sin \left( \frac{\alpha}{2} \right) \quad (3.5)$$

Siendo  $\alpha$  el valor de ángulo girado por la válvula.

$$\alpha_{io} = 2 \arcsin \left( \frac{r}{R} \right) \quad (3.6)$$

Una vez planteadas las ecuaciones, éstas permiten llevar a cabo la programación de un algoritmo que calcule el área de paso de fluido en relación al tiempo transcurrido, figura 3.5, y al ángulo de giro, figura 3.6.

### 3.2.2. Conclusión

Observando los resultados obtenidos, se demuestra que durante una buena parte del ciclo de giro de la válvula, ésta permanece cerrada. Mirando de nuevo el [vídeo](#) de la descripción del movimiento de la bomba volumétrica proporcionado por la empresa o la figura 1.1, llegamos al planteamiento de una nueva hipótesis descrita en el siguiente capítulo.

Con estas pruebas obtenidas y debido a **condiciones de contorno no definidas para poder interpretar el modelo**. La relevancia del **golpe de ariete pasa ahora a un segundo plano**, cambiando el enfoque de la investigación a la etapa durante la cual la válvula está completamente cerrada y el pistón ya ha comenzado su movimiento ascendente.



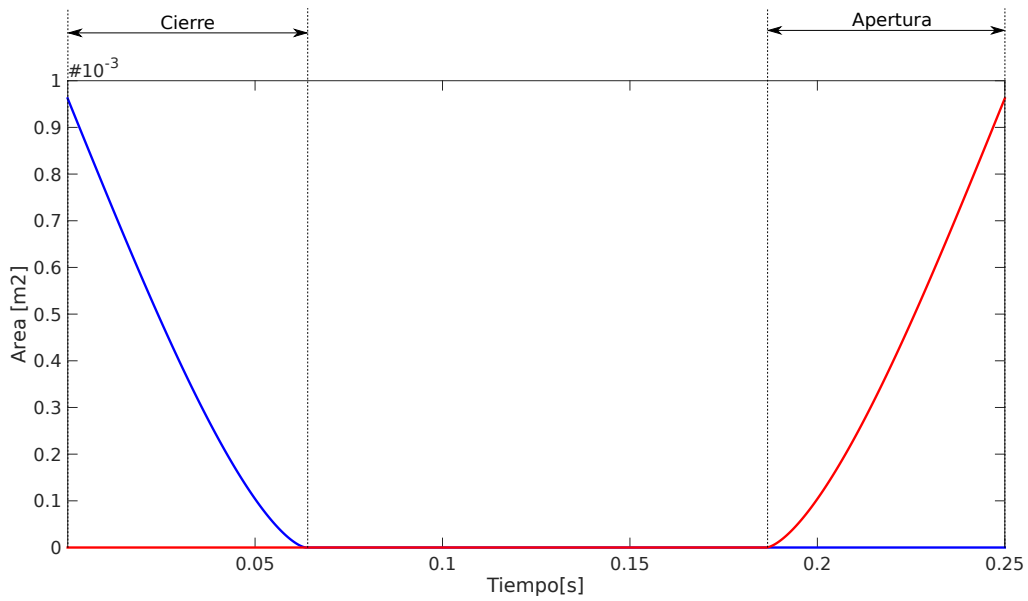


Figura 3.5: Variación del área de paso de fluido respecto al tiempo

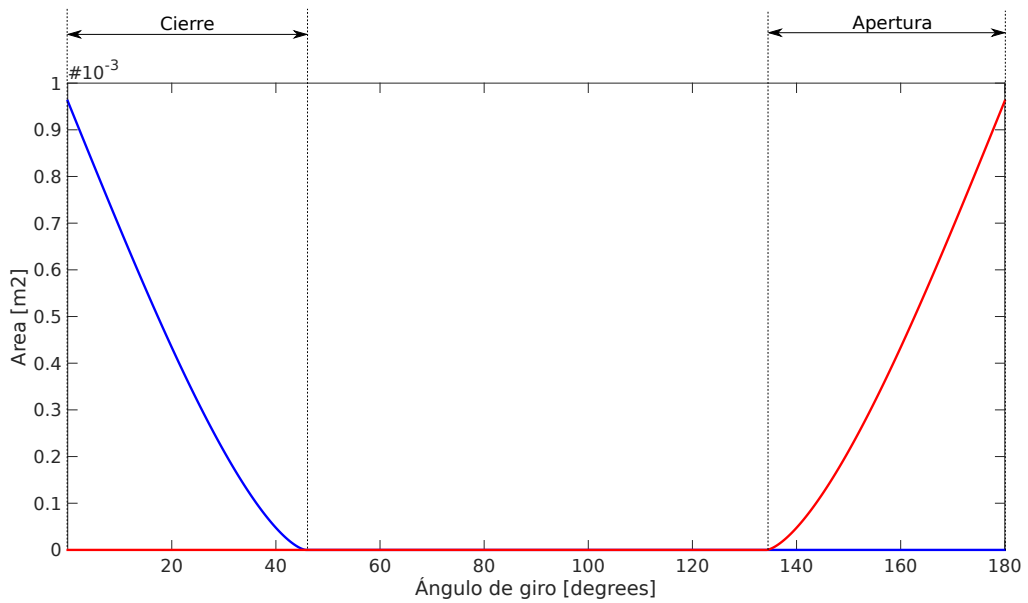


Figura 3.6: Variación del área de paso de fluido respecto al ángulo girado por la válvula



## Capítulo 4

# Sobrepresión interna

### 4.1. Planteamiento de la hipótesis

Los resultados obtenidos acerca del movimiento cíclico de la válvula permiten observar que hay una parte considerable del tiempo en el que la válvula está girando de la zona de succión a la de impulsión en la que el área de paso de fluido es nula. Durante éste tramo el movimiento del pistón está sincronizado de manera que ya está moviéndose en sentido ascendente, ejerciendo por tanto una fuerza de impulsión sobre el fluido que permanece estático, pues no puede abandonar la cámara de la bomba por ningún orificio, se trata de comprimir un fluido líquido, por lo tanto con poca compresibilidad.

Según nuestra hipótesis esto se traduce en un gran aumento de presión en la bomba durante esta etapa, lo que provoca que la cabeza del pistón tenga que soportar una enorme fuerza. Ésta se transmite a lo largo de todos los elementos mecánicos que participan en la acción de impulsión, siendo más desfavorable para las piezas con menor sección, la señalada en la figura 4.1, donde la fuerza a soportar es mayor, provocando en ellas desgaste por fatiga y terminando en rotura de la bomba.

### 4.2. Elección del modelo de compresibilidad

El proceso que sufre la bomba y que nos interesa estudiar se trata de la compresión de un líquido y por tanto debe establecerse alguna regla para la variación de su densidad con el aumento de presión al que se ve sometido. Se plantean dos funciones barotrópicas que permitan solventar esa reducción de volumen que sufre la cámara del pistón en forma de un aumento de la densidad.

La primera función que se plantea 4.1 es una relación lineal que trae incorporada en los solvers de compresibilidad de líquidos el programa de CFD OpenFOAM y se trata de un aumento lineal de la densidad en función de la diferencia de presión y con una pendiente igual al factor de compresibilidad del líquido a estudiar,  $\psi$ .

$$\rho = \rho_0 + \psi(P - P_0) \quad (4.1)$$

La segunda función que se plantea es la llamada ecuación de Tait 4.2 [4], propone una variación exponencial de la densidad en función de la diferencia de presiones y una constante  $k$ , el módulo de bulk del fluido.

$$\rho = \rho_0 e^{\frac{P-P_0}{k}} \quad (4.2)$$

Tras la realización de una comparativa con diferentes valores de presión, se obtienen los siguientes resultados que se muestran en la figura 4.2 de la cual podemos sacar como conclusión que la diferencia en el cálculo de la densidad por una función u otra es mínima. Se decide escoger la ecuación de Tait para modelar la compresión puesto que, aunque mínimamente, absorbe mejor los incrementos de presión aumentando más la densidad, lo que dará ligeramente un mejor resultado numérico.

Para conseguir una mejor comprensión y obtener un cálculo numérico que respalde la hipótesis planteada, en el próximo capítulo se recurrirá a la simulación del caso y de otros similares que ayuden a comprender la física del problema y el funcionamiento del software de simulación CFD.

