

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Modelado CFD del movimiento del aire en el cilindro de un motor maqueta de dos tiempos

Trabajo Fin de Grado en Ingeniería Mecánica

Autor:

Borja Mas Viñes

Tutor:

Jose María García Oliver

Valencia, junio de 2018

Resumen

El objetivo de este proyecto es conseguir una mejora en los resultados de los cálculos CFD de la dinámica del fluido de un motor monocilindrico diésel. El trabajo realizado anteriormente utilizaba una geometría aproximada. En cambio, ahora se tiene una geometría más exacta.

Los datos experimentales provienen de un trabajo experimental en el que se cuantificó el movimiento del aire en la cámara de combustión a través de la técnica PIV. A partir de este hecho, el alumno usa el Converse CFD para preparar el pre-procesado del estudio, introducir los datos necesarios para la *Set Up* y seguidamente, lanzar los cálculos. Después del cálculo se ejecuta el post-procesado con ayuda de Matlab y Enight. Con estos dos programas se puede convertir los datos obtenidos en formato numérico en tablas, gráficas y representaciones para poder analizar los resultados. Se comparan los resultados de los cálculos con las medidas experimentales, de manera que se pueden observar las variaciones introducidas por la nueva geometría.

Índice

1.	Introducción	9
1.1	Antecedentes	9
1.2	Objeto del proyecto	10
1.3	Justificación	10
1.4	Motivación	11
1.5	Viabilidad	11
1.6	Estructura del proyecto	11
2.	Conceptos previos	13
2.1	Introducción	13
2.2	Funcionamiento de un motor Diésel	13
2.3	Funcionamiento de un motor de 2T	14
2.4	Flujos internos de un MCI	16
2.5	Geometría lumbreras de admisión y escape	17
3.	Modelado en Converge CFD	19
3.1	Introducción	19
3.2	Mallado en Converge	19
3.3	Condiciones de contorno	20
3.4	Ecuaciones de la mecánica de fluidos	20
3.5	Resolución de un flujo transitorio	24
4.	Software de post-procesado	26
4.1	Ensign	26
4.1.1	Geometría en Ensign	26
4.1.2	Visualización de la geometría	26
4.2	Matlab	27
5.	Análisis de resultados	29
5.1	Introducción	29
5.2	Descripción del <i>motor maqueta</i>	29
5.3	Configuración de las simulaciones	32
5.3.1	Mallado	32
5.3.2	Condiciones iniciales	32
5.3.3	Número de ciclos	33
5.4	Presión en cámara	34
5.5	Campo de velocidades	34
5.5.1	Velocidades axiales	41
5.6	Turbulencia	42

6.	Conclusiones	44
6.1	Trabajos futuros	45
7.	Bibliografía	46
	ANEXO	47
I.	Pre-procesado	47
I.1	Introducción	47
I.2	Diseño del motor	47
I.3	Asignación de los contornos	48
I.4	Preparación de la geometría	50
I.4.1	Seals	51
I.5	Preparación de la <i>Set Up</i>	53
I.5.1	Applications	54
I.5.2	Materials	55
I.5.3	Simulation Parameters	56
I.5.4	Boundary Conditions	58
I.5.5	Initial Conditions & events	58
I.5.6	Physical Models	60
I.5.7	Grid Control	60
I.5.8	Output/Post Processing	60
II	Cálculo del modelo	61
II.1	Introducción	61
II.2	MobaXterm	61
III	Post-procesado	61
III.1	Introducción	61
III.2	Rutinas de Matlab	62
III.2.1	Comparación PIV y modelo actual	62
III.2.2	Comparación modelo CFD anterior y actual	72
III.2.3	Unión imágenes varios planos	77
III.2.4	Imágenes a vídeo	78
III.3	EnSight	80
III.3.1	Introducción	80
III.3.2	Primeros pasos en EnSight	80
III.3.3	Creación de <i>clips</i>	81
III.3.4	Configuración de la representación	83
III.3.5	Grabación de la simulación en imágenes	85

PRESUPUESTO	86
A. Introducción	86
B. Gastos	87
B.1 Gastos en la mano de obra	87
B.2 Gastos en licencias	88
B.2 Gastos en equipos informáticos	89
B.3 Gastos energéticos	90
C. Cuadro de precios descompuestos	91
C.1 Introducción	91
C.2 Ingeniero Director del Proyecto	91
C.3 Ingeniero Tutor de Proyecto	92
C.4 Ingeniero de Proyecto	92
D. Cuadro de mediciones	93
E. Resumen del presupuesto	94

Índice de figuras

Figura 2.1: Gráfica P-V de un motor de encendido por compresión	13
Figura 2.2: Representación del ciclo completo de un motor de 4T.	14
Figura 2.3: Representación de un ciclo completo de un motor de 2T	15
Figura 2.4: Representación del barrido de gases de un motor de 2T	16
Figura 2.5: Representación de los diferentes movimientos flujos en el interior de un cilindro. Swirl, tumble y squish de izquierda a derecha respectivamente.	17
Figura 2.6: Tipos de barrido y posición y tamaño de las lumbreras de un motor de 2T. De izquierda a derecha: transversal, uniflujo con pistones opuestos, uniflujo con válvulas de escape en la cabeza, por lazo de Schnürle, por lazo Man, por lazo Curtis y en gancho con válvulas en cabeza	18
Figura 3.1: Pasos para la resolución de las ecuaciones en Converge	25
Figura 5.1: Medidas de la cámara de combustión modificada	28
Figura 5.2: Vista de los planos usados en la técnica PIV	29
Figura 5.3: Campo de velocidades promediadas en el plano CC en -10 CAD	34
Figura 5.4: Campo de velocidades promediadas en el plano CC en +10 CAD	35
Figura 5.5: Campo de velocidades ciclo a ciclo en el plano CC en -10 CAD	36
Figura 5.6: Campo de velocidades ciclo a ciclo en el plano CC en +10 CAD	36
Figura 5.7: Gasto másico a lo largo de la cámara de combustión en -10 CAD	37
Figura 5.8: Gasto másico a lo largo de la cámara de combustión en +10 CAD	37
Figura 5.9: Representación del <i>motor maqueta</i> en Ensignht en la carrera de compresión	38
Figura 5.10: Campo de velocidades promediadas en el plano CC en 3 CAD	39
Figura 5.11: Turbulencia promediada en el plano CC en -10 CAD	41
Figura 5.12: Turbulencia promediada en el plano CC en +10 CAD	42
Figura I.1: Geometría modelo antiguo sin el plenum de admisión	48
Figura I.2: Configuración actual de la geometría con el plenum de admisión (izquierda) y el modelo actual sin el plenum de admisión (derecha)	48
Figura I.3: Ventana Geometry>Boundary>Flag para asignar las superficies deseadas	49
Figura I.4: Ventana de Diagnosis con los errores encontrados de la geometría actual	50
Figura I.5: Ventana para la creación de seals	53

Figura I.6: Ventana de la Case Setup para la configuración del modelo	54
Figura I.7: Ventana para la configuración de applications del modelo	55
Figura I.8: Ventana para la configuración de materiales del modelo	56
Figura I.9: Menú de los Run Parameters	57
Figura I.10: Menú de los Simulation Time Parameters	57
Figura I.11: Menú de los Boundary	58
Figura I.12: Ventana para la configuración de Initial Conditions del modelo	59
Figura I.13: Menú de los Initial Conditions para cada región	59
Figura I.14: Ventana para la configuración de Grid Control del modelo	60
Figura III.0.1: Representación de la geometría actual en EnSight	81
Figura III.0.2: Ventanas para la creación de un plano de corte	82
Figura III.0.3: Representación de la velocidad en el plano de corte ortogonal a X	82
Figura III.0.4: Menú de las variables representadas en el modelo	83
Figura III.0.5: Representación de los valores de RPM y ángulo de cigüeñal (CAD)	83
Figura III.0.6: Ventanas para la modificación del color de fondo	84
Figura III.0.7: Ventanas para la modificación del formato de las leyendas de las variables	84
Figura III.0.8: Ventanas para elegir el formato de la imagen	85
Figura III.0.9: Ventana para la elección del momento de representación del modelo o de ver la simulación como un vídeo	85

Índice de tablas

Tabla 5.1: Características del motor maqueta	30
Tabla 5.2: Condiciones iniciales en cada región del modelo	33
Tabla B.3: Coste anual del salario de un Ingeniero Mecánico Junior	87
Tabla B.4: Coste anual del salario de un Ingeniero Tutor de Proyecto	87
Tabla B.5: Coste anual del salario de un Ingeniero Director de Proyecto	88
Tabla B.1: Coste, vida útil, amortización, dedicación y coste total de las licencias usadas en este proyecto	89
Tabla B.2: Coste, vida útil, amortización, dedicación y coste total del equipo informático usado en este proyecto	89
Tabla B.6: Rendimiento y potencia [kW] de los sistemas eléctricos	90
Tabla C.1: Precio desglosado del coste total para el Ingeniero Director del Proyecto	91
Tabla C.2: Precio desglosado del coste total para el Ingeniero Tutor del Proyecto	92
Tabla C.3: Precio desglosado del coste total para el Ingeniero Mecánico Junior	92
Tabla D.1: Coste total del proyecto sin IVA	93
Tabla E.1: Coste total del proyecto con IVA	94

1. Introducción

1.1 Antecedentes

El sector del automóvil ha sufrido grandes cambios durante los últimos años, sobre todo el motor de encendido por compresión, también conocido como motor Diesel. Los gobiernos de los países del primer mundo se han dado cuenta de lo peligroso que pueden llegar a ser los contaminantes del motor diésel, especialmente las partículas sólidas que forma a diferencia del encendido provocado. Por esta razón, hay un esfuerzo muy grande por parte de las grandes marcas automovilísticas para mantener el mercado del diésel y que cumplan los requisitos mínimos que cada vez restringen más los gobiernos. Además, siempre hay un intento de mejorar las prestaciones de los motores, como la potencia de estos con menos cilindrada o el consumo de carburante para que sea más atractivo para el comprador.

Las características de los motores diésel están condicionadas por la mezcla entre la inyección de combustible y el fluido que está en el interior del cilindro, cuanto más rápida sea la mezcla también más rápida será la combustión. En consecuencia, el estudio del modelado aerodinámico en el interior de un cilindro diésel es el aspecto más importante hablando del rendimiento de este.

El motor que se usa en el Instituto Universitario CMT-Motores Térmicos para entender mejor el funcionamiento fluidodinámico de un motor diésel es el *motor maqueta*, un motor monocilindro de gran cilindrada con accesos ópticos en los laterales de la culata, con el cual se ha podido realizar un estudio mediante la Velocimetría de Imágenes por Partículas (PIV). La técnica PIV consiste en dopar el aire de la cámara de combustión con partículas sólidas. En el momento de la inyección, se hace un *Time-lapse* con láser, donde las partículas reflejan la luz y de esta forma se puede caracterizar el campo de velocidades en el interior de la cámara. Esta técnica es muy acertada para ver los resultados empíricos del estudio dinámico del cilindro.

En el trabajo de final de grado realizado por P. Fernández Bérjon. “Caracterización del flujo interno en la cámara de combustión de un motor óptico mediante velocimetría por imágenes de partículas” se pudo observar que el campo de velocidades dentro del *motor maqueta* no era simétrico dentro de la cámara de combustión.

En la realización de ensayos se ha visto una dispersión clara entre los primeros ciclos y los últimos para la mayoría de los parámetros que se miden. Los experimentos se han repetido en otra instalación del departamento, que reproduce las condiciones de presión y temperatura en la inyección, pero con un flujo estacionario y no se observan estas diferencias de simetría que se ven en el caso anterior. Por este motivo, una posible causa es que el movimiento fluidodinámico dentro de la cámara de combustión del motor maqueta afecta a los ensayos.

Es importante citar el trabajo anterior a este, el TFM de María José Ruiz Lucas, “Modelado CFD del movimiento del aire en el cilindro en un motor de dos tiempos con accesos ópticos”. Es un proyecto muy parecido al presente, pero con una geometría aproximada, hecho que se rectifica en este estudio.

Con los datos empíricos de velocidad obtenidos, el siguiente objetivo es validar modelos teóricos con CFD. El modelo teórico es ideal para el estudio de infinidad de situaciones y de

diseños, sin gastos en ensayos ni limitaciones tecnológicas. El CFD contribuye a mejorar la comprensión del funcionamiento de motores y a ampliar la información sobre ellos. El inconveniente que tiene es la complejidad de los cálculos de las variables, donde se necesita una gran capacidad computacional para que sean cálculos que se resuelvan relativamente en poco tiempo. El código que se usa para los cálculos es el código Converge.

En este proyecto también se comprueba que se necesita un gran conocimiento en la teoría de la combustión y en el uso de programas de código CFD y de post-procesado como el Matlab o el Enight para poder llevar a cabo el estudio de un simple caso.

1.2 Objeto del proyecto

El objetivo de este proyecto es la realización de un estudio fluidodinámico del interior de un motor diésel de 2 tiempos. Se busca que las simulaciones calculadas con el programa Converge CFD tengan las mismas tendencias que el estudio experimental efectuada con la técnica PIV.

Seguidamente a obtener los resultados de los cálculos de Converge, se efectuará un post-procesado con los programas Matlab y Enight, usando y modificando unas rutinas ya hechas por el equipo del Instituto Universitario CMT-Motores Térmicos. Será necesario este post-procesado para poder comparar de una forma fácil y efectiva los resultados obtenidos del Converge y del estudio PIV. Si el nuevo modelo teórico se acerca a la realidad y poder ejecutarlo en más ocasiones para que haya un ahorro en ensayos prácticos.

Los resultados de este estudio se comparan con los resultados de un Trabajo de Final de Máster que se realizó anteriormente. El pasado proyecto se realizó con una geometría aproximada y, en este estudio, se va a usar una geometría exacta después de un escaneado del bloque y lumbreras.

Este trabajo también es de ámbito académico y tiene el objetivo de obtener el título de Grado de Ingeniería Mecánica en la Escuela Superior de Ingeniería de Diseño de la Universidad Politécnica de Valencia.

1.3 Justificación

El sector automovilístico está a un nivel de perfeccionamiento muy alto, ya sea por su antigüedad, cada avance en la actualidad es mucho más costoso que en sus inicios, o sea por la dificultad de la resolución de problemas de la combustión o la aerodinámica.

Actualmente, es necesario el uso de programas de apoyo para la resolución de problemas de fluidos. Estos programas se encargan de la simulación de los fluidos mediante la técnica de la dinámica de fluidos computacional, CFD (en inglés, *Computational Fluid Dynamics*).

Si se consigue un modelo teórico a través de los programas CFD que se ajuste a un caso real, se podrá modificar ese modelo teórico para obtener mejoras sin tener que gastar el dinero en bancos de ensayo, que acarrear gastos en material, trabajadores para su instalación, etc.

1.4 Motivación

El siguiente proyecto se ha hecho por el interés personal en el sector del automóvil, específicamente, el mundo de la combustión. Los conocimientos aprendidos durante la preparación de este estudio han servido para afianzar las ideas ya obtenidas durante las diferentes asignaturas que se han trabajado a lo largo del Grado en Ingeniería Mecánica.

Al ser un proyecto de un calibre superior a cualquiera que se ha hecho con anterioridad, plantea una actitud de superación y esfuerzo para sacar adelante cualquier complicación que aparezca durante el transcurso de este.

Es importante comentar que este proyecto intenta ser la finalización de un estudio que han hecho otros alumnos para llegar a crear un modelo teórico para el *motor maqueta* instalado en el DMMT de la UPV.

1.5 Viabilidad

Antes de empezar un proyecto hay que conceptuar factores que puedan hacer un proyecto viable o inviable. Los factores son el económico, el humano y el tecnológico.

El factor económico queda cubierto, ya que el DMMT cuenta con la subcontratación de grandes empresas de automoción para cubrir las investigaciones y desarrollo de éstas. Gracias a esta intensa actividad, el DMMT se garantiza su economía para llevar a cabo los trabajos encomendados por las empresas que los contratan y sus propias investigaciones.

El factor humano depende del personal de la empresa. En el DMMT hay una gran cantidad de trabajadores y estudiantes con una gran cualificación. Tienen la capacidad para garantizar un buen trabajo para las empresas que los contratan y son de gran ayuda para realizar proyectos como en este caso, ya que se pueden aprender los conocimientos y escuchar los consejos de personal especializado en el sector de la combustión.

El factor tecnológico también queda acreditado en el DMMT. Necesitan una gran potencia computacional para realizar los cálculos de las simulaciones y para nuestro caso, queda más que cubierto. Además, son de su propiedad numerosos bancos de ensayo, como, por ejemplo, el *motor maqueta* que se ha usado en proyectos precedentes al presente.

1.6 Estructura del proyecto

El presente proyecto se estructura en dos documentos:

- **Memoria:** es el principal documento del proyecto. Están escritos los conocimientos previos necesarios para entender el procedimiento de este estudio. Así como el proceso que se ha establecido para obtener los resultados del modelado y, por último, la interpretación y comparación de los resultados que se han obtenido tras el estudio y los ya obtenidos experimentalmente.

La memoria se distribuye de la siguiente forma:

- Apartado 2: se hace una introducción teórica previa de los conocimientos generales de la combustión interna de un motor, como el flujo interno del aire en un cilindro o los procesos que ocurren en los ciclos de un motor de dos y cuatro tiempos. Estos conocimientos allanan la compresión del actual proyecto.
 - Apartado 3: se centra en explicar brevemente el funcionamiento de Converge, un programa CFD. Se hace hincapié en el mallado de la región de estudio y en la forma de sacar los resultados de la simulación a partir de las condiciones iniciales que determina el usuario.
 - Apartado 4: se analizan los resultados de la simulación que se haya obtenido. Seguidamente se hace una comparación entre el modelo experimental desarrollado con la técnica PIV y la presente simulación.
 - Apartado 5: se formulan las conclusiones sustraídas de los resultados obtenidos en el apartado anterior.
 - Anexos: conciernen al pre-procesado y al post-procesado. Se explica el pre-procesado en Converge y todos los pasos que se han seguido antes de lanzar el cálculo. También, se reproduce la rutina utilizada para realizar el post-procesado y obtener unos resultados que se puedan comprender.
- Presupuesto: se especifican los costes del proyecto actual.

2. Conceptos previos

2.1 Introducción

En este apartado se explica de una forma breve el funcionamiento de Motores de Combustión Interna Alternativo (MCIA) y, dentro de este grupo, se especifica el funcionamiento de los motores de encendido por compresión (MEC).

Se define los tipos de formas de circulación de un fluido que hay dentro de un motor, como sus tendencias. Se especifica el barrido para motores de dos tiempos para entender su funcionamiento. Como este proyecto se centra en un MEC de dos tiempos experimental, las características geométricas son diferentes a las de un motor para un uso normalizado.

Estos conceptos previos se podrán aplicar en los siguientes apartados para hacer el estudio detallado del motor en cuestión.

2.2 Funcionamiento de un motor Diésel

Los motores de combustión interna alternativos se pueden diferenciar por su encendido, por chispa o por compresión, pero también por la forma de renovar la carga, motor de 2 tiempos o de 4 tiempos.

El motor encendido por compresión, más conocido como motor Diésel, llamado así por su creador, es un motor de combustión interna en el que la ignición del combustible que se inyecta en la cámara de combustión es causada por la temperatura elevada del aire en el cilindro debido a la compresión del fluido, una compresión adiabática. Los motores Diésel funcionan al comprimir solo aire, esto aumenta la temperatura del aire dentro del cilindro a tal punto que el combustible Diésel atomizado que se inyecta en la cámara de combustión se enciende. Por esto mismo, en el motor Diésel no es necesario el uso de bujías para el encendido del combustible como ocurre en los motores de gasolina.

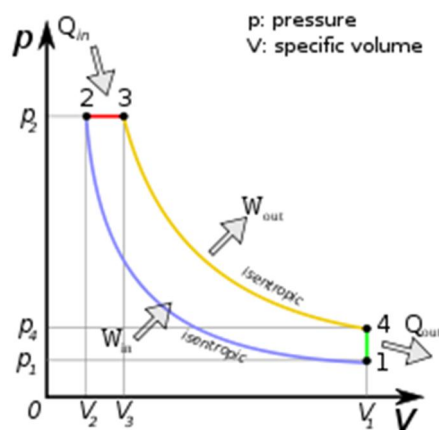


Figura 2.1: Gráfica P-V de un motor de encendido por compresión

El motor Diésel tiene el mayor rendimiento térmico de cualquier MCI, así como otros Motores de Combustión, como puede ser un turbofan de un avión comercial. Esto ocurre gracias a la alta relación entre la energía obtenida en la expansión y la energía que le aportamos con un combustible pobre como es el Diésel. Los motores Diésel de baja velocidad, como los utilizados en barcos, pueden tener un rendimiento térmico que sobrepasa el 50%.

Los motores Diésel pueden diseñarse como ciclos de dos o cuatro tiempos. En su origen, fue pensado para reemplazar, gracias a su eficiencia, motores de vapor estacionarios. Luego, su evolución fue hacia los submarinos y los barcos, ya que podían funcionar a muy bajas revoluciones y conseguir un gran par. Se siguió el uso en locomotoras, camiones, equipos pesados y plantas de generación de electricidad de menor medida. A partir de 1930 comenzaron a usarse en algunos modelos de automóviles y en 1970, su uso se extendió a automóviles de gran tamaño y todoterrenos. Actualmente, la media de la Unión Europea para los automóviles Diésel es del 50%, llegando en Francia a ser del 70%.

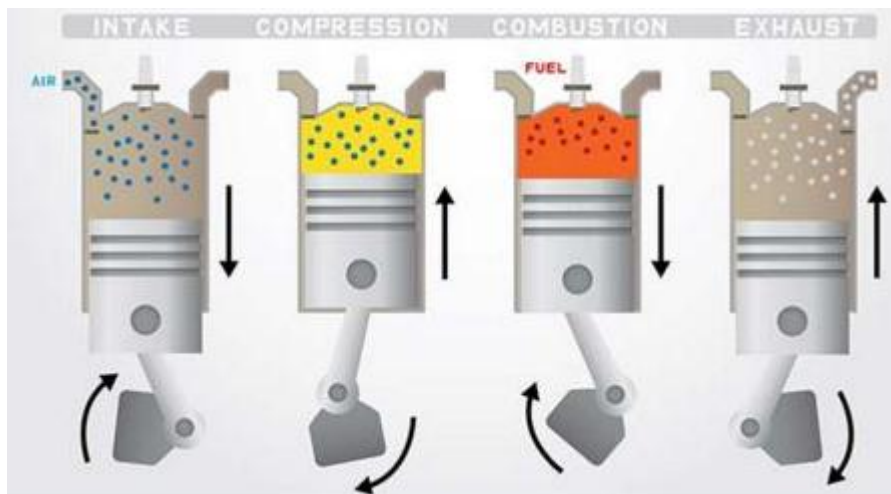


Figura 2.2: Representación del ciclo completo de un motor de 4T.

2.3 Funcionamiento de un motor de 2T

Un motor de dos tiempos (o dos tiempos) es un motor de combustión interna alternativo que completa un ciclo (admisión, compresión, explosión y escape) con dos carreras del pistón, es decir, con una sola revolución del cigüeñal. Es una idea diferente al motor de cuatro tiempos, que necesita cuatro carreras del pistón (dos hacia arriba y dos hacia abajo) y dos revoluciones del cigüeñal para completar un ciclo.

Los motores de 2T tienen una alta relación potencia-peso, pero esa potencia solo está disponible en un rango muy estrecho de velocidades de rotación llamada banda de potencia. En comparación al motor de 4T, el motor de 2T al carecer de muchos sistemas de admisión y de escape, le permite ser más compacto y a su vez, más liviano.

Seguidamente se hará una breve explicación de las fases del motor de dos tiempos de barrido por cárter, diseño bastante usual:

- Fase de admisión-compresión: la dirección del pistón en esta fase es hacia la culata, desde el PMI (lumbreras de admisión descubiertas) hasta el PMS. Mientras la cabeza del pistón realiza la compresión, la parte de debajo de este succiona el fluido de entrada a través de las lumbreras de admisión y lo almacena en el cárter.
- Fase de combustión-escape: al llegar la cabeza del pistón al PMS se finaliza la fase de admisión-compresión y se provoca la combustión por encendido o por compresión. El trabajo que provoca el encendido del combustible empuja el pistón hacia abajo y transmite esa fuerza al cigüeñal a partir de la biela. Antes de llegar al PMI abre las lumbreras de escape, donde los gases quemados saldrán empujados por los gases inquemados.

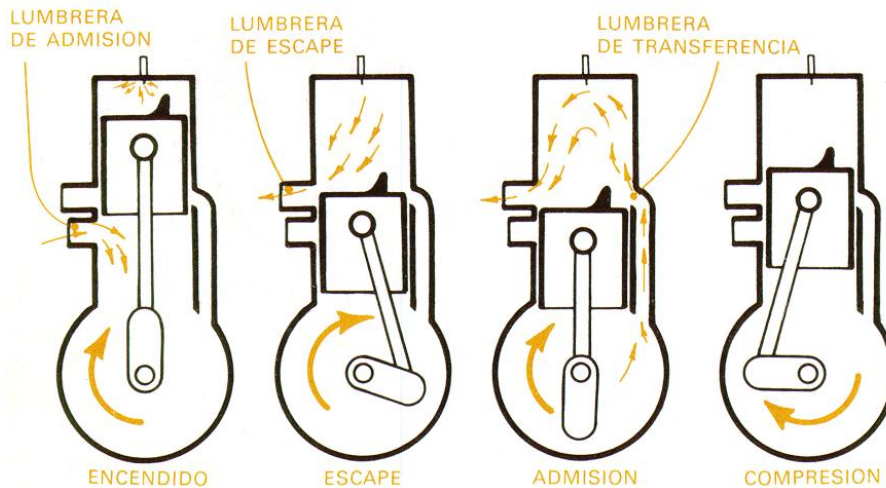


Figura 2.3: Representación de un ciclo completo de un motor de 2T

Barrido en motor de 2T

El barrido de un motor de 2T consiste en el intercambio de gases quemados a limpios. Mientras el pistón está subiendo y comprimiendo los gases, la lumbrera de admisión está abierta y entrando gases limpios. Cuando el pistón llega al puerto muerto superior (PMS) y llega la expansión, se abre la lumbrera de escape y una vía separada de la cámara de combustión, los gases limpios empujan a los gases quemados y se renueva la carga. Cuando está llegando al punto muerto inferior (PMI), se cierra la lumbrera de admisión y, en ese momento, casi todos los gases quemados ya se han evacuado.

El barrido de gases es un proceso muy delicado en el momento de elegir la opción más adecuada para un tipo de motor. El sistema condiciona la forma de la cabeza del pistón y la de la cámara de combustión, donde esta varía la dirección del fluido hacia el sistema de escape el

flujo que antes estaba entrando por la admisión. Después de estudios y experimentaciones sobre configuraciones del proceso de barrido, que la mejor forma de dirigir un flujo es adaptándolo a las paredes del cilindro y de la cámara de combustión, ya que cualquier flujo que se mueva libremente, será de gran dificultad dirigirlo donde el usuario quiera.

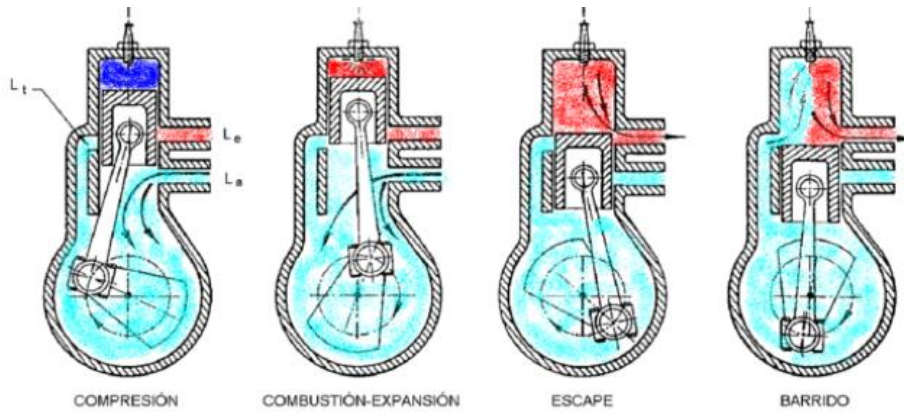


Figura 2.4: Representación del barrido de gases de un motor de 2T

2.4 Flujos internos de un MCIA

La comprensión de la circulación de los fluidos dentro de un cilindro de un MCIA es un estudio muy complejo por las diferencias geométricas de cada motor, donde cada modelo puede ser prácticamente único. Dicho flujo tiene las siguientes características:

- Flujo turbulento: se les llama así a los movimientos de un fluido que tiene una forma caótica, es decir, las partículas no se ordenan de ninguna forma específica y acaban creando pequeños remolinos repetitivos no coordinados. La trayectoria de una partícula es impredecible.
- Flujo tridimensional: el vector velocidad se constituye de tres coordenadas, normalmente en las tres direcciones ortogonales xyz y del tiempo. No hay simetría en todo el flujo.
- Flujo no estacionario: la velocidad de una partícula de fluido cambia con el tiempo y la posición.

Como se ha comentado antes, es de suma importancia el movimiento del fluido y el combustible y su mezcla, ya que dilución entre los dos provoca una combustión mejor. Esta mezcla se consigue a través de la turbulencia, cuanto más turbulencia haya, más se mezclarán los fluidos. Para ello, se modifica la geometría de las lumbreras de entrada y de salida del cilindro, como la geometría de la cabeza del pistón y de la cámara de combustión. Otro aspecto que ayuda a crear la turbulencia es la propia combustión, la cual empuja el fluido colindante a los quemados que se mezcla con el que está próximo a las paredes del pistón. Por último, elementos mecánicos del motor pueden provocar una turbulencia, como es el caso del inyector.