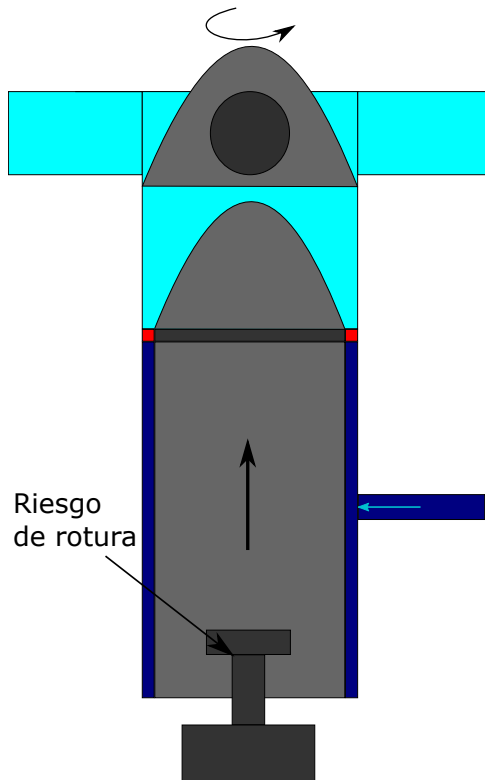


Figura 4.1: Etapa de compresión estática del fluido

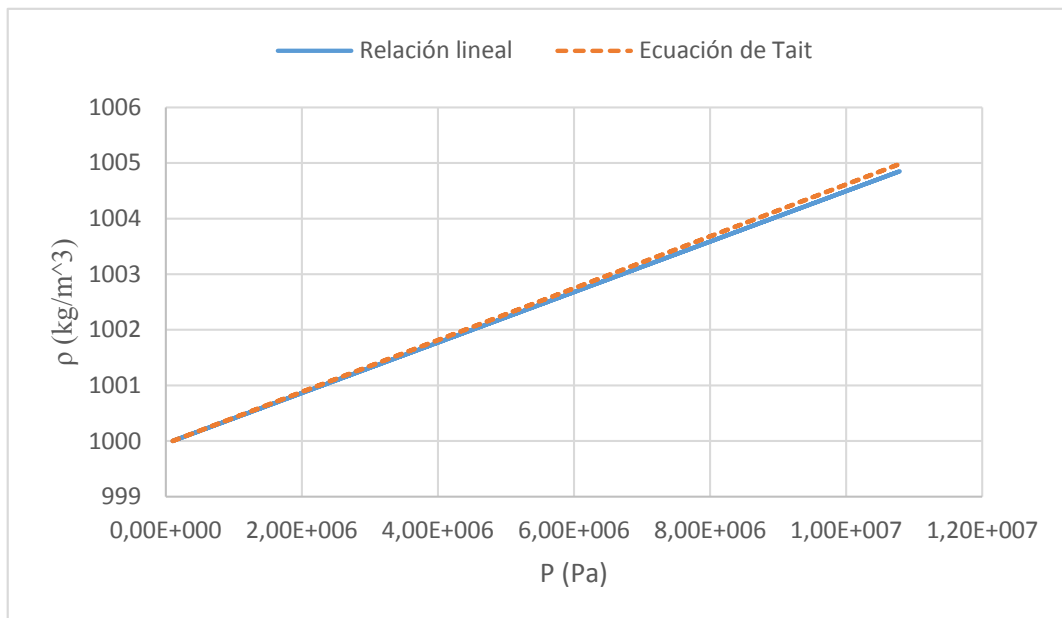


Figura 4.2: Comparativa de las funciones barotrópicas



## Capítulo 5

# Simulación numérica computacional

### 5.1. Estudio previo

Con el objetivo de adquirir un mayor conocimiento acerca de software de simulación en Dinámica de Fluidos Computacional **CFD**, se ha realizado un estudio previo con un caso similar al caso real que se trata en este informe tratando de evaluar el comportamiento de un solver de malla dinámica y las distintas condiciones de contorno impuestas. Para ello usaremos el software libre OpenFOAM que además de ser libre, da la posibilidad de acceder a los códigos fuente de los distintos solvers y utilidades pudiendo modificarlo para adaptar el comportamiento de los mismos a los casos particulares que se quieran simular. En todos los casos que se estudian se realiza una simulación de tipo *Reynolds Averaged Navier Stokes* **RANS**, y con fluido laminar, no se aplicarán modelos de turbulencia.

#### 5.1.1. Geometría

La geometría utilizada se trata de un pistón cilíndrico de cabeza plana, el cual presenta una salida de fluido en su eje central, que se representa con un alargamiento de la pared como si fuese una tubería.

La figura 5.1 muestra la geometría descrita, es importante tener en cuenta que la simulación que se va a llevar a cabo es bidimensional, por lo tanto se requiere del uso de una utilidad que genere la axisimetría necesaria para que se corresponda con el pistón cilíndrico comentado. Para ello debe establecerse previamente una geometría bidimensional y denominar las caras de la pared que van a formar parte del eje todas con un mismo nombre y seguir el mismo proceso para las caras frontal y trasera que conforman la bidimensionalidad de la geometría. Una vez hecho esto, se ejecuta la utilidad pasándole dichos nombres y genera el colapso de las caras frontal y trasera en la zona de las caras donde se ha dicho que iría el eje de simetría.

En cuanto a la malla a utilizar se trata de una malla estructurada sin ningún tipo de *grading*, el tamaño de las celdas no va aumentando a medida que se alejan de un cierto punto. El número de celdas será de 12000, se ha tomado este número de celdas para conseguir una buena velocidad de cálculo y poder evaluar con mayor rapidez la efectividad del software. Al tratarse de un estudio previo no se ha tenido en cuenta el estudio de sensibilidad de malla que si se realizará para el caso real de la bomba volumétrica.

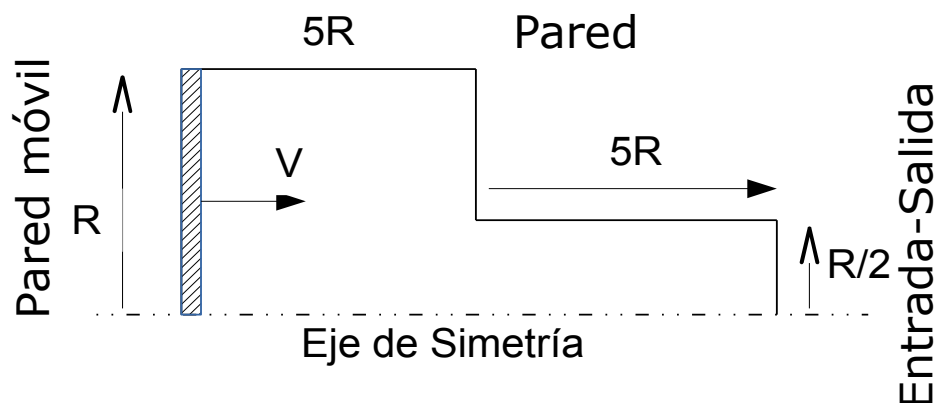


Figura 5.1: Geometría del estudio previo  $R = 15mm$

## 5.1.2. Simulación

### 5.1.2.1. Implementación de la malla dinámica

El movimiento a simular implica la compresión de las celdas de la malla cuando el pistón está moviéndose en el sentido de expulsar fluido, la pared que se va a mover será la pared vertical de la izquierda de la geometría, que se correspondería en el caso real con la cabeza del pistón.

Existen dos posibles soluciones a este problema de compresión de la malla:

1. Utilizar un tipo de malla dinámica en el que a medida que avanza el pistón se van comprimiendo las celdas, reduciendo su tamaño, esto provocaría que al final de la carrera hubiese una excesiva acumulación de celdas minúsculas en un espacio muy pequeño, y el solver tendría problemas de estabilidad debido al ínfimo tamaño de celda.
2. Emplear una malla dinámica que haga desaparecer y aparecer celdas cuando su tamaño esté por debajo o por encima de unos ciertos límites establecidos. De esta manera en el final de la carrera no habría esa acumulación de celdas, simplemente habría menor número de ellas.

Se ha seleccionado por tanto la segunda solución, el hacer desaparecer y aparecer las celdas con el movimiento de la cabeza del pistón, pero resulta que en las utilidades de malla dinámica de OpenFOAM sólo existe una que haga este proceso, denominada *movingConeTopoFvMesh*, en ella se distinguen tres zonas en la malla las zonas -1, 0, 1:

- 1:** Zona hacia la que se mueve la geometría en cuestión y en la que desaparecen las celdas cuando su tamaño es inferior a uno establecido.
- 0:** Zona de no variabilidad de la malla.
- 1:** Zona de aparición de las celdas a medida que se desplaza la geometría.

Las diferentes zonas se ilustran en la figura 5.2 que muestra un tutorial de OpenFOAM con esta utilidad.



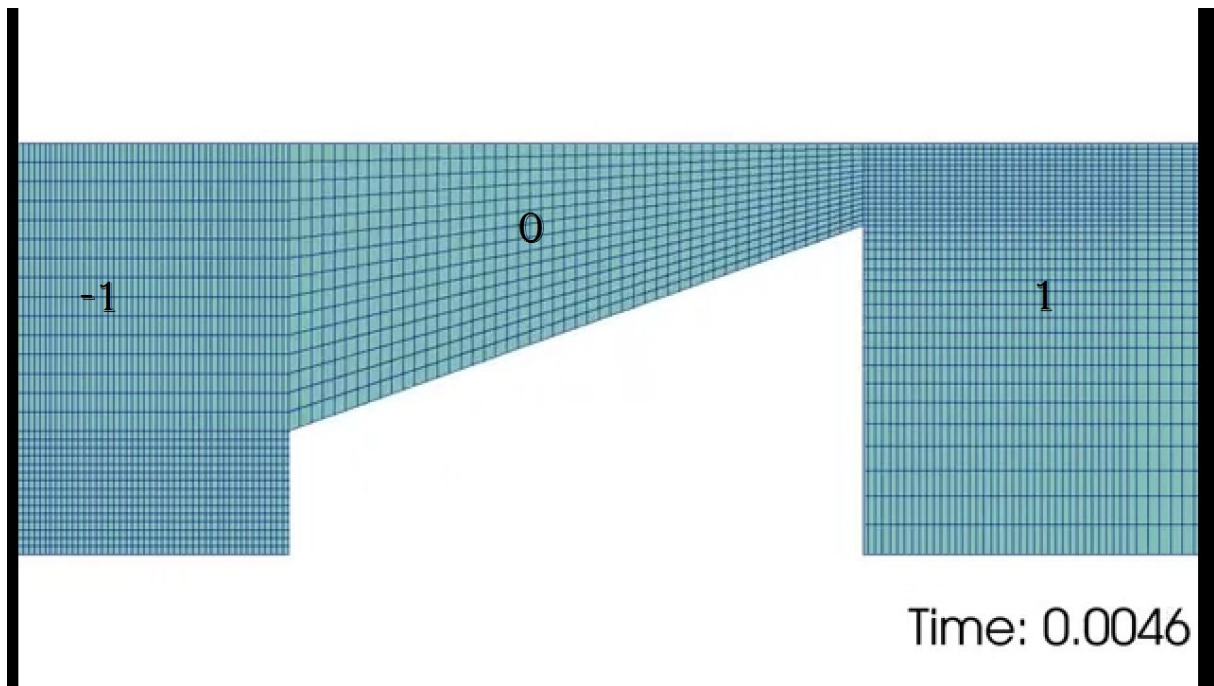


Figura 5.2: Tutorial de la utilidad *movingConeTopoFvMesh* de *OpenFOAM*

En nuestro caso no existen ni la zona 0 ni la zona  $-1$ , sólo nos interesa que desaparezcan las celdas de la zona 1, que comprenderá toda la geometría, cuando el pistón esté en la etapa de impulsión y que aparezcan de nuevo en esta misma zona a medida que el pistón avanza durante la etapa de absorción de fluido. Para ello se ha recurrido a modificar ésto en el código fuente de la utilidad, denominando a la nueva utilidad *movingConeTopoFvMesh\_v2*.

Las principales modificaciones consisten en comentar la parte del código relativa a la constante *curRight*, con ésto se evita que se extrusionen nuevas celdas en el espacio recorrido por la pared móvil. En el apéndice A se adjunta el código fuente de la utilidad y se pueden apreciar las partes comentadas en el mismo.

Una vez adaptada la utilidad al caso sobre el que se va a trabajar, es necesario añadir una serie de diccionarios que establezcan ciertos parámetros imprescindibles para que ésta funcione:

**topoSetDict:** En él se define un región, denominada *box*, que comprende la pared que realizara el movimiento y sobre la que se producirá el cambio de tamaño de las celdas. Además es preciso definir la distancia que ocupa la primera hilera de celdas que van a desaparecer, su anchura en la dirección del movimiento.

**meshModifiers:** Aquí se delimita la anchura de celda mínima y máxima para que aparezcan y desaparezcan. La anchura mínima se ha considerado como la cuarta parte de la anchura real de celda y la anchura máxima como la anchura real más la cuarta parte.

**dynamicMeshDict:** En este diccionario se establecen los parámetros de movimiento de la pared, el solver está diseñado para trabajar con un movimiento de pared sinusoidal definido por la fórmula 5.1<sup>1</sup>, para establecer este movimiento los parámetros que se piden son la carrera del pistón en velocidad, es decir multiplicada por la velocidad angular, y el semi-periodo del movimiento.

<sup>1</sup> $A = \frac{\text{Carrera}}{2}$ , Amplitud del movimiento sinusoidal

$$x = A + A\cos(\omega t) \quad (5.1)$$

### 5.1.2.2. Resolución numérica

Una vez configurada la utilidad que simulará el movimiento del pistón, es necesario seleccionar el tipo de solver que se utilizará para llevar a cabo los cálculos, en éste caso se dispone de una entrada y salida de flujo por un mismo orificio y se bombea un fluido líquido, para el estudio preliminar se trata de agua.

Criterios que debe cumplir el solver para éste caso:

- Solver de tipo incompresible
- Solver de malla dinámica que tenga en cuenta el movimiento del pistón

El solver que mejor cumple estos criterios y que ya viene por defecto en OpenFOAM se trata del *pimpleDyMFoam*.

### 5.1.2.3. Esquemas numéricos aplicados

Para cualquier simulación es necesario tener en cuenta los esquemas de discretización de los términos de las ecuaciones de Navier-Stokes utilizados por OpenFOAM. A continuación se hace un breve resumen sobre los diferentes esquemas posibles:

En primer lugar se tratarán los esquemas de discretización espacial, los términos a discretizar son los siguientes:

**Términos convectivos (Divergencia):** Para éste término los esquemas de primer orden proporcionan una solución muy robusta en términos de estabilidad y convergencia pero con poca precisión, los esquemas de segundo orden tienen mayor precisión matemática y ofrecen resultados más ajustados a lo esperado, con la desventaja de una convergencia más lenta. Para éste caso se ha recurrido a un esquema de segundo orden denominado en OpenFOAM como *linearUpwind*.

**Términos difusivos (Laplaciano):** Lógicamente como la definición del laplaciano es una derivada de segundo orden, OpenFOAM sólo dispone de este tipo de esquemas para el laplaciano. Estos términos presentan sensibilidad ante parámetros de la calidad de la malla como son la ortogonalidad de las celdas, medido por el ángulo que forma la cara de la celda con línea que une sus centros. Idealmente éste ángulo debe estar comprendido entre  $0 - 40^\circ$ . En la geometría de éste caso la no ortogonalidad es mínima con un valor medio de  $5,32e^5$ , el esquema usado es un *Gauss linear corrected* aunque no sería necesario efectuar la corrección.

**Términos de gradiente:** Estos términos también están influenciados por la calidad de la malla en varios aspectos, como son la ortogonalidad de las celdas y, dado que los gradientes son vectores, la orientación de los mismos con las caras de las celdas. Para casos generales sin demasiada complejidad y turbulencia, como es el nuestro, el esquema estándar denominado *Gauss Linear* suele funcionar adecuadamente.

En lo relativo a la discretización temporal existen de nuevo esquemas de primero y segundo orden, los primeros son más estables y proporcionan soluciones que convergen más rápido en cada paso de tiempo a costa de una menor precisión para recoger las variaciones temporales de las variables fluidas, algo de importancia en análisis de frecuencias, desprendimiento de vórtices o frecuencia de vaporización y condensación. Para nuestro caso es suficiente con utilizar un esquema temporal de primer orden como es *Euler*.

También es de gran importancia a la hora de configurar un caso establecer un paso de tiempo que cumpla con 3 criterios básicos:

- Obtención de los fenómenos fluidos que sean considerados relevantes en términos de frecuencia de muestreo.
- Cumplimiento de criterios numéricos que hagan la solución estable numéricamente.
- Tiempo total de cálculo viable a efectos de aplicación industrial

El paso de tiempo para este caso se tomará autoajutable mediante el número de Courant,  $CFL = \frac{v\Delta t}{\Delta x}$ , que autoajusta el  $\Delta t$  en función de la velocidad,  $v$ , y el ancho de celda  $\Delta x$  a un valor prefijado, lo ideal es un numero de Courant máximo en general  $CFL \leq 1$  para asegurar estabilidad numérica. El numero de Courant máximo establece el número de veces que una partícula fluida pasa por la celda más pequeña de todo el dominio discretizado. Por tanto un  $CFL > 1$  implicaría que hay celdas del dominio en las que no se han calculado las variables fluidas.

### 5.1.3. Estudio comparativo de las condiciones de contorno

Se ha realizado la configuración de diferentes casos que presentan la misma geometría, malla y parámetros de resolución, diferenciándose únicamente en la condición de contorno que modela la entrada y salida de fluido, por tanto se tratará de evaluar cual de las condiciones de contorno descritas a continuación presenta un mejor comportamiento físico y adaptación a la teoría fluidodinámica.

Para ello es necesario conocer de que tipo de condiciones de contorno estamos hablando, existen dos tipos básicos de condiciones de contorno:

**Dirichlet:** Establece como condición de contorno un valor fijo, por tanto el valor del campo de la variable a integrar tomará como valor fijo en los puntos de contorno el aplicado por este tipo de condición.

**Neumann:** Establece como condición de contorno un valor que se obtiene a partir del campo de una variable, habitualmente una derivada del mismo.

Para la condición de entrada y salida de fluido en este estudio previo se ha utilizado una condición de Dirichlet para la presión, estableciendo como valor de referencia 0 la presión que hay a la salida del pistón mediante una condición *fixedValue*, se trabaja por tanto con presiones relativas y cinemáticas al tratarse de un solver incompresible. Para la condición de velocidad es donde entra en juego el estudio de las diferentes opciones que existen en OpenFOAM:

**inletOutlet:** Este tipo de condición combina la tipo Neumann para el caso de la salida del fluido, pues aplica la condición *zeroGradient* que impone gradiente de velocidad cero en la salida, para la entrada aplica una condición de tipo Dirichlet pues impone una condición *fixedValue* lo que provoca que la entrada del fluido sea constante con una cierta velocidad fija marcada por el usuario previamente en la configuración.

**pressureInletVelocity:** Esta condición de contorno emplea el mismo sistema tanto para la entrada como la salida de fluido, a partir de los valores del campo de presión en ese punto se deriva un valor de velocidad con dirección normal a las caras de las celdas que delimitan el contorno.

**pressureInletOutletVelocity:** Es similar a la condición anterior con la diferencia de que en el caso de la salida de fluido se emplea, en lugar de la obtención del valor de velocidad a partir del campo de presión, una condición del tipo *zeroGradient*, por tanto aplica gradiente de velocidad nulo a la salida.

La conclusión sacada sobre el estudio de estas condiciones de contorno es que la que mejor se adapta al caso en cuestión es la *pressureInletOutletVelocity* dado que la condición de *inletOutlet* presenta un problema de conservación de la masa, puesto que al haber impuesto un valor fijo de velocidad cuando el pistón está aspirando fluido, está entrando un caudal de fluido que no corresponde con la cantidad que debería entrar dado que el pistón presenta una velocidad de tipo sinusoidal y esto se refleja en la velocidad de entrada de fluido que no es realmente constante, en la simulación llevada a cabo con esta condición de contorno se produce un error y parada de la simulación cuando se trata de hacer la segunda impulsión de fluido, debido al problema comentado el solver no es capaz de amortiguar éste error. La condición de *pressureInletVelocity* tiene como inconveniente que durante la salida de fluido está tratando de calcular la velocidad a partir de la presión de celdas que en verdad no existen pues están fuera del dominio del pistón, no se establece la presión que hay a la salida después de las celdas límites de contorno. La condición de contorno escogida soluciona este problema planteando para la salida de fluido la imposición de un gradiente de velocidad nulo.

Como se puede observar en la figura 5.3b la zona de mayor velocidad del fluido se da a lo largo de todo el eje central del pistón, es decir de su eje de simetría, algo que no tiene ningún significado físico y por tanto la utilización de esta condición de contorno para modelar un proceso de entrada y salida de fluido queda descartada. Analizando la figura 5.3a todo parece comportarse de acuerdo a las leyes físicas que rigen el comportamiento de los fluidos apareciendo las mayores velocidades en las zonas de torbellinos debido a la turbulencia.

## 5.2. Estudio del caso real

Para la bomba volumétrica que se está estudiando trataremos de simular la presión estática que tiene lugar en el interior de la cámara del pistón durante el proceso de compresión pura. Éste transcurre en el tramo de tiempo en el que la válvula está cerrada y la impulsión ya está teniendo lugar, esos 0,123 segundos, que se deducen de la figura 3.5. Se simulará mediante OpenFOAM con una simulación de tipo RANS y laminar pues no hay apenas movimiento de fluido.