El mecanismo que genera más turbulencia es la inyección directa de combustible en el cilindro, ya que se inyecta a una velocidad muy superior al aire que está confinado en la cámara de combustión. Esa interacción forma vórtices turbulentos de mayor o menor medida. Las otras formas de generar esa turbulencia tan buscada no tienen tanta importancia. El flujo que entra por la admisión y que crea una turbulencia, probablemente no perdure después de la combustión. La turbulencia que se crea gracias a geometría es más importante que la mencionada anteriormente, ya que en el momento que el fluido de la cámara de combustión empieza a comprimirse y a ordenarse tiende a formar vórtices que ayudan a mezclar el fluido y el combustible.

Se pueden especificar tres tipos de movimiento del fluido dentro del cilindro, provocado por la geometría de la cabeza del cilindro y de la apertura y cierre de las válvulas de admisión y de escape. Esos movimientos son el *swirls*, *tumble* y *squish*.

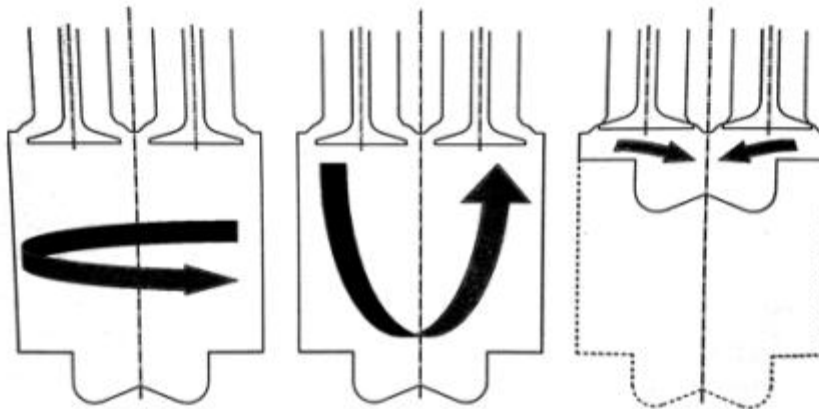


Figura 2.5: Representación de los diferentes movimientos flujos en el interior de un cilindro. Swirl, tumble y squish de izquierda a derecha respectivamente.

2.5 Geometría lumbreras de admisión y escape

La geometría de las lumbreras es de gran importancia para poder renovar la carga de una forma eficaz. Si no se hicieran adecuadamente, nos encontramos con gases ya quemados que no se han renovado y se han mezclado con los gases limpios que entran por la admisión. Esto provoca que empeore el rendimiento del proceso.

En la Figura [2.6] se pueden ver varias maneras de distribuir las lumbreras. En nuestro caso, el *motor maqueta* tiene la misma distribución que el segundo dibujo empezando por la izquierda que está representado. Esta distribución se llama lazo Curtis.

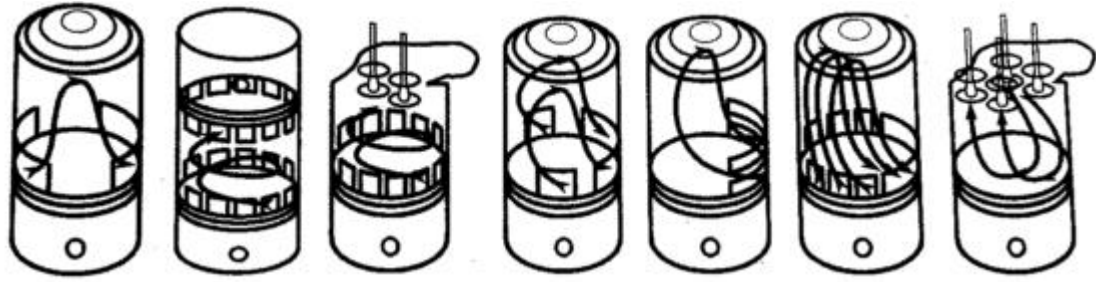


Figura 2.6: Tipos de barrido y posición y tamaño de las lumbreras de un motor de 2T. De izquierda a derecha: transversal, uniflujo con pistones opuestos, uniflujo con válvulas de escape en la cabeza, por lazo de Schnürle, por lazo Man, por lazo Curtis y en gancho con válvulas en cabeza

3. Modelado en Converge CFD

3.1 Introducción

Los programas CFD (*Computational Fluid Dynamics*) se basan en el estudio de flujo de fluidos con métodos numéricos y algoritmos. Puesto que la cantidad de cálculos y su dificultad puede ser muy elevada, es necesario el uso de ordenadores. Aun así, se requieren ordenadores de última generación, ya que el coste computacional es muy elevado.

Para poder rebajar la dificultad de las ecuaciones complejas se realiza una discretización. Para poder discretizar el estudio de un flujo, se tiene que crear en el espacio una “malla espacial”, dividiendo de esta manera una región en celdas con un pequeño volumen de control. Se calculan las variables de cada celda con ecuaciones lineales, mucho más sencillas que las ecuaciones complejas, pero a su vez, se requiere mucho tiempo, ya que pueden crearse una gran cantidad de celdas, del orden de millones, y calcularse las variables con un corto período de tiempo. Al final, se resuelve una matriz algebraica en cada celda de forma iterativa hasta que el residuo sea lo suficientemente pequeño.

Este estudio se basa en poder caracterizar un motor de pruebas del CMT. Para ello, se usa el programa Converge CFD para obtener un modelo teórico y compararlo con el modelo experimental que ya se obtuvo en su momento. Se tiene que aclarar que Converge CFD es un *software* de uso general. Este modelo representa el flujo que hay dentro de la cámara de combustión del motor maqueta; presión, campo de velocidades y densidad.

Para poder conseguir el modelo teórico con el Converge CFD es necesario seguir tres etapas fundamentales. Estas son: el preprocesado, el cálculo del modelo y el postprocesado. En los siguientes apartados se explica el funcionamiento del programa, como crea el “mallado espacial” y que cálculos realiza para llevar a cabo el estudio. En los anexos se hará hincapié en el preprocesado, el cálculo y el postprocesado que se ha seguido en este caso.

3.2 Mallado en Converge

Converge se diferencia del resto de programas CFD en la forma de mallar una región. El programa en cuestión genera de forma automática una malla ortogonal durante el cálculo del estudio, fundamentándose en las variables iniciales descritas por el usuario. El mallado de una geometría siempre ha sido una complicación bastante grande para los usuarios de programas CFD, ya que una mala elección dispara el coste computacional del estudio o hay un refinamiento muy pobre. De esta forma, el mallado en Converge se adapta a las posibilidades del computador y no como el resto de los *softwares*, que se adaptan a la geometría. La malla generada por Converge viene dada por unos sencillos parámetros de control definidos por el usuario.

A partir de la malla que genera Converge hay la opción de refinar ciertas zonas de la geometría donde se quieran unos resultados más exhaustivos. El refinamiento de una región busca crear muchas más celdas con un tamaño menor para garantizar una convergencia más precisa. Hay varias opciones para el refinamiento de una región:

- Escalado de base: permite refinar la malla base para toda la geometría el tiempo que el usuario desee durante la simulación.
- Los refinamientos fijos (*embedding*): son aquellos los cuales se basan en delimitar una zona que tendrá un refinado mayor que el resto de la geometría. El *embedding* es una zona también delimitada por el usuario en las zonas más críticas, mientras que en el resto deja un mallado más grueso, donde no es tan importante el resultado, para optimizar el coste computacional.
- El *AMR (Adaptative Mesh Refinement)*: Ajusta la malla automáticamente las zonas donde el flujo sea más crítico basándose en diferentes variables (velocidad, parcelas, temperatura, especies, pasivos o contornos). Puede ir variando durante la simulación para ahorrar coste computacional.

No se ha elegido la opción del AMR ya que provocaba un disparo del coste computacional del cálculo y no variaba prácticamente la resolución de los resultados. Por este motivo, como solamente nos importa la resolución en la cámara de combustión, se usa la opción del *embedding*, debido a que se puede elegir en que momento aplicar el refinamiento fijo y definir su resolución, de esta forma obtienes un buen procesado sin un coste computacional altísimo.

3.3 Condiciones de contorno

Es necesario especificar las condiciones de contorno e iniciales para que Converge inicie los cálculos. Estas condiciones determinan el estado de las variables en la frontera de la geometría de cálculo, así como en el instante inicial. Estas variables son la presión, la temperatura, entre otras. Se puede deducir que, si se especifica unas condiciones cercanas a la solución final, la convergencia será más rápida.

Converge diferencia ocho tipos diferentes de contorno, estos son: Periodic, Symmetry, Two_D, GT-Suite, Wall, Inflow, Outflow e Interface. En este trabajo se han empleado Inflow, Outflow y Wall.

3.4 Ecuaciones de la mecánica de fluidos

En este apartado se analizan las ecuaciones de la mecánica de fluidos que usa Converge para determinar el estado del flujo de la región. La dinámica de fluidos se guía por la conservación de masa, de momento y de energía.

- Ecuación de estado

Existen flujos compresibles e incompresibles y, las ecuaciones de la mecánica de fluidos lo contemplan para sus resoluciones. Para las ecuaciones de conservación de masa y de la cantidad de movimiento es necesario aplicar una ecuación de estado para relacionar las variables de temperatura, densidad y presión.

En Converge hay cuatro ecuaciones de estado disponible, estas son:

- Peng-Robinson: esta ecuación da unos resultados similares a la de Redlich-Kwong-Soave, aunque es mejor para predecir las densidades de muchos compuestos en fase líquida, sobre todo los apolares (átomos que poseen igual electronegatividad).
- Redlich-Kwong: es adecuada para calcular las propiedades de la fase gaseosa cuando el cociente entre la presión y la presión crítica es menor que la mitad del cociente entre la temperatura y la temperatura crítica. Pero, funciona mal con la fase líquida.
- Redlich-Kwong-Soave: tiene un comportamiento muy similar a la ecuación de Redlich-Kwong, pero llega a una resolución más decente los hidrocarburos.
- Ley de gases ideales: impone que las moléculas de gas no ocupan volumen. Además, considera nulas las fuerzas de atracción-repulsión entre las moléculas.

Las ecuaciones nombradas se pueden representar de una forma generalizada:

$$p = \frac{RT}{v - b} - \frac{a}{v^2 + uv + w} \quad (3.1)$$

Donde a , b , u y w son variables que dependen de la ecuación de estado que se haya elegido para el estudio.

En este estudio se realizará la ecuación de Redlich-Kwong, ya que no se hará inyección alguna durante el proceso y evitaremos cálculos en comparación a la ecuación de Redlich-Kwong-Soave.

Para llegar a resolver las ecuaciones comentadas anteriormente es necesario hacer un proceso de resolución iterativa y secuencial de las ecuaciones de continuidad, momento energía y especies. Primero, se dan unos valores iniciales a las incógnitas que se quieren resolver. Se resuelve la ecuación de momento y se procede a aplica un factor de corrección si no se llega a cumplir la ecuación de conservación de masa. Una vez se haya aplicado ese factor de corrección y se cumplan las ecuaciones de conservación de masa y cantidad de movimiento, se debe continuar resolviendo las ecuaciones de conservación de energía y especies.

- Ecuación de conservación de la masa y de la cantidad de movimiento.

Esta ecuación solo se puede aplicar para flujos compresibles. Las ecuaciones de conservación de masa y conservación de cantidad de movimiento son, respectivamente:

$$\frac{\rho v}{v} + \frac{\rho v^2}{v} = \rho \quad (3.2)$$

$$\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} = -\frac{\partial P}{\partial x} + \frac{\partial \tau_{yx}}{\partial x} + S_x \quad (3.3)$$

Donde el tensor de esfuerzos viscosos es:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \mu' \frac{\partial u_k}{\partial x_k} - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \quad (3.4)$$

A continuación, se muestra la definición de cada variable:

S : es el término fuente;

u : es la velocidad;

ρ : es la densidad;

μ : es la viscosidad;

μ' : es la viscosidad dilatacional;

P : es la presión;

δ_{ij} : es la Kronecker delta.

Además, si en el modelado está activa la turbulencia, la viscosidad se sustituirá automáticamente por la viscosidad turbulenta, esa es:

$$\mu_t = \mu + C_\mu \rho \frac{k^2}{\varepsilon} \quad (3.4)$$

donde:

C_μ : es una constante dependiente de la turbulencia;

ε : es la disipación turbulenta;

k : es la energía cinética turbulenta.

Converge brinda la opción de resolver las ecuaciones de conservación de masa y de la cantidad de movimiento de forma conservativa o no conservativa. Para este caso, se ha hecho uso de la forma conservativa.

Las ecuaciones de Navier-Stokes la componen las tres componentes espaciales particularizadas de las ecuaciones de conservación de masa y de la cantidad de movimiento.

- Ecuación de la conservación de la energía

Esta ecuación solo se puede aplicar para flujos compresibles. La ecuación de conservación de energía es:

$$\frac{\partial}{\partial t} \left(\rho c_p T \right) + \frac{\partial}{\partial x_j} \left(\rho c_p T u_j \right) = - \frac{\partial}{\partial x_j} \left(K \frac{\partial T}{\partial x_j} \right) + \sigma_{ij} \frac{\partial u_i}{\partial x_j} + \frac{\partial}{\partial x_j} \left(\rho \sum_m Y_m h_m \right) + \frac{\partial}{\partial x_j} \left(\rho D \frac{\partial Y_m}{\partial x_j} \right) + \dot{q}''' \quad (3.4)$$

Donde:

T : es la temperatura;

h_m : es la entalpía de las especies;

K : es la conductividad;

ϵ : es la energía interna específica;

D : es el coeficiente de difusión másica;

Y_m : es la fracción másica de la especie m .

Si la turbulencia está activa en el modelado, la conductividad se reemplaza por la conductividad turbulenta, esta es:

$$K_t = K + \mu_t \frac{\mu}{Pr_t} \quad (3.4)$$

Donde:

Pr_t : es el número de Prandtl;

μ_t : es la viscosidad turbulenta.

- Ecuación de la conservación de especies

$$\frac{\partial \rho m}{\partial t} + \frac{\partial}{\partial x_j} \left(\rho m u_j \right) = \frac{\partial}{\partial x_j} \left(\rho D \frac{\partial Y_m}{\partial x_j} \right) + \dot{m}''' \quad (3.4)$$

Donde:

u : es la velocidad;

ρ : es la densidad;

ρ_m : es la densidad de la especie;

Y_m : es la fracción de masa de la especie m ;

D : es el coeficiente de difusión de masa;

S_m : es el término fuente.

El coeficiente de difusión másica (D) se consigue con la siguiente fórmula:

$$D = \frac{\mu}{\rho_m} \quad (3.4)$$

La variable Sc es el número de Schmidt.

- Modelo de turbulencia

El modelo de turbulencia se puede usar para muchos tipos de estudio en Converge. Estos pueden ser como la combustión en un motor, un spray de un inyector, la aerodinámica de un vehículo o la transferencia de calor en unas paredes. Como en este estudio no se ha empleado la inyección de un combustible en la cámara ni una combustión en el ciclo del motor, no se ha considerado ni el modelo de turbulencia de la combustión ni la del spray. En consecuencia, solo se ha considerado la transferencia de calor en las paredes internas del motor.

3.5 Resolución de un flujo transitorio

Las ecuaciones descritas en el apartado anterior son discretizadas a partir de la malla que crea Converge. Discretizar las ecuaciones permite emplearlas como un sistema de ecuaciones, pero es necesario aplicar una técnica iterativa para resolver esos sistemas.