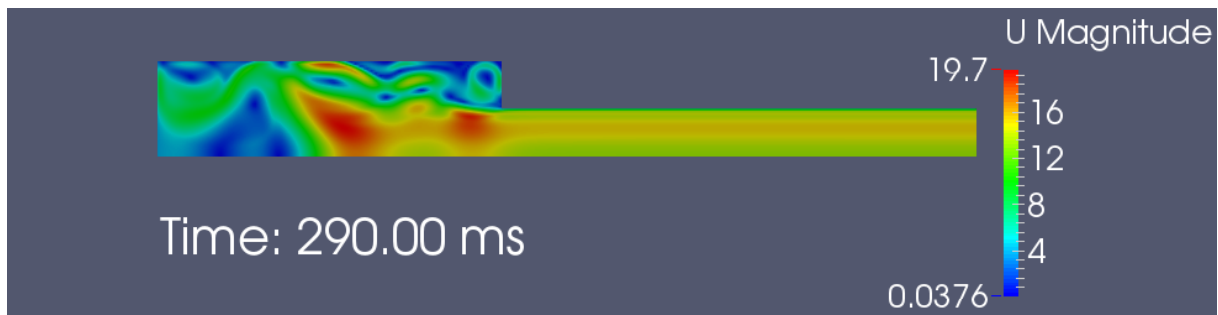
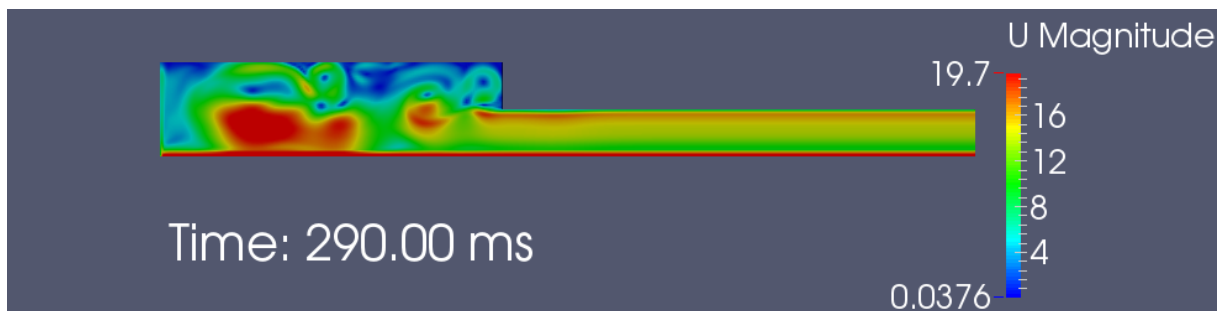
(a) *pressureInletOutletVelocity*(b) *pressureInletVelocity*

Figura 5.3: Estudio de las condiciones de contorno

### 5.2.1. Geometría

La geometría empleada en la simulación consta de un pistón cilíndrico de cabeza plana, sin ningún orificio para la salida de fluido, simplemente se comprime un fluido confinado inicialmente con presión atmosférica.

Al igual que en el caso del estudio previo se procede de manera similar para la formación de la axisimetría a partir del diseño bidimensional de la figura.

La malla utilizada se trata de una malla estructurada hexahédrica sin ningún tipo de *grading*, y el número de celdas que componen la malla es 800. Más adelante se realizará un estudio de sensibilidad de la malla duplicando el número de celdas para la malla media y cuadruplicándolo para la malla fina y se demostrará que por tratarse de un caso estático el número de celdas no interviene en los resultados.

### 5.2.2. Simulación

#### 5.2.2.1. Implementación del solver

En el caso en estudio se simula una compresión puramente estática, no presenta ninguna salida de fluido, por lo que su velocidad es prácticamente nula, solo presentará cierto movimiento en las cercanías a la cabeza del pistón que se va a mover.

Por tanto dado que lo que se está haciendo es una compresión de líquido al reducirse el volumen de la cámara del pistón a causa de su movimiento, es necesario utilizar un tipo de solver que presente las siguientes características:

- Compresibilidad de líquidos
- Malla dinámica
- Representación del aumento de densidad mediante la función de Tait, ecuación 4.2.

Para disponer de un solver que cumpla todo lo indicado se parte de un solver propio de OpenFOAM, *sonicLiquidFoam* que tiene implementada la compresibilidad de fluidos mediante la ecuación 4.1. Modificando el código fuente de éste se añade el soporte de malla dinámica que permitirá trabajar con un movimiento de pared, la cabeza del pistón, y se sustituye la función del cálculo de la variación de densidad por la ecuación 4.2. El nuevo solver adaptado para la simulación de la bomba volumétrica recibe el nombre de *sonicLiquidDyMFoam\_v2*.

### 5.2.2.2. Implementación de las propiedades físicas del fluido de trabajo

Para la simulación del bombeo de fluido, de entre los distintos alimentos con los que trabaja en la empresa este modelo de bomba, se ha seleccionado la mayonesa como fluido de trabajo.

Tras haber hecho un estudio de la asimilación de mayonesa como fluido no newtoniano, al final dado la complejidad de la obtención de esas propiedades se ha decidido no tener en cuenta para el caso de compresión estática que se va a simular, puesto que no hay esfuerzos tangenciales al tratarse de una compresión pura, no hay capas de fluido que deslicen entre si. Por tanto se simulará el fluido como si fuese newtoniano pero con las siguientes propiedades obtenidas en la bibliografía sobre la mayonesa tradicional.

- Densidad  $\rho = 910\text{kg}/\text{m}^3$ . [1]
- Viscosidad dinámica  $\mu = 2,5\text{Pa}\cdot\text{s}$ . [2]
- Velocidad del sonido del aceite de girasol  $a = 1471,6\text{m}/\text{s}$ . [5]

Se ha recurrido a utilizar la velocidad del sonido del aceite de girasol por similitud con la mayonesa dado que como mínimo toda mayonesa lleva un 75 % de este líquido en su composición y la obtención de esta propiedad en la mayonesa es complicada de obtener debido a la presencia de gases disueltos en ella, principalmente burbujas de aire cuya presencia interfiere en la medida de esta propiedad, como también al contenido en aceite y el tamaño de las burbujas.

La presencia de estos gases disueltos en la mayonesa provocan que durante el ciclo de bombeo en la tapa de succión aumenten su volumen por la depresión generada y cuando se produce la compresión pura estas burbujas se comprimen mucho más que la mayonesa lo que contribuye a amortiguar la sobrepresión producida por el empuje del pistón, además de la elasticidad del material de la bomba que tenderá a dilatarse y reducir también la presión máxima alcanzada.

Debido a problemas técnicos de implementar todas estas variables simultáneamente en la simulación, no se tendrá en cuenta la multifase, gases disueltos, ni la elasticidad del material, por lo que el valor de presión máximo obtenido será algo superior al valor real de presión máxima que debe soportar en cada ciclo la bomba volumétrica.

### 5.2.2.3. Esquemas numéricos

Al igual que en el estudio previo, es importante configurar el caso con los esquemas de discretización apropiados, los que se han utilizado para la simulación del caso real son los mismos que

en el estudio previo, ha excepción del termino convectivo que no va a afectar a los resultados por tratarse de un fluido estático, por tanto se deja el esquema por defecto *Gauss limitedLinear*.

### 5.2.3. Verificación de los resultados simulados con los calculados analíticamente

Para verificar la implementación del solver y corroborar que los resultados obtenidos son los teóricamente esperados se ha procedido a realizar una comparación entre los resultados de presión máxima obtenidos a diferentes intervalos de tiempo mediante la simulación y se han comparado con los que se obtendrían analíticamente con el empleo de las ecuaciones que rigen el comportamiento del pistón durante la etapa simulada.

Se procederá de la siguiente manera:

1. Se calcula para cada intervalo de tiempo la carrera que todavía le queda por recorrer al pistón, para ello es necesario el cálculo del desplazamiento llevado a cabo en ese intervalo de tiempo mediante la ecuación 5.1.
2. Se obtiene el volumen del fluido confinado para ese intervalo de tiempo, por simple geometría.
3. Mediante la ecuación de Tait 4.2 sustituyendo el término de  $\rho = \frac{M}{V}$  y despejando la  $P$  podemos obtener su valor analíticamente para cada intervalo de tiempo.

En la figura 5.4 se puede observar la comparativa de las presiones, viendo que a medida que transcurre la simulación, el error, la separación entre las rectas, es mayor, de todos modos para el período de los 0,123 segundos que se simula, la etapa de compresión, es asumible dando resultados bastante buenos, alcanzándose un orden de magnitud de 1000 bar de presión, lo que es un valor altísimo que inevitablemente provocará la rotura de la bomba volumétrica por fatiga aun teniendo en cuenta las posibles atenuaciones de la presión máxima antes mencionadas.

### 5.2.4. Estudio de la sensibilidad de malla

Se ha realizado un estudio de sensibilidad de malla partiendo, como se ha comentado en la sección 5.2.1, de una malla de 800 celdas estructurada hexahédrica sin *grading* alguno y con despreciable no ortogonalidad, para estudiar la independencia de los resultados con el número de celdas de la malla se han realizado dos simulaciones a mayores de ésta, en las que se ha duplicado y cuadruplicado el número total de celdas que la componen, manteniendo los demás criterios que definen la malla.

En la figura 5.5 se demuestra que los valores de presión máxima obtenidos en el último paso de tiempo de la simulación son idénticos en los tres tipos de malla utilizados.

Al tratarse de un aumento de la presión estática, ésta se reparte uniformemente en todo el volumen fluido, la razón de que haya una escala de colores es que hay pequeños errores en la solución numérica en las celdas que aparecen de colores rojizos, debido probablemente a que en ese punto hay un movimiento de la pared, de la cabeza del pistón. Observando la escala se aprecia que estos errores son mínimos como cabría esperar y la presión máxima toma los mismos valores uniformes en toda la región mallada, independientemente del número de celdas en que esté dividida la misma. Es lógico utilizar por tanto como simulación válida para obtener

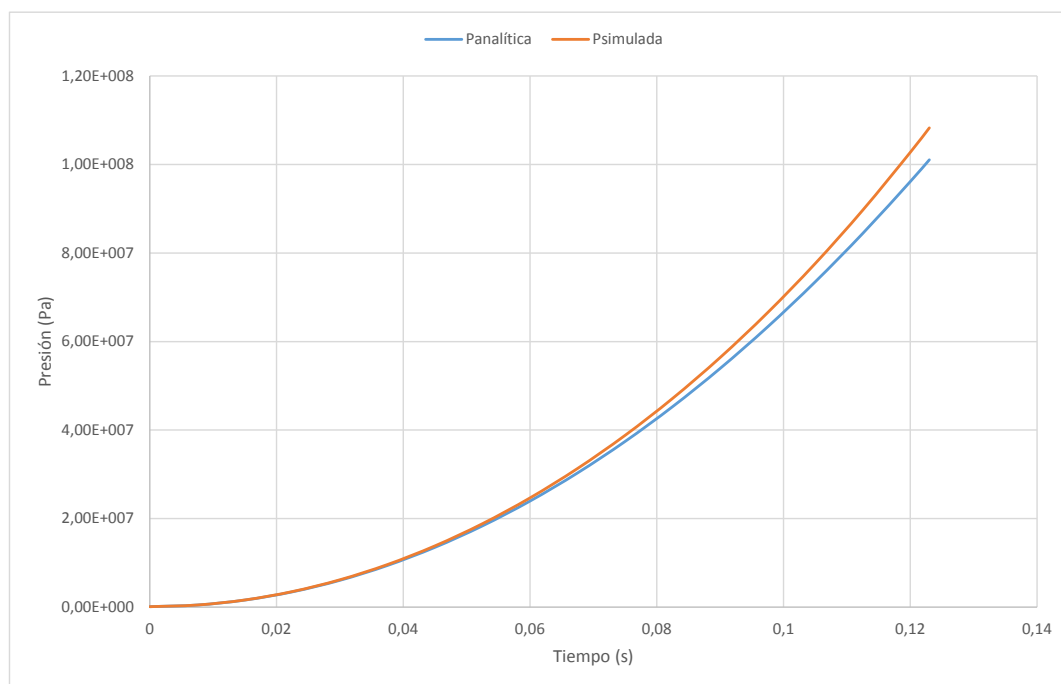


Figura 5.4: Comparativa de los resultados analíticos y simulados

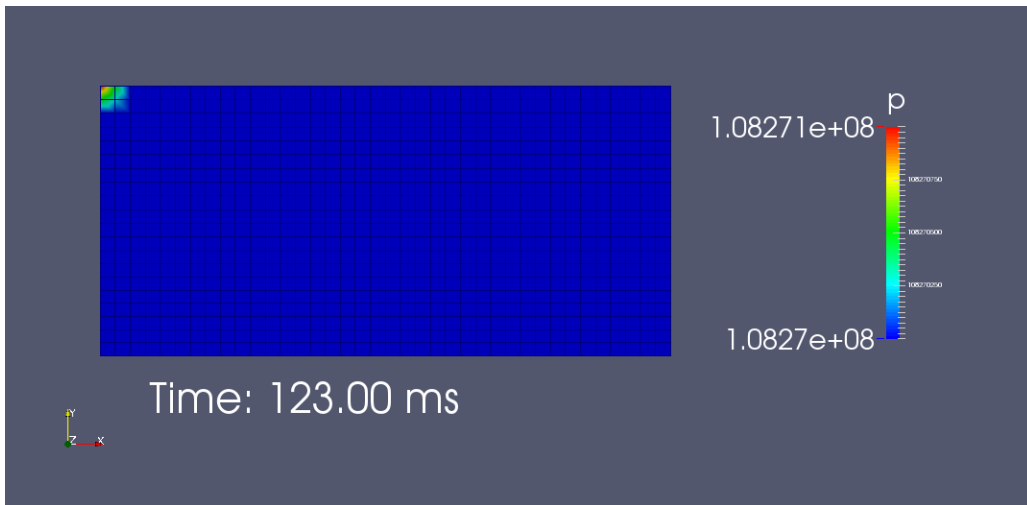
las conclusiones, aquella con menor número de celdas y por tanto menor coste computacional y mayor simplicidad.

### 5.2.5. Conclusión del estudio

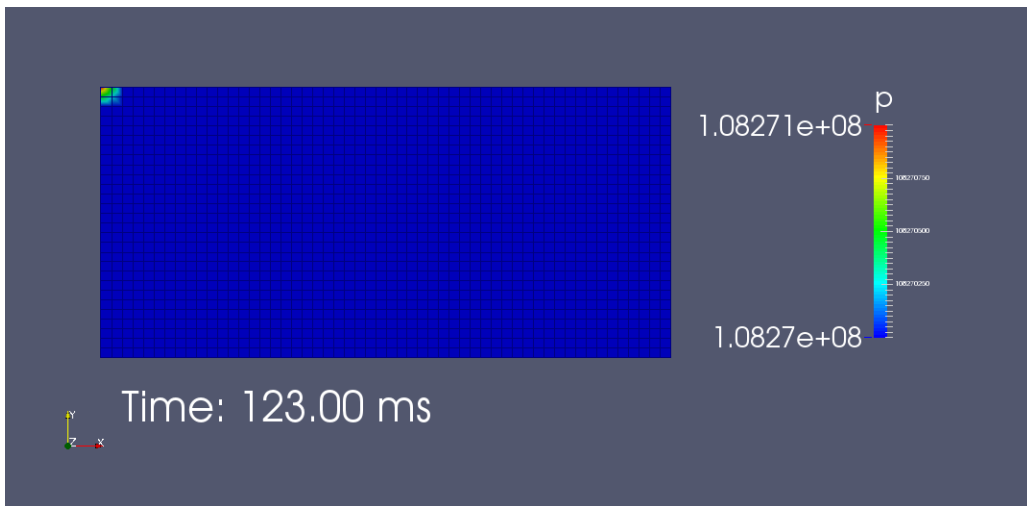
Tras haber analizado el proceso más crítico que sufre la bomba en cada uno de sus ciclos de trabajo, hemos podido observar que se generan fortísimos valores de presión en la cámara interna, que se traducen en que la cabeza del pistón tiene que soportar fuerzas tremendas que se transmiten a todo el mecanismo de impulsión de la bomba. La figura 5.6 muestra una gráfica de las fuerzas a las que se ve sometida la cabeza del pistón a lo largo del proceso de compresión estática del fluido.

Aun teniendo en cuenta los efectos de amortiguación sobre la presión, que se traducen en una fuerza sobre el pistón menor, vistos en la sección 5.2.2.2, la fuerza que ha de soportar el pistón en cada ciclo es del orden de  $10^5$  como se puede extraer de la figura 5.6, lo que provoca grandes fluctuaciones en las tensiones a las que se ve sometido el material de la bomba volumétrica, desembocando inevitablemente tarde o temprano en una rotura por fatiga en la zona del mecanismo de impulsión que presenta menor sección y por tanto mayores fuerzas a soportar.

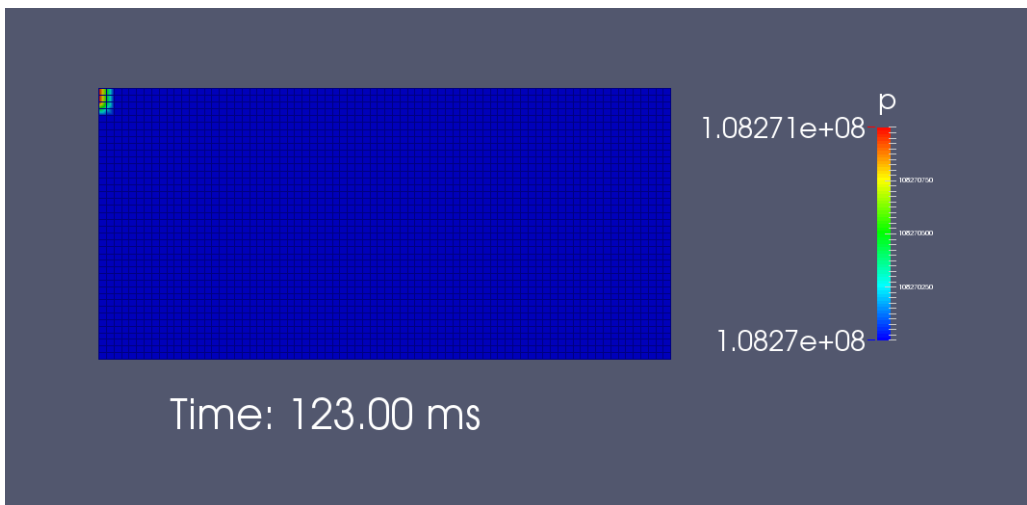




(a) *Malla gruesa*



(b) *Malla media*



(c) *Malla fina*

Figura 5.5: Estudio de la sensibilidad de malla

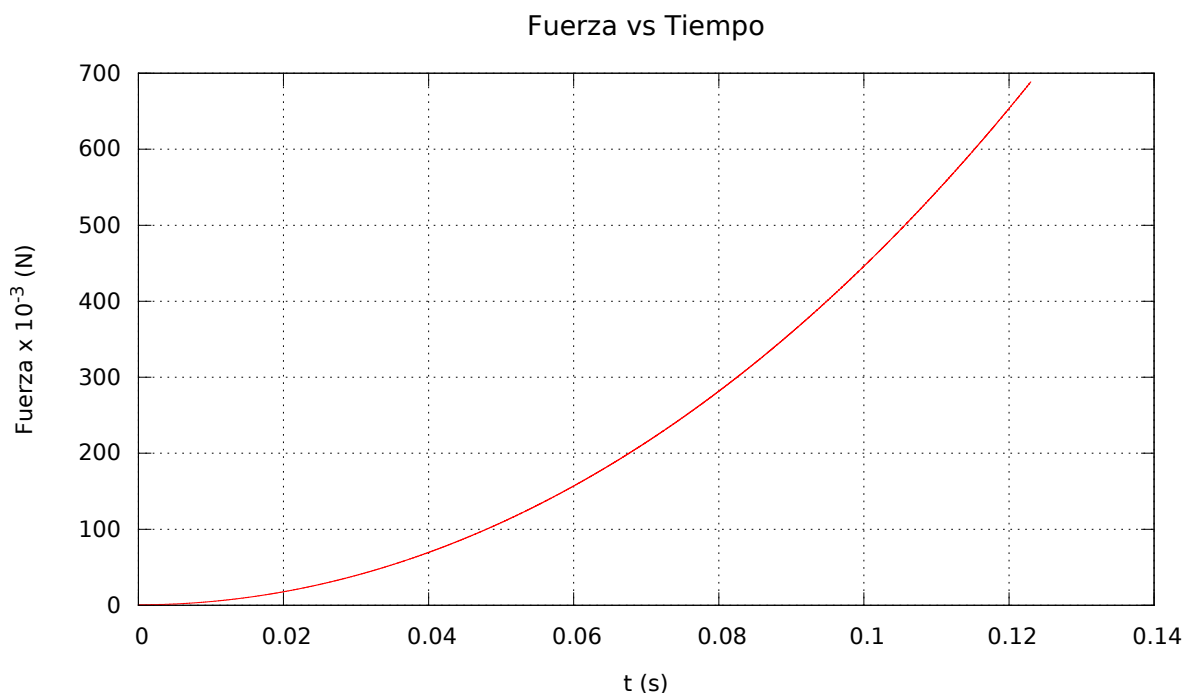


Figura 5.6: Fuerzas sobre la cabeza del pistón

### 5.3. Simulación con simetría 3D

Adicionalmente a los estudios realizados anteriormente se ha decidido hacer una simulación de las bombas volumétricas analizadas, pero esta vez empleando una geometría más próxima a la real. La geometría utilizada es un semicilindro al que se ha tratado de representar en su extremo el volumen correspondiente con la válvula cónica giratoria. La cabeza del pistón se mantiene simplificada como una superficie plana por la dificultad que conlleva aplicar la aparición y desaparición de celdas, el dinamismo de malla, a una superficie compleja como sería la cabeza real del cilindro.