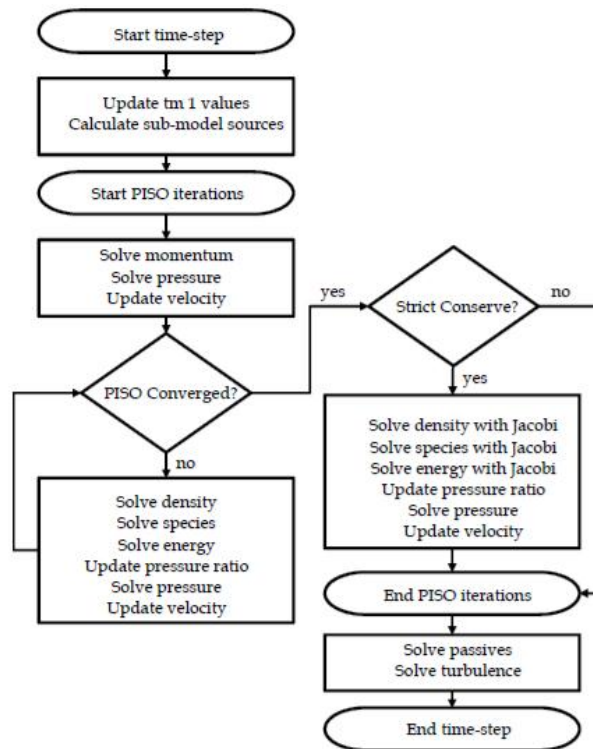


Figura 3.1: Pasos para la resolución de las ecuaciones en Converge

El método PISO (en inglés, *Pressure Implicit with Splitting of Operators*) resuelve la ecuación de Navier-Stokes. Es un procedimiento de cálculo de presión-velocidad que, en su origen, su objetivo fue el cálculo no iterativo del flujo compresible inestable, pero finalmente se ha adaptado con éxito a problemas de estado estacionario. PISO implica un paso predictor y dos pasos de corrección (iteración) y está diseñado para satisfacer la conservación de masa.

El algoritmo se puede resumir de la siguiente manera:

- Establecer las condiciones de contorno.
- Resolver la ecuación de momento discretizado para calcular un campo de velocidades intermedia,
- Computar los flujos de masa en las caras de las celdas del mallado.
- Resolver la ecuación de presión.
- Corregir los flujos de masa en las caras de la celda.
- Corregir las velocidades en función del nuevo campo de presiones.
- Actualizar las condiciones de contorno.
- Repetir desde el tercer paso el número de veces que el usuario elija.
- Aumentar el *time-step* y repetir desde el paso uno.

4. Software de post-procesado

4.1 Enight

Enight es un software para visualizar, analizar y comunicar datos de simulación o experimentos de ordenador.

Los principales mercados para Enight son la automotriz, la aeroespacial, la defensa, la producción de energía, la fabricación de alta tecnología y la combustión, en nuestro caso. Hay otros mercados que usan Enight donde se requiere una muy alta precisión de modelado de física basada en el ordenador. Debido a su potencia y flexibilidad, una de las características de Enight es que proporciona una gran cantidad de técnicas de visualización.

Se usa en mayor frecuencia para la dinámica de fluidos computacional (CFD), mecánica estructural computacional (CSM) y otros procesos de ingeniería asistida por ordenador (CAE).

El propósito del siguiente apartado es hacer una descripción general del sistema Enight y su documentación. En el anexo se explicarán los pasos que se han seguido para obtener los planos necesarios para visualizar el cilindro y la cámara de combustión del modelo actual y como se han obtenido las imágenes y vídeos para su posterior análisis.

4.1.1 Geometría en Enight

El procesamiento de Enight comienza cuando se le proporciona un modelo hecho anteriormente con otro software. Este modelo se suele agrupar por partes, tomando ejemplo del modelo de este proyecto, cada parte podría ser el cilindro, las lumbreras de escape o admisión, entre otros. Enight se encarga de asociar toda la información que se ha obtenido del cálculo en Converge a cada parte del modelo. A partir de aquí, cada acción que se haga en Enight se aplica a cada división del modelo para facilitar su estudio.

Cada parte del modelo consta de sus nodos y elementos, donde los elementos son el conjunto de nodos conectados en una forma geométrica cualquiera. Los archivos de datos procesados para que compatibilicen con Enight proporcionan variables en cada nodo, en el centro de los elementos o en ambos.

4.1.2 Visualización de la geometría

Las partes del modelo están visualizadas en la ventana de gráficos de Enight. Esta ventana deja varias opciones para modificar la visualización del modelo y realizar más fácilmente un estudio de este. Seguidamente, se nombrará las diferentes opciones de visualización que proporciona Enight.

- *Hidden lines and shaded surfaces*: permite sombrear superficies u ocultar líneas para obtener vistas más realistas del modelo.

- *Clipping*: Ensigna da la opción de cortar planos al antojo del usuario. Es realmente útil en el campo de la combustión, ya que, gracias a esta opción, el usuario es capaz de observar que ocurre en el interior de un cilindro o de la cámara de combustión.
- *Annotations*: estas anotaciones se basan en la representación a través de líneas, flechas, etiquetas o mapas de colores con leyenda los valores de las variables que se han querido estudiar, como puede ser la velocidad, la temperatura, entre otras.
- Otras opciones son la elección de color del fondo de pantalla, la elección de la vista de la representación del modelo o el formato del guardado de la imagen o vídeo.

4.1.2.1 Clips

Un *clip* (en castellano, recorte) es un plano, línea, caja, superficie *ijk*, plano *xyz*, superficie *rtz*, superficie cuádrica (cilindro, esfera, cono, etc.) o superficie de revolución que pasa a través de piezas especificadas. Si el recorte es un plano, este puede ser finito o de una naturaleza infinita, dependiendo de la elección del usuario. Los planos finitos se usan para cortar una parte o pieza del modelo en concreto, sin tener en cuenta el resto del modelo. En cambio, el plano infinito, se usa para cortar todo el modelo por una zona y ver el interior de todas las partes. El cliente controla la ubicación de los diversos planos usando el plano *xyz* o sus planos perpendiculares. También se puede usar la opción de escribir directamente las coordenadas donde el usuario quiere su recorte.

Las superficies de los recortes creados se pueden animar de diferentes formas eligiendo la variable que se quiere representar en él. Esta opción resulta muy útil para poder observar con facilidad la evolución de una variable en el interior de una parte, que vista desde fuera no se puede percibir diferencias.

4.2 Matlab

Matlab (abreviatura de Matrix Laboratory) es una herramienta de software matemático con un lenguaje de alto nivel propio (lenguaje M) que permite el cálculo numérico. Matlab permite manipulaciones de matrices, creación de interfaces de usuario, trazado de funciones y datos, implementación de algoritmos y adaptabilidad con programas en otros idiomas como C, C#, C++, Python, Fortran y Java.

Gran parte del trabajo de post-procesado se ha realizado con Matlab, ya que, de una forma generalmente sencilla, se crean unas rutinas en el lenguaje M que tratan todos los datos numéricos obtenidos tras el cálculo de un modelo y los convierte a resultados gráficos muy intuitivos y fáciles de leer. En el caso de este estudio, después del cálculo de las variables que se obtienen en Converge (como la velocidad, la presión, la turbulencia, entre otras), el usuario a través del código que él ha escrito procesa los datos y obtiene gráficas de diferentes tipos para leer de la mejor forma posible los resultados de cada variable. Además, Matlab es capaz con una simple rutina, juntar imágenes para formar un vídeo o, a partir de un vídeo, formar una secuencia de imágenes.

En el anexo se hará hincapié en las rutinas utilizadas en el post-procesado de este modelo y los cambios hechos para reducir el coste computacional de los cálculos en

comparación a proyectos anteriores. En el capítulo 5, se analizará y valorará los resultados del post-procesado.

5. Análisis de resultados

5.1 Introducción

El principal objeto de este proyecto es lograr un modelo teórico afín al modelo experimental obtenido a través de la técnica PIV. Para ello es necesario analizar el interior del cilindro del *motor maqueta* y la dinámica del fluido que hay en su interior y compararlos con los datos que ya tenemos.

Como en trabajos anteriores al presente ya se hizo una puesta a punto del problema, se han obviado las simulaciones en ciclo cerrado. Esto ha ocurrido porque en el ciclo cerrado no se contempla los sistemas de admisión y escape. En cambio, en la geometría actual solo se ha modificado esos sistemas. De esta forma resulta más fácil modificar la malla hasta encontrar una buena relación resolución-coste computacional.

Se ha elegido el mismo tamaño de malla base que los proyectos anteriores ya que se quería hacer una comparación con estos mismos y si se usa una malla diferente, no tendría un contraste fiable.

A lo largo de este capítulo se van a observar los resultados alcanzados en el actual proyecto y su comparación con el TFM de María José Ruiz Lucas, “Modelado CFD del movimiento del aire en el cilindro de un motor de dos tiempos con accesos ópticos” y el TFG de P. Fernández Bérjon. “Caracterización del flujo interno en la cámara de combustión de un motor óptico mediante velocimetría por imágenes de partículas”. Finalmente, se comparan los campos de velocidades del método experimental hecho con la técnica PIV y del actual modelo teórico realizado con la técnica CFD.

5.2 Descripción del *motor maqueta*

El *motor maqueta* es una instalación experimental que se ubica en el DMMT de la Universitat Politècnica de Valencia. Esta instalación se constituye por un motor térmico y los equipos y sensores esenciales para representar lo que ocurre dentro del motor.

Este motor se comporta como un motor Diésel de dos tiempos de inyección directa. Este comportamiento viene explicado en el primer apartado de la memoria.

No es de vital importancia conocer los efectos de que la inyección se haga de forma directa, ya que en el actual proyecto no se estudia ni la inyección de combustible ni la combustión. El motor ha sido alterado para que cuando se desee el motor no produzca trabajo. Es decir, si en ningún momento se hace una inyección de combustible y no hay una combustión, no se produce trabajo, esto se consigue con el uso de nitrógeno como aire. El nitrógeno tiene unas propiedades muy parecidas al aire atmosférico que se usa en todos los motores. Con esta modificación del *motor maqueta* se pueden imitar condiciones inertes (nitrógeno) y reactivas (aire).

En este estudio se hace uso de nitrógeno, puesto que solo se quiere observar el movimiento del fluido en el interior de la cámara de combustión y del cilindro, sin tener en

cuenta la combustión. Para ello, es necesaria la instalación de un motor eléctrico que haga girar el cigüeñal para poder hacer un estudio fluidodinámico del interior del cilindro. Este motor eléctrico mantiene el cigüeñal a una velocidad constante de 500 rpm.

En la Tabla [5.1] se muestra la geometría del *motor maqueta*:

Diámetro del pistón [mm]	150
Carrera [mm]	170
Carrera efectiva [mm]	108
Cilindrada [cm ³]	3000
Relación de compresión	15.8:1
Distancia entre el inyector y la cabeza del pistón en PMS [mm]	87.8

Tabla 5.1: Características del motor maqueta

Una variación muy destacable en comparación a un motor comercial es que se ha aumentado el tamaño diez veces las dimensiones de la cámara de combustión respecto a un motor real de cilindrada parecida, para facilitar el estudio del flujo en la misma. Además, se ha sustituido parte de las paredes de la cámara de combustión por paredes transparentes, para tener la posibilidad de ejecutar la técnica PIV.

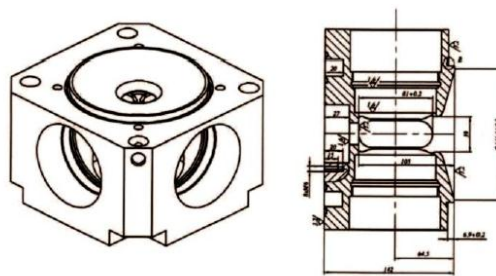


Figura 5.1: Medidas de la cámara de combustión modificada

A continuación, se muestran las dimensiones de la cámara de combustión y de las ventanas ópticas:

- Cámara de combustión: diámetro 50 mm y altura 81 mm.
- Ventanas ópticas: ancho de 28 mm y altura de 81 mm.
- Garganta: diámetro de 39 mm

El sistema de admisión consta de ocho lumbreras, en cambio, el sistema de escape tiene tres lumbreras. Estas lumbreras están colocadas simétricamente, ya que en cada lado del

cilindro hay cuatro lumbreras de admisión y una lumbrera y media de escape. Las lumbreras de escape tienen unas dimensiones mayores que las lumbreras de admisión. Esto provoca que cuando el pistón baja, se descubra primero el escape. Por otra parte, cuando el pistón está subiendo, las últimas lumbreras en esconderse son las de escape. Esta situación conlleva la formación de turbulencias tipo *tumble*, turbulencias típicas en motores de encendido provocado.

En el proyecto que aplicaron la técnica PIV en el *motor maqueta* para averiguar su funcionamiento usaron varios planos para después poder hacer comparaciones con modelos teóricos con más facilidad. Estos planos son el L (izquierdo), C (central), R (derecho) y ortogonalmente a estos, los planos F (frontal), CC (central) y B (trasero).

En la Figura [5.2] se muestra de forma detallada la posición de cada plano de la cámara de combustión:

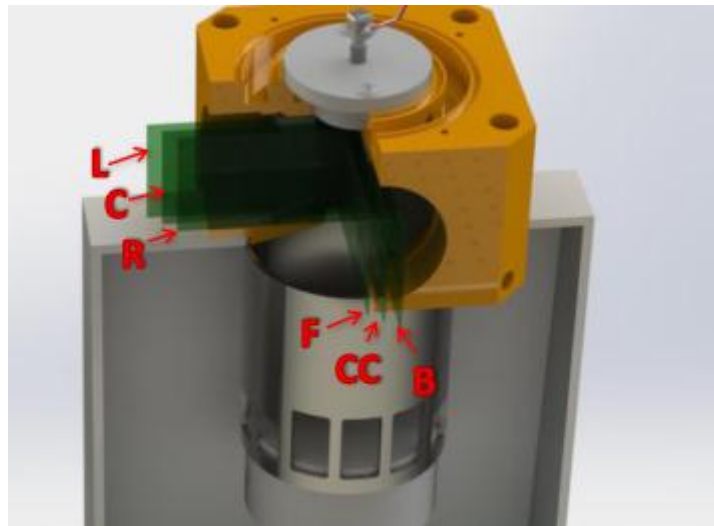


Figura 5.2: Vista de los planos usados en la técnica PIV

Hay varias técnicas para medir el campo de velocidades de un flujo dado. Entre ellas están el PIV, el láser Doppler y el anemómetro de hilo caliente. El PIV se centra en la medida simultánea con una buena determinación en el espacio, trabajando en el mismo plano. Por otra parte, el láser Doppler y el anemómetro de hilo caliente están mejor preparados para la medición de la velocidad en un solo punto, pero con una resolución temporal más decente que el PIV. El objetivo del estudio implica conocer el campo de velocidades de todos los puntos del mismo plano de la cámara de combustión para observar un patrón, por ese motivo es necesario usar la técnica PIV.