Por tanto la simulación en este caso presenta las siguientes diferencias respecto a la de la sección 5.2:

- Geometría 3D simétrica respecto a un plano central, con la adición del volumen correspondiente a la válvula giratoria.
- El mallado se ha realizado en el grupo integrado de ingeniería con el mallador ICEM, por la limitación del *blockMesh* de OpenFOAM para geometrías más complejas. Creando una malla de 34145 celdas hexahédricas, con una ortogonalidad media de 16,8125.
- El volumen fluido total simulado es mayor, por la adición del correspondiente a la válvula giratoria.

El proceso de simulación es similar, utilizando el mismo solver y esquemas numéricos, pero en este caso se ha optado por correr la simulación en 4 procesadores en paralelo para acelerar el proceso por la nueva complejidad del estudio.

La principal diferencia que se aprecia es una notable divergencia entre el valor de presión máxima obtenido en el caso 3D con respecto al 2D en el último paso de tiempo, como se puede observar en la figura 5.7, con un error en torno al 23 %. Ésto se debe principalmente al aumento de volumen realizado en la simulación 3D al añadir el correspondiente a la válvula, pues en la fórmula utilizada para simular la variación de la densidad con la compresión del fluido 4.2 depende de los valores de volumen inicial y volumen en cada paso de tiempo, incluidos dentro de la densidad. Para comprobar que el fallo se debe a esto y no a otro tipo de error, se vuelve a realizar la comprobación de la presión calculada analíticamente con los valores calculados en la simulación, como se hizo en la sección 5.2.3.

Como se aprecia en la figura 5.8 la presión máxima llega en torno a los 800 bar, una presión que sigue siendo enorme para que la pueda resistir sin problemas un equipo mecánico. Se sigue corroborando con esta nueva aproximación de la solución, el fenómeno que sospechábamos, la bomba volumétrica no es capaz de resistir estas elevadísimas fluctuaciones de presión y acaba en una rotura por fatiga del mecanismo de impulsión.

Aunque la geometría simulada en este caso sea más próxima a la real, éste está siendo simulado con un aumento de la presión evaluado por la ecuación de Tait 4.2, que depende del volumen inicial. Como se ha dicho, y por tanto basándonos en ésto podemos concluir que la compresión calculada será más exacta en el caso 2D axisimétrico de la sección 5.2, dado que el volumen que es comprimido corresponde con más exactitud con el volumen real. En este caso se ha añadido el volumen de la válvula giratoria pero no se ha tenido en cuenta el volumen que se debería quitar de la cabeza del pistón real, que sería aproximadamente igual al añadido para la válvula giratoria, por tanto a la simulación 3D habría que sustraerle un volumen igual al añadido, quedando un volumen inicial igual al caso 2D axisimétrico.

## 5.4. Análisis de los resultados

Tras haber realizado un estudio tanto a nivel cualitativo como cuantitativo en busca del por qué de la rotura de estas bombas volumétricas, abordando diferentes campos de investigación para hallar la causa, el estudio simulado de las mismas ha arrojado resultados coherentes con el fenómeno que provocaba la inutilización de las bombas, los resultados obtenidos son reflejados en la tabla 5.1. Esta tabla recoge los valores obtenidos más próximos al caso real, el caso 2D axisimétrico, como se ha concluido en el párrafo anterior.

Simulación	Presión Máxima (Pa)	Fuerza Máxima sobre la cabeza del pistón (N)
2D axisimétrica	$1,0827 \cdot 10^8$	688567

Tabla 5.1: Resultados de la simulación de la bomba volumétrica

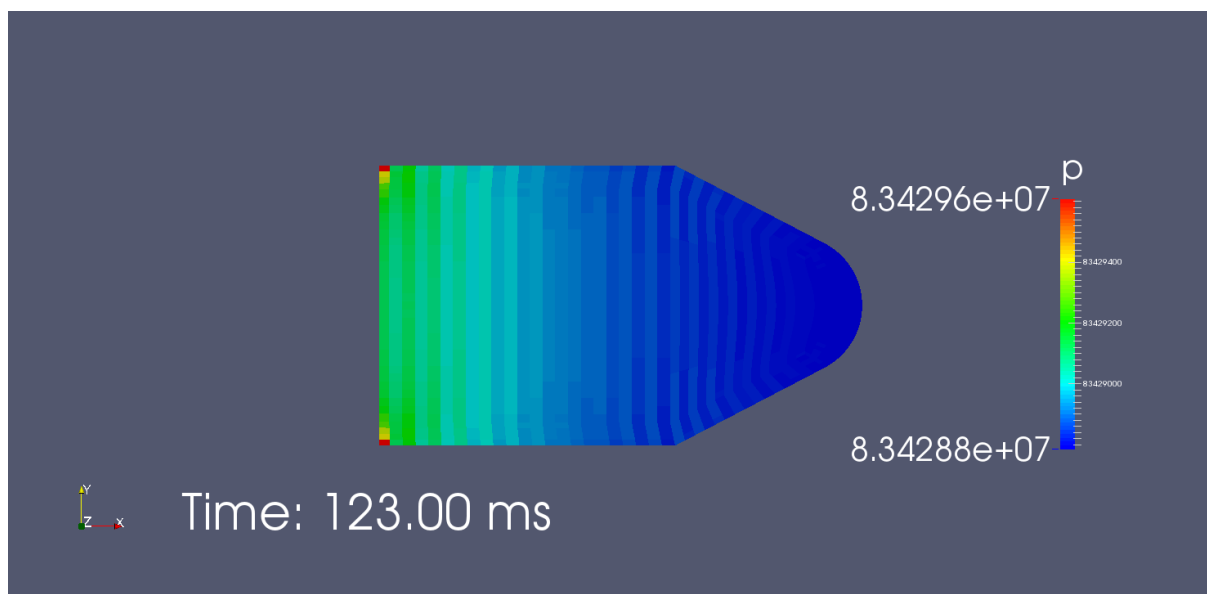


Figura 5.7: Simulación 3D con simetría de la bomba volumétrica

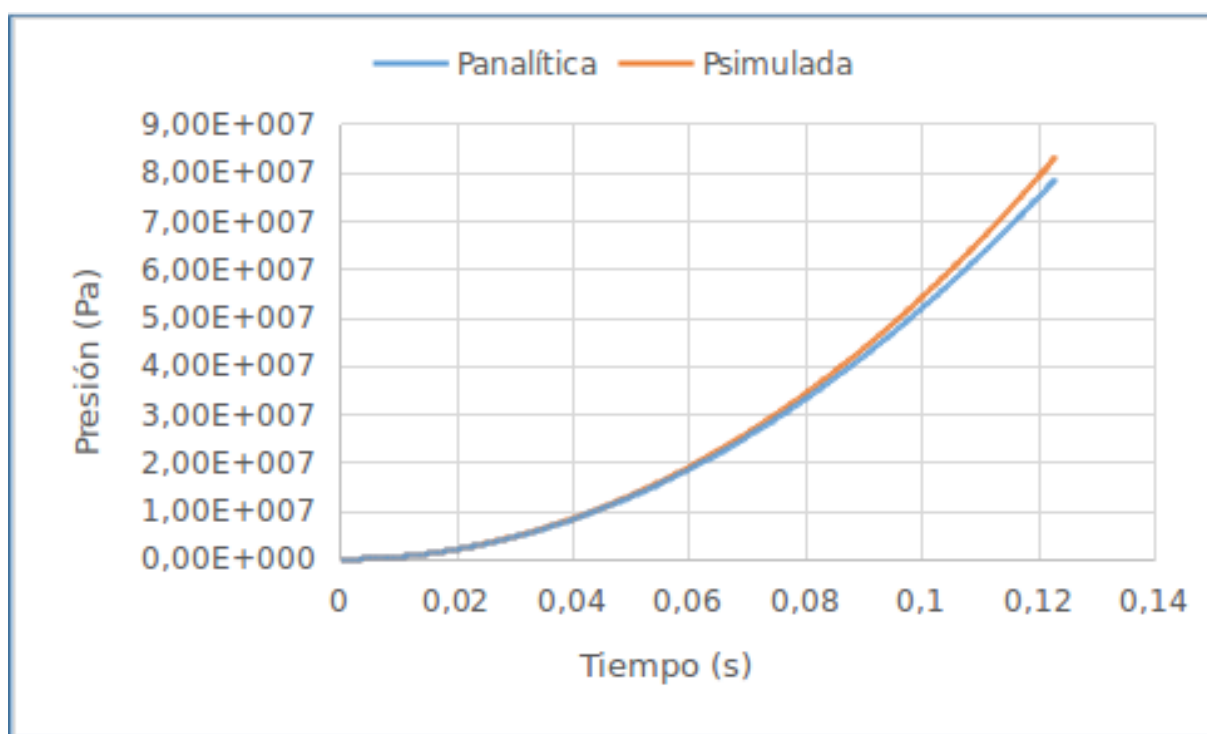


Figura 5.8: Comparativa de los resultados analíticos y simulados con el nuevo volumen

## Capítulo 6

# Conclusiones del estudio realizado y planteamiento de posibles soluciones

### 6.1. Conclusiones

El estudio de estas bombas volumétricas vino motivado por la solicitud de estudio de los problemas que le surgían a las empresas francesas que emplean dichas bombas en sus cadenas de producción de productos lácteos, mayonesas y demás fluidos alimenticios. Estas empresas alegaban que dichas bombas sufrían una rotura en el mecanismo de impulsión cuando llevaban cierto tiempo trabajado, lo que nos ha llevado a pensar que la rotura de las mismas podía ser debida a fenómenos de fatiga en dicho mecanismo.

A lo largo del estudio se han empleado distintas metodologías para tratar de evaluar el ciclo de funcionamiento de las bombas y detectar el origen de fuertes fluctuaciones de presión capaces de provocar la rotura por fatiga.

Inicialmente se abordó la posibilidad de que las sobrepresiones fuesen originadas por fenómenos de golpe de ariete en los procesos de apertura y cierre de la válvula giratoria. La indefinición de las condiciones de contorno necesarias para interpretar el modelo, junto con las sospechas relacionadas con una parte del ciclo en la que la bomba trataba de impulsar un fluido confinado, con la válvula de escape totalmente cerrada, nos hizo abandonar esta metodología de estudio para centrarnos en la evaluación de esa etapa del ciclo.

En este nuevo campo de estudio, se consiguió determinar el tiempo durante el cual la bomba estaba impulsando el fluido con la válvula cerrada, y se procedió a realizar una simulación numérica del caso durante este tiempo, con el motivo de hallar la presión máxima que se produce en la cámara del pistón, y por tanto la fuerza que debe soportar el mecanismo en cada ciclo. Esta simulación se ha contrastado con un cálculo analítico de la presión mediante la ecuación que simula la compresión de fluido por variación de su densidad, ecuación de Tait 4.2.

Los resultados obtenidos, recogidos en la sección 5.4, reflejan y contrastan las hipótesis realizadas sobre la sobrepresión generada en la cámara interna de la bomba. Este exceso de presión se transmite en forma de grandes fuerzas que deben soportar las partes del mecanismo de impulsión, siendo más afectada la pieza con menor diámetro la cual soporta las mayores fluctuaciones de fuerza durante el ciclo de trabajo de la bomba. Efectivamente es ésta pieza, señalada en la figura 4.1, donde tiene lugar la rotura por fatiga indicada por las empresas comentadas.

En un estudio más exhaustivo posterior podría tratar de evaluarse los fenómenos que tienen lugar cuando se libera esa tremenda sobrepresión generada con la apertura de la válvula para la impulsión del fluido. Creemos que en la liberación de fluido debido a las enormes velocidades que se generarán por el elevado gradiente de presión entre la cámara interna y la presión de salida del fluido, probablemente la atmosférica, es muy posible que aparezcan fenómenos de cavitación y golpe de ariete los cuales pueden contribuir aun más al fenómeno de rotura por fatiga de estas bombas.

## 6.2. Planteamiento de soluciones al problema

Con el objetivo de solventar este problema y aumentar la vida útil de estas bombas, a continuación se plantean a la empresa unas medidas que permitan que durante el tramo de tiempo en el que el fluido se está comprimiendo sin ser impulsado, la presión en la cámara del pistón no aumente enormemente.

Para ello hemos pensado la instalación de un *bypass* que conecte la cámara interna del pistón y la admisión de fluido, colocado de manera que durante los 0,123 segundos de la etapa de impulsión durante la cuál la válvula está cerrada, se libere fluido a la zona de admisión, con el objetivo de evitar la compresión del mismo a costa de la pérdida de una pequeña parte de volumen de fluido impulsado en cada ciclo.

La tubería de *bypass* debe ir colocada de tal forma que la propia cabeza del pistón tapone la salida de fluido por ella, justo cuando se produce la apertura de la válvula giratoria. Para esto es necesario colocar la tubería a una distancia obtenida con la ecuación 5.1 para  $t = 0,123$  s, que indica que esta tubería debe colocarse a 5 mm del punto muerto inferior, por lo que con una tubería de este diámetro debería ser suficiente para solventar el problema.

La figura 6.1 muestra un esquema de la solución planteada a la empresa.

Con esta modificación en la configuración de la bomba se consigue evitar las tremendas sobrepresiones internas a costa de una ínfima cantidad de volumen reenviado a la admisión, aproximadamente 0,03 litros.

Otra posible solución es la de modificar el área de paso de la válvula giratoria alargándola de manera que la apertura sea más gradual. En el momento justo en el que deja de entrar fluido a la cámara de la bomba debe comenzar la liberación de fluido a la impulsión. Esto puede lograrse alargando el área de paso de la válvula, desde la circunferencia a una geometría con forma lagrimal como muestra la figura 6.2, de esta forma el fluido no sufriría compresión estática.

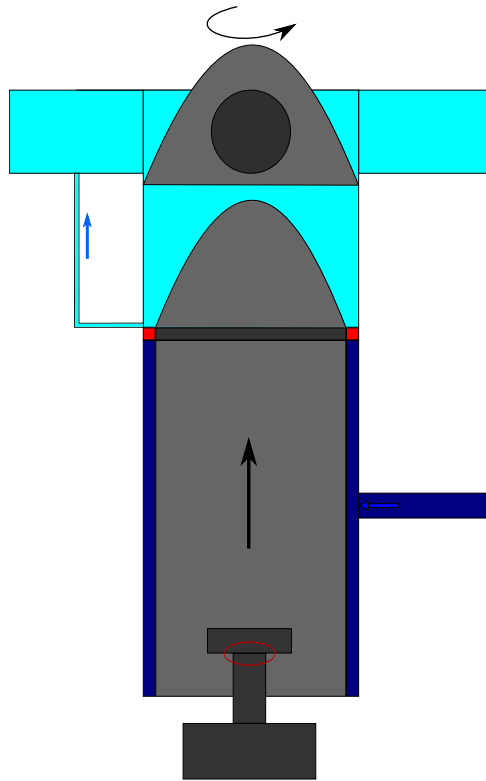


Figura 6.1: Solución mediante bypass

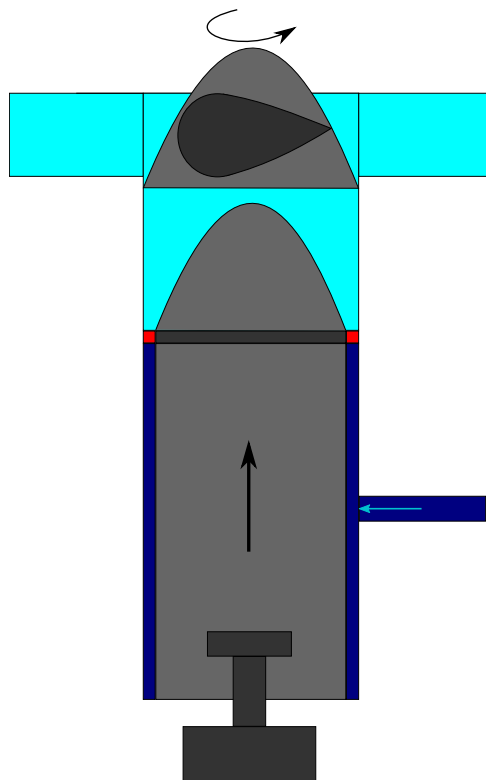


Figura 6.2: Solución por modificación del área de paso





# Bibliografía

- [1] U. R. Charrondiere, D. Haytowitz, and B. Stadlmayr. Fao/infods density database version 2.0. *Food and Agriculture Organization of the United Nations*, 2008.
- [2] M. A. Ghoush, M. Samhouri, M. Al-Holy, and T. Herald. Formulation and fuzzy modeling of emulsion stability and viscosity of a gum–protein emulsifier in a model mayonnaise system. *Journal of Food Engineering*, 84(2):348 – 357, 2008.
- [3] T. Leephakpreeda. Design factors for “linear” ball valve: theoretical and experimental studies. *Songklanakarin J. Sci. Technol.*, 27(2):353–361, 2005.
- [4] M. Lema Rodríguez. *Multiphase Fluid Hammer: Modeling, Experiments and Simulations*. PhD Thesis from the von Karman Institute / Université Libre de Bruxelles / Universidade da Coruña, 2013.
- [5] D. McClements. Ultrasonic characterization of foods and drinks: Principles, methods, and applications. *Critical Reviews in Food Science and Nutrition*, 37(1):1–46, 1997. cited By 155.
- [6] J. Parmakian. *Waterhammer analysis*. Dover, New York, 1963.
- [7] I. Romero Olleros. *Mecánica de sólidos*. Universidad Politécnica de Madrid, 2016. Apuntes de clase.
- [8] A. Tijsseling. Fluid-structure interaction in liquid-filled pipe systems: A review. *Journal of Fluids and Structures*, 10(2):109–146, 1996. cited By 157.
- [9] H. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007.



## Apéndice A

# Código fuente de la utilidad *movingConeTopoFVMesh\_v2*

```
/*-----*\
===== /
\\ / F i e l d / OpenFOAM: The Open Source CFD Toolbox
\\ / O p e r a t i o n /
\\ / A n d / Copyright (C) 2011 OpenFOAM Foundation
  \\/ M a n i p u l a t i o n /
-----*/

License

This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

/*-----*/

#include "movingConeTopoFvMesh_v2.H"
#include "Time.H"
#include "mapPolyMesh.H"
#include "layerAdditionRemoval.H"
#include "addToRunTimeSelectionTable.H"
#include "meshTools.H"
#include "OFstream.H"

// * * * * * S t a t i c D a t a M e m b e r s * * * * * //

namespace Foam
{
    defineTypeNameAndDebug(movingConeTopoFvMesh_v2, 0);

    addToRunTimeSelectionTable
    (
```

```

        topoChangerFvMesh,
        movingConeTopoFvMesh_v2,
        IOobject
    );
}

// * * * * * Private Member Functions * * * * * //

Foam::tmp<Foam::scalarField> Foam::movingConeTopoFvMesh_v2::vertexMarkup
(
    const pointField& p,
    const scalar curLeft
    // const scalar curRight /*
) const
{
    Info<< "Updating vertex markup.  curLeft: "
        << curLeft << endl;

    tmp<scalarField> tvertexMarkup(new scalarField(p.size()));
    scalarField& vertexMarkup = tvertexMarkup();

    //Definición de vertexMarkup, función que determina la forma de la máscara con
    los
    // valores de entrada p (puntos de la malla), curleft (eje del obstaculo
    izquierdo)
    // y curlright (eje del obstáculo derecho)
    forAll(p, pI)
    {
        if (p[pI].x() > curLeft + SMALL) /**v2
        {
            vertexMarkup[pI] = 1; //Zona izquierda
        }
        /*else if (p[pI].x() > curRight + SMALL) /**v2
        {
            vertexMarkup[pI] = 1; //Zona derecha
        }*/
        else
        {
            vertexMarkup[pI] = 0; //Zona intermedia. Sin cambios
        }
    }

    return tvertexMarkup;
}

void Foam::movingConeTopoFvMesh_v2::addZonesAndModifiers()
{
    // Add zones and modifiers for motion action
    //Inicialización? Sólo se ejecuta la primera iteración?

    if //Si ya están presentes alguno de los parámetros de las zonas
        // estas no se seleccionan en su configuración por defecto
    (
        pointZones().size()
        || faceZones().size()
        || cellZones().size()
        || topoChanger_.size()
    )
    {
        Info<< "void movingConeTopoFvMesh_v2::addZonesAndModifiers() : "