El sistema de velocimetría de partículas (en inglés *Particle Image Velocimetry*) es un método óptico que se usa para conocer la velocidad instantánea real y las propiedades de un flujo. El flujo se dopa con unas partículas, que seguirán la dirección del flujo en cuestión. Si no se acierta con el tamaño de las partículas no seguirán la dirección de la velocidad del fluido, este tamaño vendrá dado por el número de Stokes. Seguidamente, el fluido es iluminado por un haz de luz para que las partículas la reflejan y sean visibles. Cuando el haz está activo se realizan una serie de capturas de imagen para unir las y formar una reproducción del campo de velocidades

del flujo. Se toman dos fotogramas casi consecutivos a partir de los cuales se calcula el módulo y la dirección de la velocidad viendo cuánto se ha movido una de las partículas que se observan en la imagen.

5.3 Configuración de las simulaciones

5.3.1 Mallado

Se ha determinado la misma malla que se utilizó en proyectos anteriores al actual. La malla base es de 2 mm de tamaño de celda sin aplicar AMR en ninguna zona. Tiene un límite de celdas no superior a 1.5 millones, porque solo necesitamos precisión en la cámara de combustión y no se quiere que se dispare el coste computacional.

La malla base se eligió en base a varias estrategias, donde se buscaba la mejor malla resolución/coste computacional. Las mallas bases que se trataron fueron de 3, 2 y 1.5 mm. La malla base de 3 mm tenía muy poca resolución, aunque un coste computacional muy bajo. Por otra parte, la malla base de 1.5 mm las tendencias eran muy parecidas a la de 2 mm, por esa razón se eligió la malla base de 2 mm, porque con la misma resolución el tiempo de cálculo dura cuatro veces menos.

Para obtener una mayor resolución en la cámara de combustión se usa un *embedding* de nivel 2 fijo en esa zona. Ese *embedding* se aplica entre -30 y 30 grados, porque es el intervalo cercano al PMS, donde se centra el estudio.

5.3.2 Condiciones iniciales

Las condiciones iniciales se han definido para cada región: *Cylinder* (Cilindro), *Intake* (Admisión) y *Outflow* (Escape). Todos estos valores son datos experimentales que se obtuvieron a través de la técnica PIV en un proyecto hecho por un alumno en el pasado. Como empíricamente es más complicado adquirir el valor de la turbulencia, la alumna que realizó este mismo proyecto, pero con una geometría mucho más simple, hizo una aproximación para obtener los valores de T_{ke} y E_{ps} . En este trabajo solo se ha tenido que recopilar los datos de los anteriores estudios y aplicarlos a la nueva geometría más realista.

En la Tabla [5.2] se detallan los valores de cada variable:

Variable	Región		
	<i>Cylinder</i>	<i>Intake</i>	<i>Exhaust</i>
Presión [Pa]	149127	140000	136600
Velocidad [m/s]	0	0	0
Temperatura [K]	304.74	328.35	311.15
Tke [m ² /s ²]	1	1	1
Eps [m ² /s ³]	100	100	100
Especies	1	1	1

Tabla 5.2: Condiciones iniciales en cada región del modelo

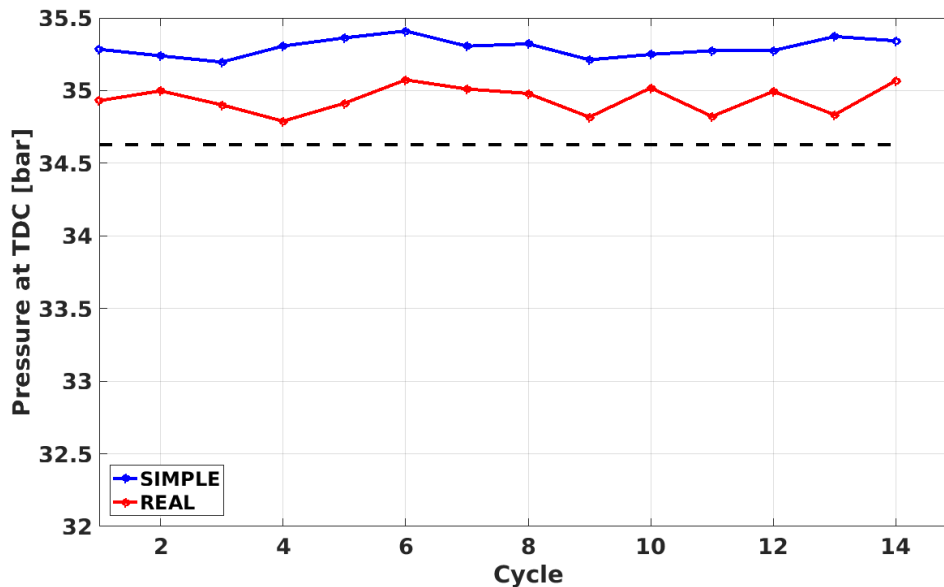
5.3.3 Número de ciclos

Una característica de las simulaciones de ciclos de un motor de combustión es que necesitan un número mínimo de ciclos para que los resultados se establezcan y sigan una tendencia. En los proyectos anteriores al actual se hicieron pocos ciclos y no se comprobó cuánto tardaba en estabilizarse el flujo. El último proyecto que se hizo se amplió el número de ciclos a quince. El primer ciclo no procede, debido a que cuando empieza la simulación la solución está inicializada a unos valores homogéneos de variables como velocidad y temperatura, que hasta que no se calcula el primer ciclo no se modifican sustancialmente. Es por ello que el primer ciclo se considera más bien de inicialización.

Se ha elegido calcular quince ciclos, catorce para el estudio para poder comparar de una forma idónea con el trabajo anterior. Es imprescindible elegir el número de ciclos que se van a calcular, ya que en este proyecto conllevó mucho tiempo el proceso de cálculo, aproximadamente cada ciclo podía tardar unas 36 horas.

5.4 Presión en cámara

En la Gráfica [5.1] se muestra la comparación entre el modelo PIV (línea discontinua), el modelo CFD antiguo (línea azul) y el modelo CFD actual (línea roja):



Gráfica 5.1: Comparación de la presión ciclo a ciclo entre el modelo actual (rojo), el modelo antiguo (azul) y el modelo PIV (negro discontinuo)

Viendo la comparación de la presión en la cámara de combustión en el punto muerto superior se puede percibir que la presión entre el modelo experimental (negro), el modelo teórico anterior (azul) y el modelo teórico actual (rojo) es ligeramente diferente.

El modelo teórico actual se ha acercado más a la realidad que el modelo anterior, ya que la presión en cámara está un poco por debajo. Aun así, no se ha conseguido la misma presión que en el modelo real, debido a que este tiene la presión un poco más baja.

5.5 Campo de velocidades

En este apartado se valora los resultados del campo de velocidades en la cámara de combustión del modelo CFD actual en comparación al campo de velocidades empírico medido con la técnica PIV. Todos los planos vistos en los siguientes apartados son del plano CC. También se han hecho representaciones de las variables desde el plano C, pero no ayudó a la comprensión del problema. Por esa razón, no se muestra en este proyecto y solo se trabaja sobre el plano CC, ya que es el mismo plano sobre el que se ha procedido en los anteriores estudios.

Se ha comparado el modelo CFD con los experimentos en el intervalo entre del -10 al 15 CAD (grados de cigüeñal). En los experimentos se ha hecho la media de todos los ciclos en cada ángulo de cigüeñal, habiendo promediado 75 ciclos para el grado -10, otros 75 ciclos para el grado -9, etc. Por otro lado, en el modelo CFD, se ha hecho una media progresiva. Esa media

progresiva consiste en mostrar la media del ciclo uno y dos en la casilla del ciclo dos, luego mostrar la media del ciclo uno, dos y tres en la casilla del ciclo tres y así sucesivamente hasta llegar a la media de todos los ciclos en la casilla del ciclo catorce. Por esta razón, la representación más importante para analizar es la del ciclo catorce.

Los mapas de velocidades muestran líneas de corriente que indican la dirección de la velocidad en cada posición, sobre unos contornos cuyo color indican cuál es el módulo de dicha velocidad. El color rojo oscuro es la velocidad máxima que puede alcanzar el fluido, unos 25 m/s y el azul oscuro la mínima velocidad, cercana a 0 m/s.

En las siguientes figuras se observa la comparativa entre el plano CC del PIV y el plano CC de la media ciclo a ciclo actual en dos posiciones angulares, la Figura [5.3] a -10 CAD, la Figura [5.4] a +10 CAD:

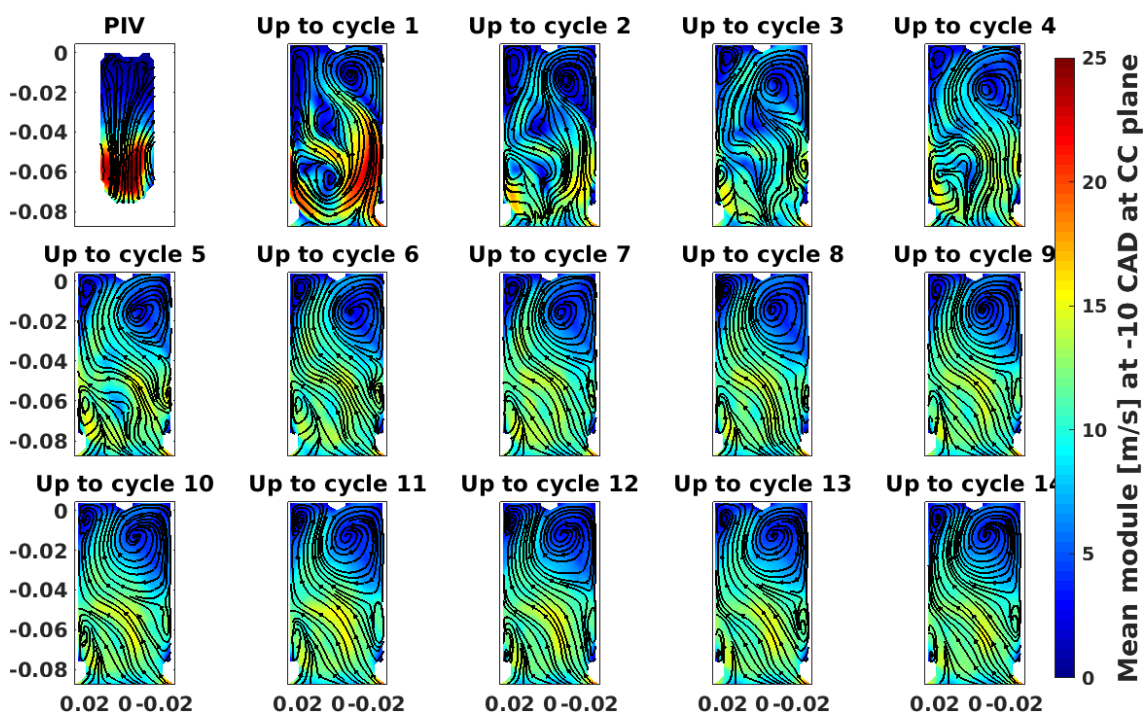


Figura 5.3: Campo de velocidades promediadas en el plano CC en -10 CAD

Se puede observar que en el caso expuesto arriba hay una tendencia a estabilizarse a partir de los ciclos siete y ocho. Esta tendencia se aplica a todos los ángulos del cigüeñal graficados. Gracias a esta información, se puede confirmar que a partir del ciclo ocho no sería necesario seguir calculando ciclos, debido a que después del ciclo ocho se obtendrían los mismos resultados con diferencias insignificantes. La relación resolución-coste computacional no saldría rentable.

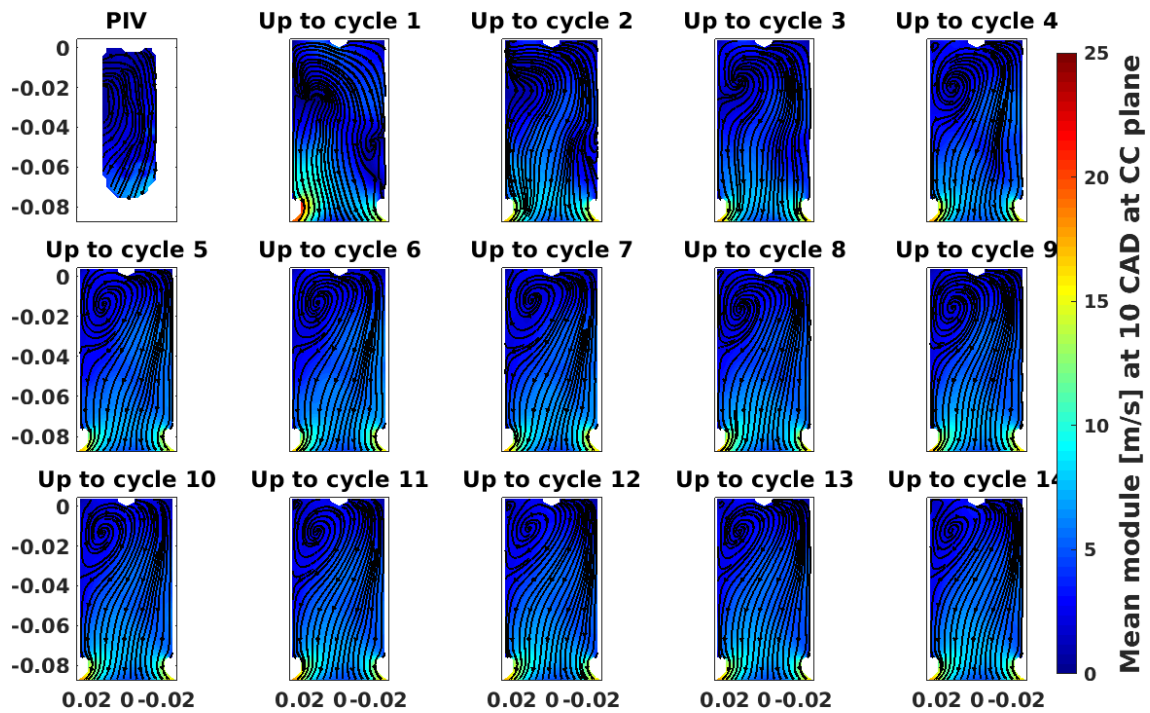


Figura 5.4: Campo de velocidades promediadas en el plano CC en +10 CAD

Después de analizar con detenimiento el campo de velocidades del modelo empírico con el modelo de este estudio no se puede decir que los dos modelos sean lo suficientemente parecidos para poder usar el modelo CFD para proyectos futuros. Una de las razones por la cual no puede haber surgido un modelo correcto es por la forma de cálculo de Converge CFD. Converge CFD tiene un procedimiento de cálculo muy sensible y si cualquier variable se ha podido desviar unos decimales del resultado correcto, es muy fácil que cambie la estructura del flujo entero. Esto es fácil de comprobar en los campos de velocidades de los ciclos sin promediar, donde se puede comprobar que en cada uno de ellos el flujo cambia su vórtice de lado de la cámara de combustión o que las zonas con más velocidad van variando.