```

```

        << "Zones and modifiers already present. Skipping."
        << endl;

    return;
}

Info<< "Time = " << time().timeName() << endl
    << "Adding zones and modifiers to the mesh" << endl;

const vectorField& fc = faceCentres();
const vectorField& fa = faceAreas();

labelList zone1(fc.size());
boolList flipZone1(fc.size(), false);
label nZoneFaces1 = 0;

labelList zone2(fc.size());
boolList flipZone2(fc.size(), false);
label nZoneFaces2 = 0;

forAll(fc, faceI)
{
    if //Si fc[faceI].x() == -0.0035 // fc=faceCentres()
    (
        fc[faceI].x() > -0.003501
        && fc[faceI].x() < -0.003499
    )
    {
        if ((fa[faceI] & vector(1, 0, 0)) < 0) // Se comprueba si está
            orientada? correctamente // fa=faceAreas
        {
            flipZone1[nZoneFaces1] = true;
        }

        zone1[nZoneFaces1] = faceI;
        Info<< "face " << faceI << " for zone 1. Flip: "
            << flipZone1[nZoneFaces1] << endl;
        nZoneFaces1++;
    }
    else if //Si fc[faceI].x() == -0.007
    (
        fc[faceI].x() > -0.00701
        && fc[faceI].x() < -0.00699
    )
    {
        zone2[nZoneFaces2] = faceI;

        if ((fa[faceI] & vector(1, 0, 0)) > 0)
        {
            flipZone2[nZoneFaces2] = true;
        }

        Info<< "face " << faceI << " for zone 2. Flip: "
            << flipZone2[nZoneFaces2] << endl;
        nZoneFaces2++;
    }
}

zone1.setSize(nZoneFaces1);
flipZone1.setSize(nZoneFaces1);

zone2.setSize(nZoneFaces2);

```

```

flipZone2.setSize(nZoneFaces2);

Info<< "zone: " << zone1 << endl;
Info<< "zone: " << zone2 << endl;

List<pointZone*> pz(0);
List<faceZone*> fz(2);
List<cellZone*> cz(0);

label nFz = 0;

/* fz[nFz] =
   new faceZone
   (
       "rightExtrusionFaces",
       zone1,
       flipZone1,
       nFz,
       faceZones()
   );
nFz++; */

fz[nFz] =
  new faceZone
  (
      "leftExtrusionFaces",
      zone2,
      flipZone2,
      nFz,
      faceZones()
  );
nFz++;

fz.setSize(nFz);

// Se añaden las zonas creadas fz=facezone (dimensión 2), cz=cellzone y
  pz=pointzone (dimensión 0)
Info<< "Adding mesh zones." << endl;
addZones(pz, fz, cz);

// Add layer addition/removal interfaces
List<polyMeshModifier*> tm(2);
label nMods = 0;

// Se define topoChanger_ con los valores mínimos y máximos de espesor para
  cambios en la topología
// Es una función especial que permite saber cuando se realizan cambios y
  cuando no.
tm[nMods] =
  new layerAdditionRemoval
  (
      "right",
      nMods,
      topoChanger_,
      "rightExtrusionFaces",
      readScalar
      (
          motionDict_.subDict("right").lookup("minThickness")
      ),
      readScalar
      (

```

```

        motionDict_.subDict("right").lookup("maxThickness")
    )
);
nMods++;

tm[nMods] = new layerAdditionRemoval
(
    "left",
    nMods,
    topoChanger_,
    "leftExtrusionFaces",
    readScalar
    (
        motionDict_.subDict("left").lookup("minThickness")
    ),
    readScalar
    (
        motionDict_.subDict("left").lookup("maxThickness")
    )
);
nMods++;
tm.setSize(nMods);

    // Se añade tm (dimensión 2 (right y left)) que indica layer modificable
    // y cuánto
Info<< "Adding " << nMods << " mesh modifiers" << endl;
topoChanger_.addTopologyModifiers(tm);

write();
}

// * * * * * Constructors * * * * * //

// Construct from components
Foam::movingConeTopoFvMesh_v2::movingConeTopoFvMesh_v2(const IOobject& io)
:
//Se LEEN los datos de motionDict
topoChangerFvMesh(io),
motionDict_
(
    IOdictionary
    (
        IOobject
        (
            "dynamicMeshDict",
            time().constant(),
            *this,
            IOobject::MUST_READ_IF_MODIFIED,
            IOobject::NO_WRITE,
            false
        )
    ).subDict(typeName + "Coeffs")
),
motionVelAmplitude_(motionDict_.lookup("motionVelAmplitude")),
motionVelPeriod_(readScalar(motionDict_.lookup("motionVelPeriod"))),
curMotionVel_
(
    motionVelAmplitude_*
    Foam::sin(time().value()*M_PI/motionVelPeriod_)
),
leftEdge_(readScalar(motionDict_.lookup("leftEdge"))),

```

```

//Se LEEN curLeft y curRight en motionDict como los valores asignados a
leftObstacleEdge y rightObstacleEdge
curLeft_(readScalar(motionDict_.lookup("leftObstacleEdge")))
// curRight_(readScalar(motionDict_.lookup("rightObstacleEdge"))) //v2
{
    Pout<< "Initial time:" << time().value()
        << " Initial curMotionVel_:" << curMotionVel_
        << endl;

//Se añaden las zonas y modificadores a mover:
//      pointZones faceZones cellZones topoChanger_
//Para el ejemplo pointZones y cellZones están vacíos
addZonesAndModifiers();

//Se LEEN los datos de faceZones
//Se buscan las caras nombradas como leftExtrusionFaces y rightExtrusionFaces
//Se redefinen el valor de curleft y curRight como el valor medio en x de estas
caras
curLeft_ = average
(
    faceZones()
    [
        faceZones().findZoneID("leftExtrusionFaces")
    ]().localPoints()
).x() + SMALL;

/* curRight_ = average //v2...
(
    faceZones()
    [
        faceZones().findZoneID("rightExtrusionFaces")
    ]().localPoints()
).x() + SMALL;*/ //v2...

//Se definen los valores de motionMask como la asignación de 0 -1 y 1 a cada
punto de la malla
// tomando como referencia los valores de curLeft y curRight
// Estas líneas podrían ser redundantes (salvo *)
motionMask_ = vertexMarkup
(
    points(),
    curLeft_
    //curRight_ //v2
);
}

// * * * * * Destructor * * * * * //

// El destructor libera la memoria empleada para el constructor
Foam::movingConeTopoFvMesh_v2::~movingConeTopoFvMesh_v2()
{}

// * * * * * Member Functions * * * * * //

bool Foam::movingConeTopoFvMesh_v2::update()
{
    // Do mesh changes (use inflation - put new points in topoChangeMap)
    autoPtr<mapPolyMesh> topoChangeMap = topoChanger_.changeMesh(true);

    // Calculate the new point positions depending on whether the

```



```

// topological change has happened or not
pointField newPoints;

vector curMotionVel_ =
    motionVelAmplitude_*
    Foam::sin(time().value()*M_PI/motionVelPeriod_);

Pout<< "time:" << time().value() << " curMotionVel_:" << curMotionVel_
    << " curLeft:" << curLeft_ << endl;

// Se calculan los nuevos puntos, newPoints, pertenecientes a la máscara
// Para calcularlos se parte de topoChangeMap().preMotionPoints() si existe
// en caso contrario de points()
if (topoChangeMap.valid())
{
    Info<< "Topology change. Calculating motion points" << endl;

    if (topoChangeMap().hasMotionPoints())
    {
        Info<< "Topology change. Has premotion points" << endl;

        motionMask_ =
            vertexMarkup
            (
                topoChangeMap().preMotionPoints(),
                curLeft_
                //curRight_ //**v2
            );

        // Move points inside the motionMask
        newPoints =
            topoChangeMap().preMotionPoints()
            + (
                pos(0.5 - mag(motionMask_)) // cells above the body
            )*curMotionVel_*time().deltaT().value();
    }
    else
    {
        Info<< "Topology change. Already set mesh points" << endl;

        motionMask_ =
            vertexMarkup
            (
                points(),
                curLeft_
                //curRight_ //**v2
            );

        // Move points inside the motionMask
        newPoints =
            points()
            + (
                pos(0.5 - mag(motionMask_)) // cells above the body
            )*curMotionVel_*time().deltaT().value();
    }
}
else
{
    Info<< "No topology change" << endl;
    // Set the mesh motion
    // En este punto no se redefine la motionMask_, no es necesario
    *

```

```

        newPoints =
            points()
            + (
                pos(0.5 - mag(motionMask_)) // cells above the body
            ) * curMotionVel_ * time().deltaT().value();
    }

    // The mesh now contains the cells with zero volume
    Info << "Executing mesh motion" << endl;
    movePoints(newPoints);
    // The mesh now has got non-zero volume cells

//Se LEEN los datos de faceZones
//Se buscan las caras nombradas como leftExtrusionFaces y rightExtrusionFaces
//Se redefinen el valor de curLeft y curRight como el valor medio en x de estas
    caras
//Como se han movido los puntos se esperan valores diferentes
    curLeft_ = average
    (
        faceZones()
        [
            faceZones().findZoneID("leftExtrusionFaces")
        ]().localPoints()
    ).x() + SMALL;

    /*curRight_ = average /**v2...
    (
        faceZones()
        [
            faceZones().findZoneID("rightExtrusionFaces")
        ]().localPoints()
    ).x() + SMALL; */ /**...v2

    return true;
}

// ***** //

```