En la Figura [5.5] y la Figura [5.6] se muestran los campos de velocidades de los ciclos sin promediar en los instantes en los instantes -10 y +8 CAD:

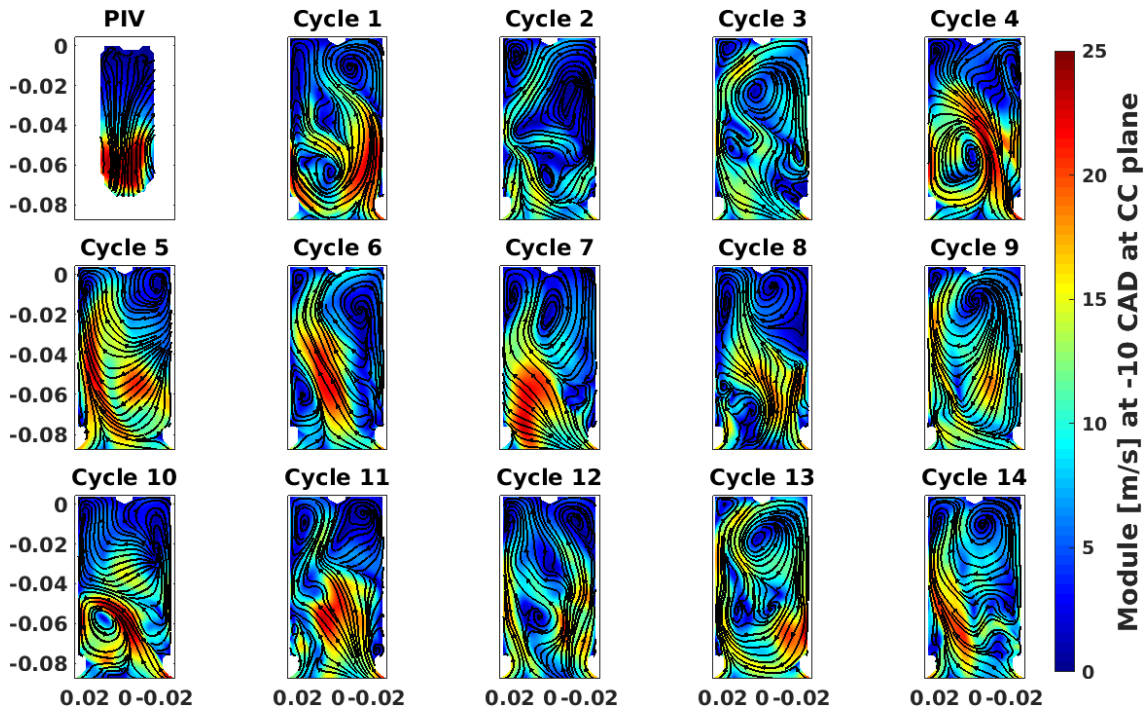


Figura 5.5: Campo de velocidades ciclo a ciclo en el plano CC en -10 CAD

Por otra parte, solo a partir de la expansión, después de superar el punto muerto superior y que el aire del interior de la cámara de combustión empiece a vaciarse, sobre el 8 CAD, se puede ver una afinidad entre los dos modelos. Esto puede llegar a ocurrir porque es bastante más sencillo prever el movimiento de un flujo vaciando un recinto en comparación a un fluido llenándolo.

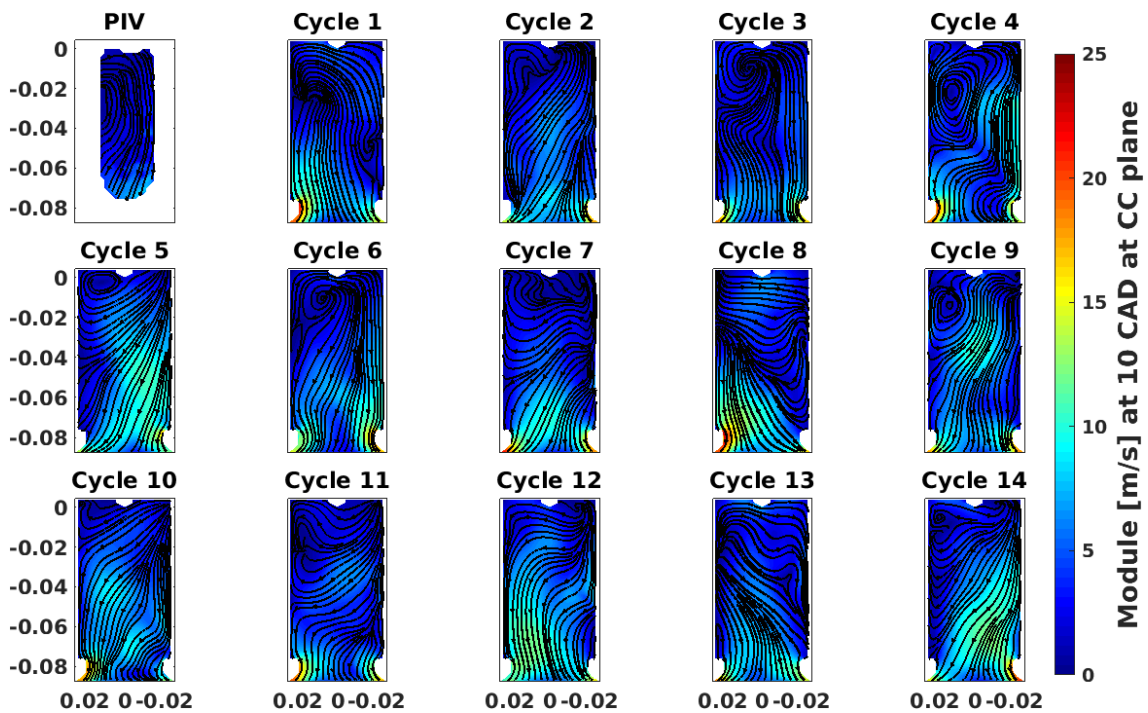


Figura 5.6: Campo de velocidades promediadas en el plano CC en +10 CAD

Esto puede inducir a que se piense que la diferencia entre los dos modelos este en el intervalo de tiempo de la compresión. Por la geometría de las lumbreras, durante el recorrido de expansión/escape del pistón, provoca que cuando las lumbreras de escape se vean haya más presión fuera del cilindro que dentro. Eso hace que se introduzcan gases del exterior por el escape. Más tarde, cuando se descubren las lumbreras de admisión, empujan estos gases que han entrado hacia el exterior hasta que el pistón vuelve a tapar admisión y escape.

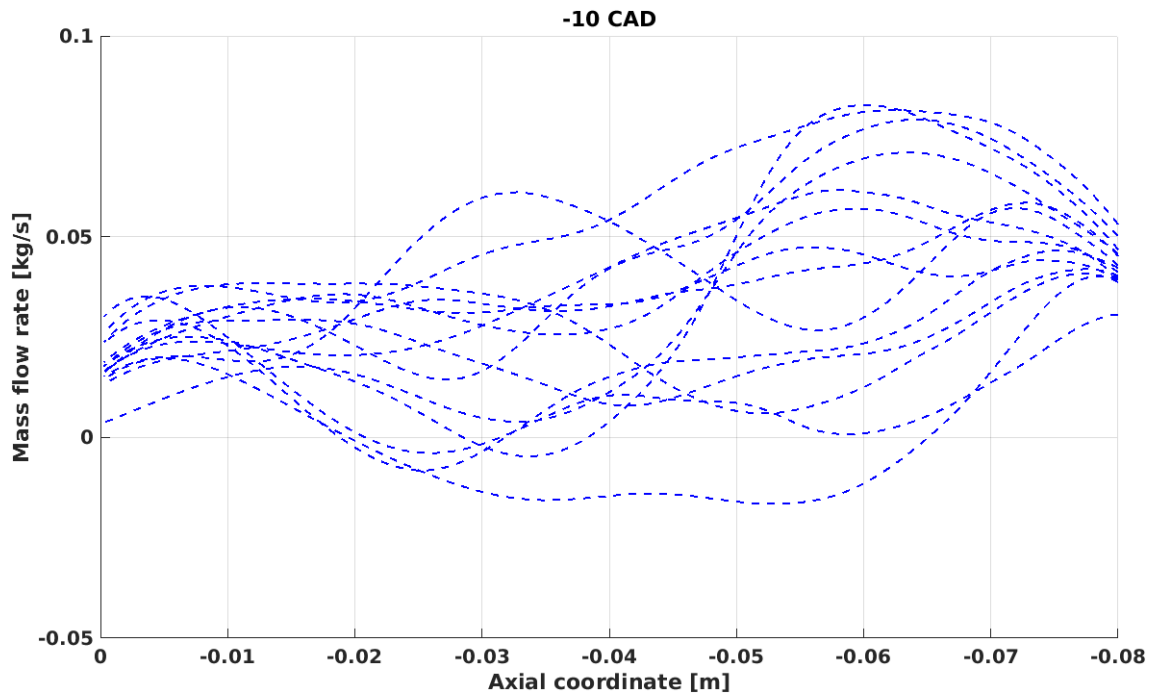


Figura 5.7: Gasto másico a lo largo de la cámara de combustión en -10 CAD

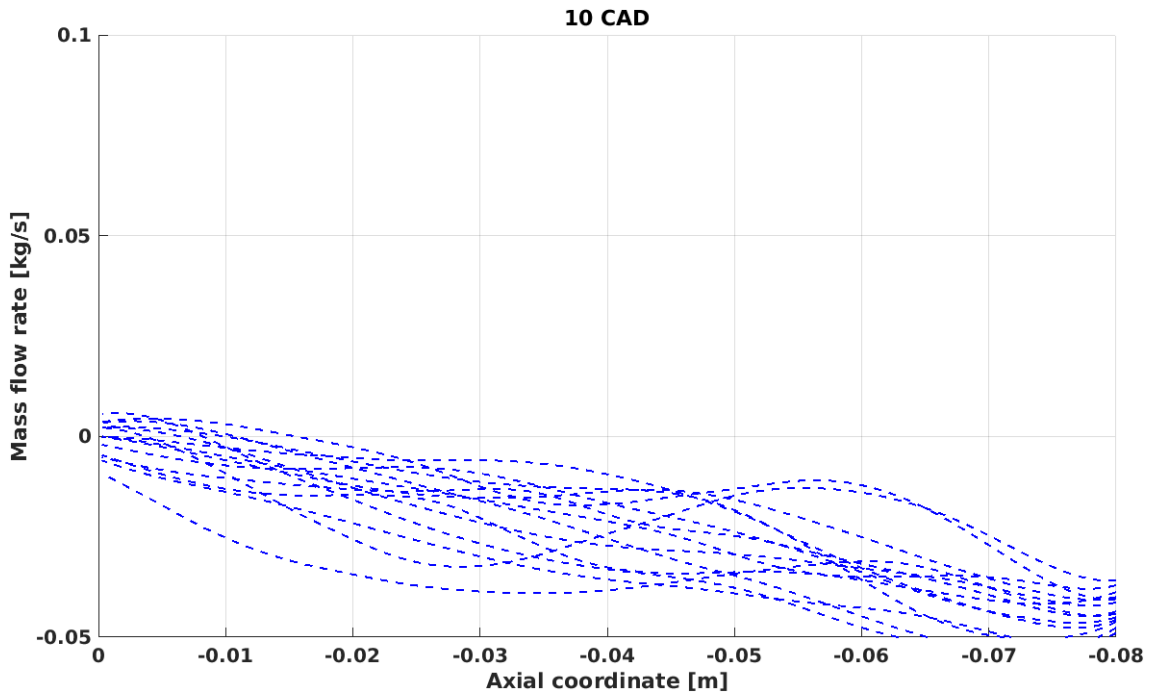


Figura 5.8: Gasto másico a lo largo de la cámara de combustión en +10 CAD

Este movimiento complica mucho el estudio del fluido de compresión, debido a que cuando el pistón está comprimiendo el fluido está muy desordenado dentro del cilindro y, cuando este tiene que entrar a la cámara de combustión por la garganta, lo hace de una manera muy caótica.

Time = 1.65339

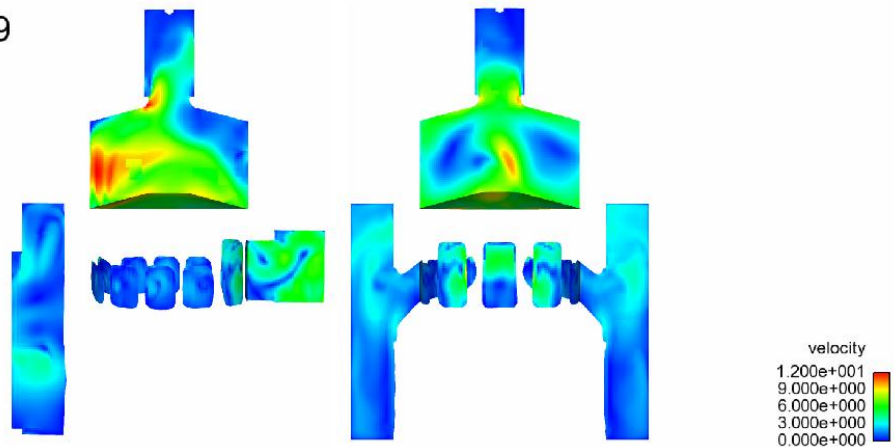


Figura 5.9: Representación del motor maqueta en EnSight en la carrera de compresión

Un comportamiento curioso del fluido es que, desde el -10 CAD del modelo actual se puede observar un vórtice en la zona de arriba a la derecha de la cámara de combustión.

Mientras avanza la simulación se ve como ese vórtice avanza de derecha a izquierda de la cámara. Hasta que el pistón no llega a su PMS, el fluido entra dentro de la cámara de combustión a una velocidad considerable. Esa velocidad se va reduciendo desde que el fluido entra en la cámara hasta que llega al centro del vórtice. En cambio, después del PMS ocurre todo lo contrario. Sobre los 2 y 3 CAD el vórtice se coloca en medio de las representaciones. En ese momento, el flujo del vórtice sale de dentro hacia fuera, cuando antes del punto muerto superior. Esto provoca que el fluido empiece a acelerarse desde el centro del vórtice hasta su extremo y del extremo del vórtice hasta fuera de la cámara de combustión.

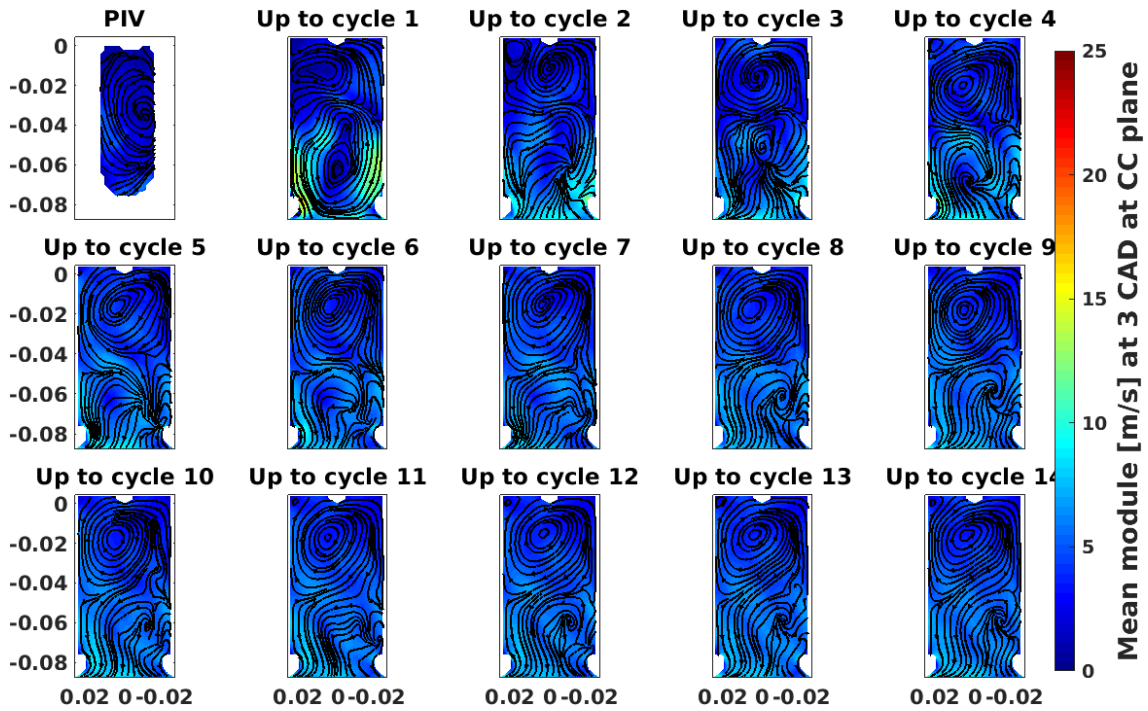


Figura 5.10: Campo de velocidades promediadas en el plano CC en 3 CAD

En las representaciones del caso experimental se puede percibir el comienzo de un vórtice en la zona derecha de la cámara de combustión. Ese vórtice no se genera hasta ya sobrepasado el punto muerto superior, aproximadamente 3 CAD. Desde ese punto se observa que el vórtice del modelo actual que se forma desde el -10 CAD es el mismo que el vórtice generado en 3 CAD en los experimentos.

Otro punto importante a tener en cuenta es el valor del módulo de la velocidad. Hasta que el fluido no se frena llegado el pistón al PMS hay una clara diferencia en el módulo de la velocidad entre los dos modelos. Las gráficas del -10 CAD muestran una diferencia de prácticamente 10 m/s en la entrada de aire entre los experimentos y el cálculo CFD, siendo la velocidad del primero de 25m/s y la del segundo de 15 m/s. Esta diferencia se va reduciendo durante el paso del ángulo del cigüeñal, hasta que en el punto muerto superior ya se asemejan sus valores a unos 10 m/s.

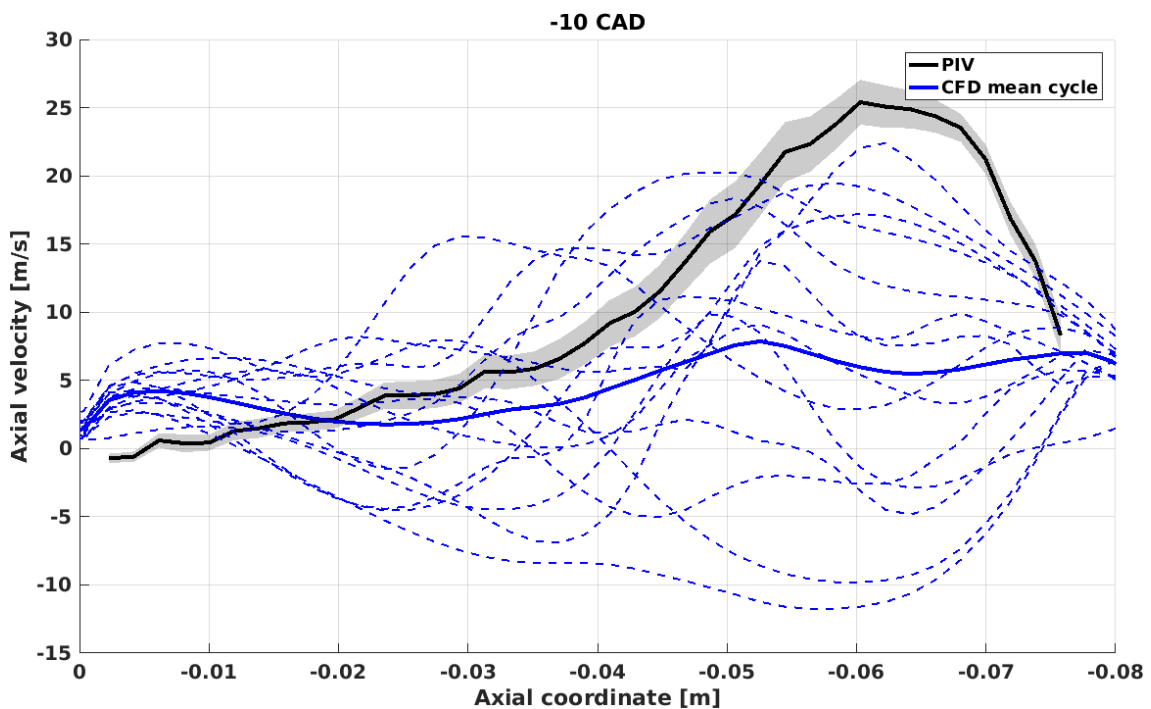
5.5.1 Velocidades axiales

A continuación, se muestran diferentes representaciones de la velocidad axial en el centro de la cámara de combustión a diferentes alturas.

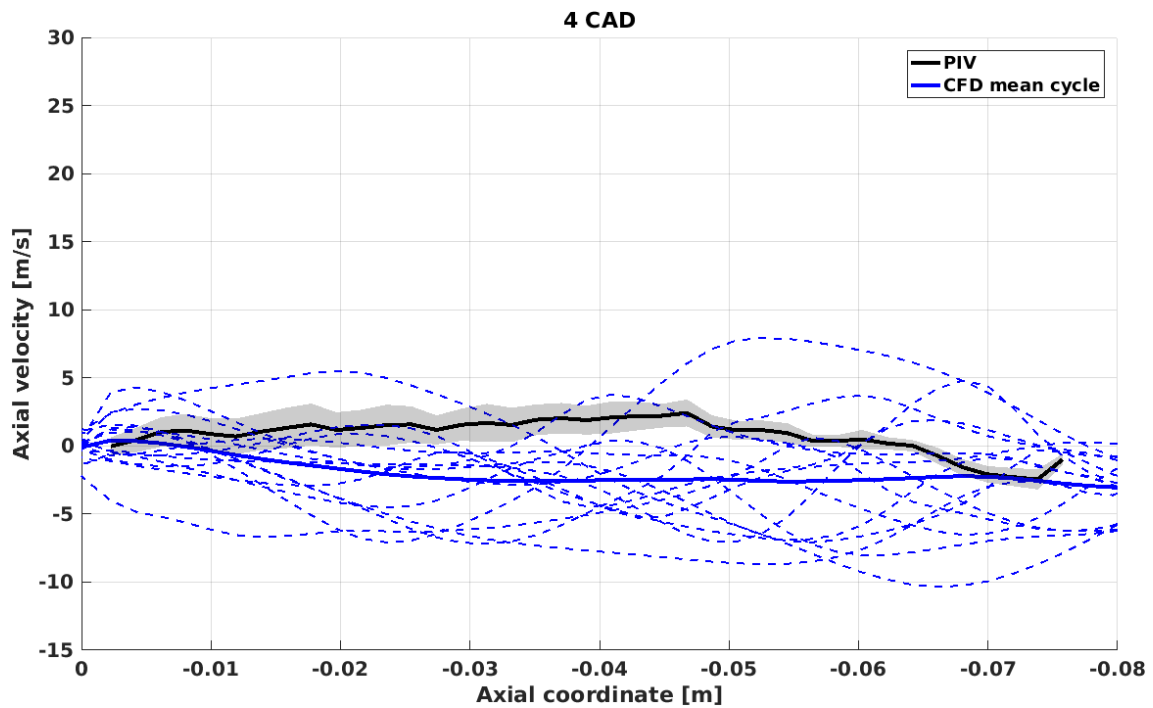
La línea gruesa negra representa la media de todos los ciclos en cada ángulo de cigüeñal, la línea gruesa azul muestra la media de todos los ciclos en cada ángulo de cigüeñal y las líneas azules discontinuas es la velocidad axial de cada ciclo por separado.

La coordenada axial cero corresponde a la posición del inyector, en la parte superior de la cámara de combustión y la coordenada axial -0.8 es el final de la cámara de combustión y principio de la garganta.

Las gráficas nos muestran unos datos parecidos a los obtenidos con la comparación de los campos de velocidades. Al principio, en -10 CAD, los dos modelos se alejan mucho. Se puede ver de una forma más nítida como el módulo de la velocidad del modelo PIV alcanza un valor máximo de 25 m/s, mientras el modelo CFD escasamente llega a los 10 m/s. No obstante, mientras avanza la simulación y llegamos al punto muerto superior, se puede visualizar como los dos modelos se asemejan, dando valores cercanos a los 0 m/s en 4 CAD.



Gráfica 5.2: Representación de la velocidad axial promediada (azul continua), velocidad axial por ciclo (azul discontinua) y velocidad axial en el PIV (negro continua) en el -10 CAD



Gráfica 5.3: Representación de la velocidad axial promediada (azul continua), velocidad axial por ciclo (azul discontinua) y velocidad axial en el PIV (negro continua) en el -10 CAD

5.6 Turbulencia

La turbulencia es una variable complicada a la hora de poder sacar conclusiones y análisis. Igualmente, se puede percibir unas diferencias muy notables entre los dos modelos, ya que la turbulencia del modelo CFD actual es muy constante en su posición, mientras que va variando su intensidad entre fases, en la compresión, como en el punto muerto superior y en la expansión. En cambio, el modelo empírico varía mucho entre las tres fases, teniendo una zona de mucha turbulencia que va avanzando desde la garganta al interior de la cámara hasta el PMS, la cual desaparece a partir de 2 CAD.

Las gráficas TKE muestran la media de los 75 ciclos del modelo experimental y la media progresiva de los ciclos del modelo actual, siendo como en el campo de velocidades, el ciclo catorce el más importante a analizar.

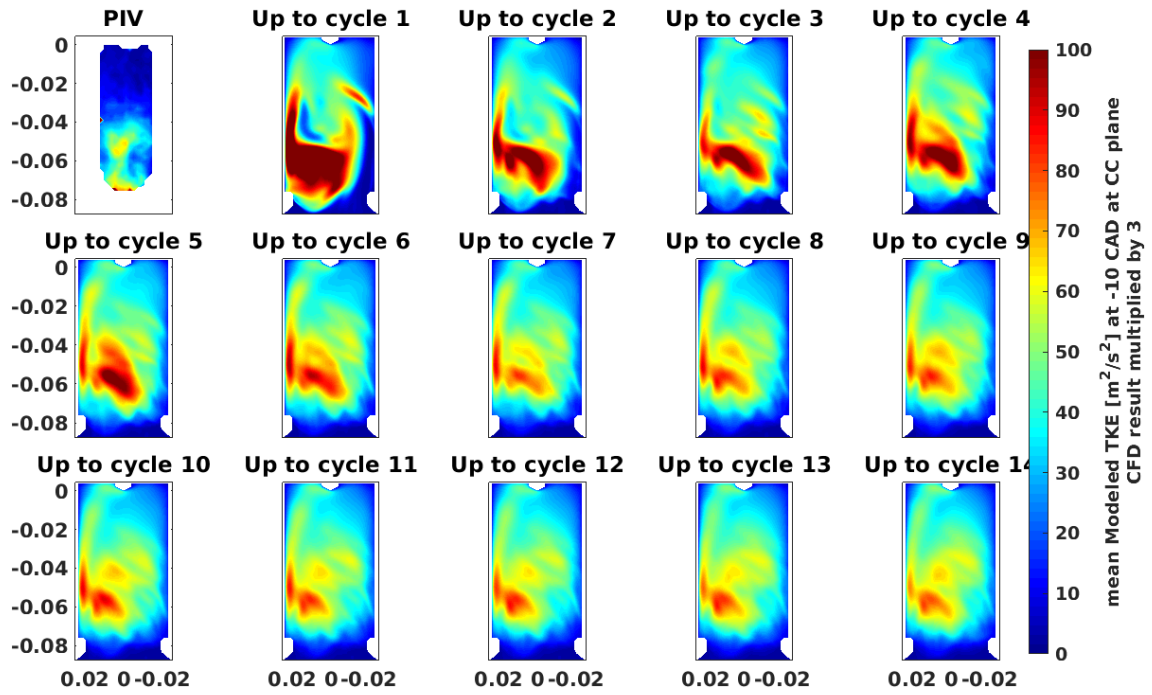


Figura 5.9: Turbulencia promediada en el plano CC en -10 CAD

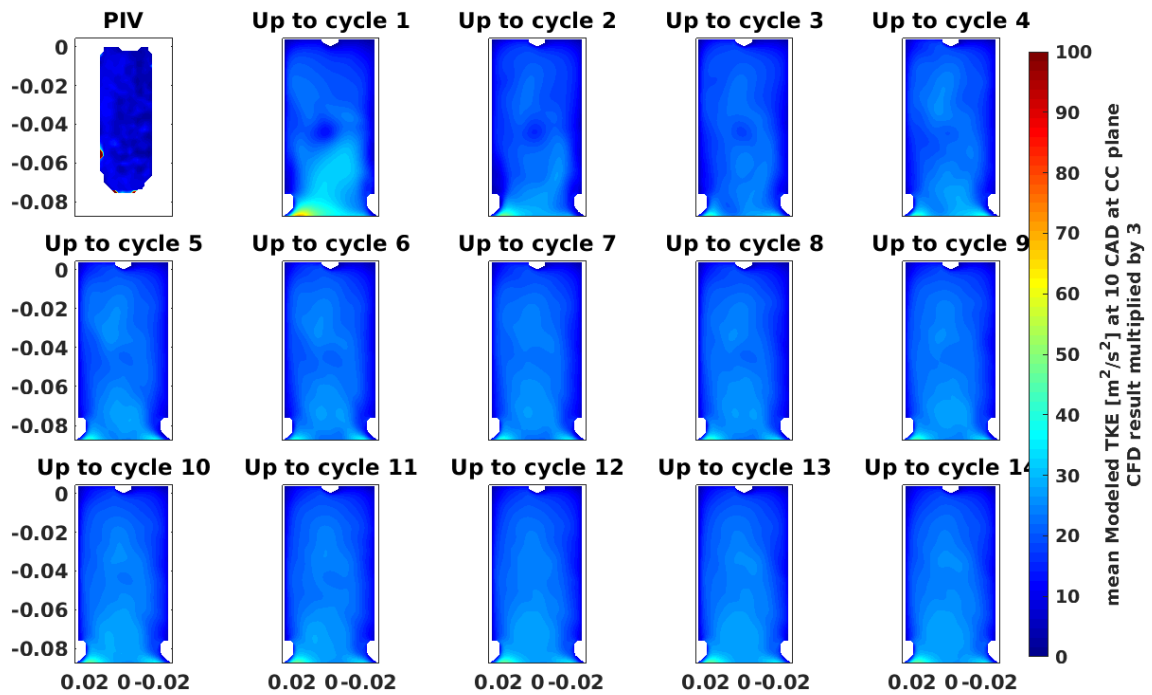


Figura 5.10: Turbulencia promediada en el plano CC en +10 CAD

6. Conclusiones

En este apartado se plantean las diferentes conclusiones que han surgido con el análisis de los resultados obtenidos con el modelo CFD actual. A parte, al final de este capítulo, se dan a conocer diferentes ideas para continuar el estudio del *motor maqueta* para realizar un modelo teórico que se adapte a los experimentos.

Las variables que se han comparado muestran diferencias entre experimentos y modelos. En el campo de velocidades se ha visto unas diferencias importantes en la intensidad de la velocidad, por debajo de la mitad del valor experimental, y la tendencia que tenía el flujo en llenar la cámara de combustión. A partir del punto muerto superior cambiaba, siendo muy parecido, tanto en dirección como en intensidad.

Estas diferencias en el tramo de compresión pueden llegar por varias razones, pero la más importante es el choque entre el flujo a gran velocidad que empuja el pistón y el flujo parado que se ubica en el interior de la cámara de combustión. Ese momento crea mucha turbulencia dentro de la cámara, con un flujo muy caótico y con una homogeneidad muy pobre. Esto provoca que los programas de simulación como Converge CFD les sea muy complicado hacer una réplica exacta de ese intervalo de tiempo, ya que a la mínima variación de una variable con la realidad altera los resultados del modelo con facilidad. En cambio, cuando se supera el punto muerto superior, se observa un flujo muy homogéneo evacuando la cámara de combustión. Por este motivo, la expansión es fácil de tratar para los simuladores y conceden resultados más afines a la realidad.

En este Trabajo de Fin de Grado se han calculado hasta 15 ciclos consecutivos para comprobar cuándo el valor medio de todos los ciclos se estabilizaba. Esa repetibilidad se inicia aproximadamente en el ciclo ocho, más temprano que los modelos CFD estudiados anteriormente, a partir del cual la media no se modifica apenas.

Por las razones antes mencionadas, se pueden dejar claras varias conclusiones que podrían provocar las diferencias entre el modelo experimental realizado con la técnica PIV y el modelo teórico obtenido con Converge CFD:

- Las diferencias en el tramo de compresión vienen dadas por el barrido de los gases. Esto lo puede provocar diferencias en la geometría del modelo, que no se haya hecho con una precisión muy avanzada.
- Las condiciones iniciales pueden haber influido en el barrido de gases, como la velocidad, la temperatura y la presión de entrada y salida del fluido. Además, las diferencias de estas variables pueden crear perturbaciones donde no deberían estar o que se reduzca el gradiente de presiones y temperatura del fluido y no avance con la suficiente rapidez.
- El uso una malla base menos refinada provoca una baja resolución en momentos de poca homogeneidad en el fluido, como se ha visto en la zona del cilindro que se produce cuando se descubren las lumbreras de escape antes que las lumbreras de admisión.

- Solo el uso del *embedding* puede provocar que no se estudie todas las zonas comprometidas del modelo, cosa que se puede asegurar implementando el AMR.

Es de gran dificultad obtener conclusiones a partir de los resultados obtenidos de los modelos. Sigue siendo visible en el modelo CFD actual cómo el campo de velocidades ciclo a ciclo puede variar en dirección, intensidad y configuración (situación de los vórtices). Una de las explicaciones de este problema es la resolución de las ecuaciones de turbulencia. Las ecuaciones de turbulencia son cálculos iterativos con un grado de dificultad altos, donde si no se alcanza una resolución mínima a costa del coste computacional, pueden variar significativamente los resultados.

6.1 Trabajos futuros

Aún queda mucho que trabajar para obtener un modelo CFD que se pueda aplicar al *motor maqueta*. Un paso importante sería no estar atado al coste computacional de los cálculos de Converge. De esta forma, se podrían aplicar mallas más refinadas en los momentos más críticos del estudio, como la entrada del flujo en la cámara de combustión en el tramo de compresión o el descubrimiento de las lumbreras de admisión y escape en el intervalo de tiempo cercano al punto muerto inferior.

Otro punto muy interesante que se puede estudiar es un cambio en el modelo de turbulencia que se ha estado usando hasta ahora. El modelo actual es el RANS que facilita la estabilidad del modelo, pero el modelo de cálculo LES se orienta más a la resolución de la turbulencia, ya que realiza una determinación mejor de las escalas de flujo. Ya se ha visto anteriormente que hay unas diferencias importantes en la ubicación y la intensidad y la turbulencia entre los dos modelos tratados.

7. Bibliografía

- [1] P. Fernández Bérjon, “Caracterización del flujo interno en la cámara de combustión de un motor óptico mediante velocimetría por imágenes de partículas”, Trabajo Final de Grado, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Valencia, 2016.
- [2] María José Ruiz Lucas, “Modelado CFD del movimiento del aire en el cilindro de un motor de dos tiempos con accesos ópticos”, Trabajo Final de Máster, Escuela Técnica Superior de Ingeniería del Diseño, Universitat Politècnica de Valencia, 2016.
- [2] Converge 2.2.0 Theory Manual. Convergent Science, Inc. 2015.
- [3] Converge studio 2.2.0 Manual. Convergent Science, Inc. 2015.
- [4] Profesor Payri González, profesor Desantes Fernández, “Motores de combustión interna alternativos”, Ed. Reverté, 2011.
- [5] EnSight User Manual for Version 10.2. Computational Engineering International, Inc. 2017.

ANEXO

I. Pre-procesado

I.1 Introducción

Este apartado explica detalladamente los pasos a seguir para la ejecución de pre-procesado igual al de este proyecto con el programa Converge. Se ha separado en diferentes secciones el pre-procesado del post-procesado, ya que estos dos se realizan con programas diferentes y se explica cada uno por separado.

Se ha dividido los apartados en el diseño del motor, la asignación de los contornos, la preparación de la geometría y la preparación de la *SetUp*.

I.2 Diseño del motor

Modelado CFD (Computational Fluid Dynamics) es un tipo de cálculo que se basa en la generación de mallas que envuelven la estructura a estudiar y saca los resultados de las variables que se deseen de cada instante en cada celda de la malla. Lo que diferencia a Converge del resto de programas de modelado CFD es que genera automáticamente una malla ortogonal durante el cálculo del modelo, con los parámetros iniciales que se definen por el usuario. Gracias a este aspecto, la malla se adapta a la potencia computacional que posee el usuario, al contrario que el resto de los programas de modelado CFD, donde la malla se adapta a la complejidad de la geometría.

Converge consigue este tipo de mallado, las celdas son delimitadas en la intersección con la superficie, así se puede adaptar a geometrías complejas. Es importante que, para la creación de este mallado, la superficie en cuestión ha de ser triangulada y cerrada.

Antes de empezar a hacer el preprocesado en Converge CFD es necesario realizar un modelo de la estructura del estudio en un programa de diseño gráfico. En este caso, la estructura del estudio es el *motor maqueta* y el programa de diseño es Ansys. El diseño de la estructura no se ha elaborado en este proyecto, lo realizó un compañero del CMT con la geometría exacta del motor.

En este trabajo, respecto a otro Trabajo de Fin de Máster (TFM) similar, la geometría representada en Ansys tiene más exactitud que el anterior. La diferencia más notable que se puede observar son las lumbreras de admisión. El modelo antiguo las lumbreras de admisión son independientes entre ellas, en ningún momento comparten la entrada de gases limpios. Por el contrario, el modelo actual representa exactamente como es el *motor maqueta*, donde las lumbreras de admisión comparten la entrada de gases limpios con un *plenum*.

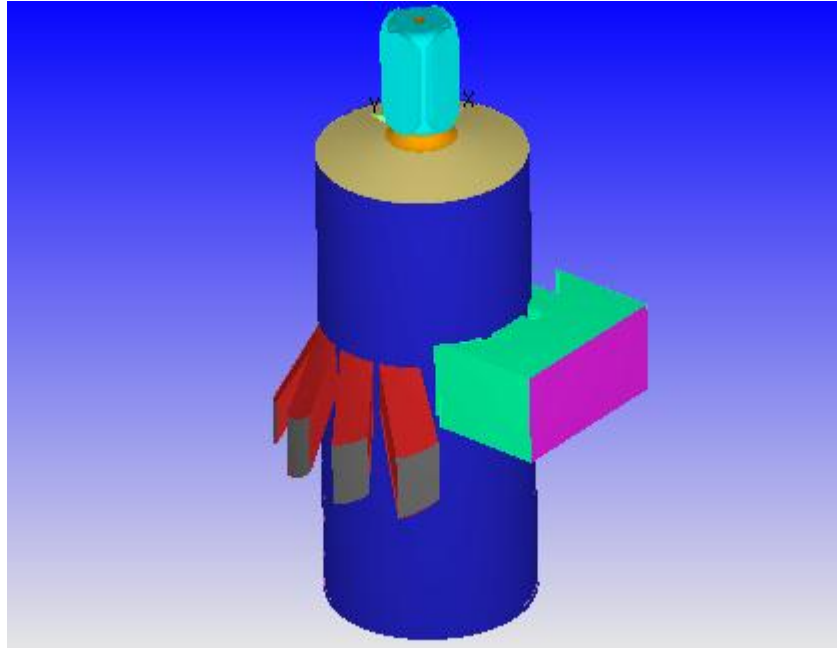


Figura I.1: Geometría modelo antiguo sin el plenum de admisión

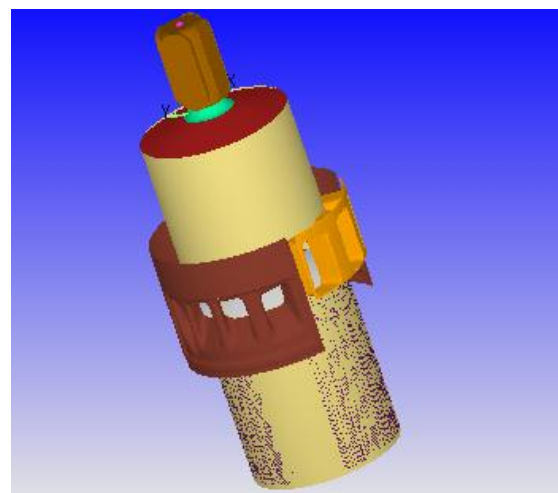
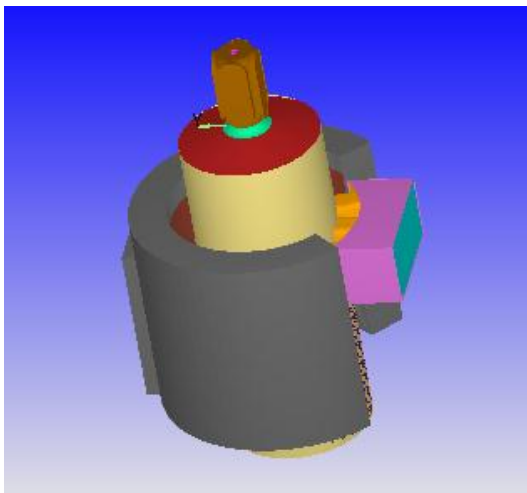


Figura I.2: Configuración actual de la geometría con el plenum de admisión (izquierda) y el modelo actual sin el plenum de admisión (derecha)

I.3 Asignación de los contornos

Así que, cuando ya se tiene el modelo del motor, importamos el fichero .CVG que lo contiene. El primer paso debe ser la asignación de las superficies del diseño en cuestión. El *motor maqueta* se dividirá en 7 contornos o *boundaries*. Se seleccionan las paredes que se quieran asignar a un *boundary* y seguidamente, se le nombra y recibe un color como el usuario desee.

El programa Converge nos facilita una opción para definir cada superficie. En la pantalla de inicio de Converge, a la izquierda se encuentra con la ventana de geometría. En esta ventana se puede modificar la geometría de la estructura para adaptarla a las necesidades del usuario.

Dentro de la ventana de geometría hay una opción, *boundary*. Dentro de *boundary* se elige la opción *Flag*. En la última opción se selecciona la superficie que se desee y se la asigna un nombre y un color.

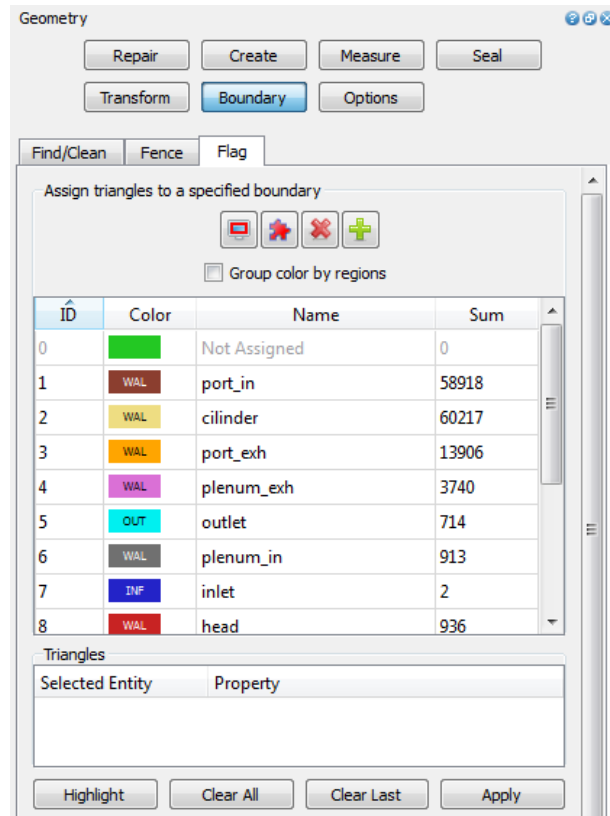


Figura I.3: Ventana Geometry>Boundary>Flag para asignar las superficies deseadas

En este trabajo se ha nombrado a las superficies de la siguiente manera:

- Superficies que corresponden a la admisión:
 - Port_in (marrón): lumbreras de admisión.
 - Plenum_in (gris): *plenum* de admisión.
 - Inlet (azul): superficie de entrada del *plenum* de admisión.
- Superficies que corresponden al escape:
 - Port_exh (naranja): lumbreras de escape.
 - Plenum_exh (violeta): *plenum* de escape.
 - Outlet (azul cielo): superficie de salida del *plenum* de escape.
- Superficies que corresponden a la cámara:
 - Cylinder (amarillo): cilindro.
 - Head (rojo): cabeza del cilindro.

- Cc_01 (verde): unión de la cámara de combustión y el cilindro.
- Cc_02 (marrón claro): cámara de combustión.
- Cc_03 (rosa): inyector.
- Piston_01 (blanco): cabeza del pistón.
- Piston_02 (lila): falda del pistón.

I.4 Preparación de la geometría

Después de asignar cada *boundary* a las superficies deseadas, el siguiente paso debe ser el hecho de solventar todos los errores que ha podido tener el diseño del modelo y la lectura de este por Converge. El programa nos simplifica esta tarea con la opción *Repair*. En la pantalla de inicio de Converge, abajo a la derecha está la opción *Diagnosis*. En esta opción se encuentra el menú para determinar que errores se desea encontrar. Después configurarlo se lanza con el botón *Find* y Converge se encarga de encontrar los errores.

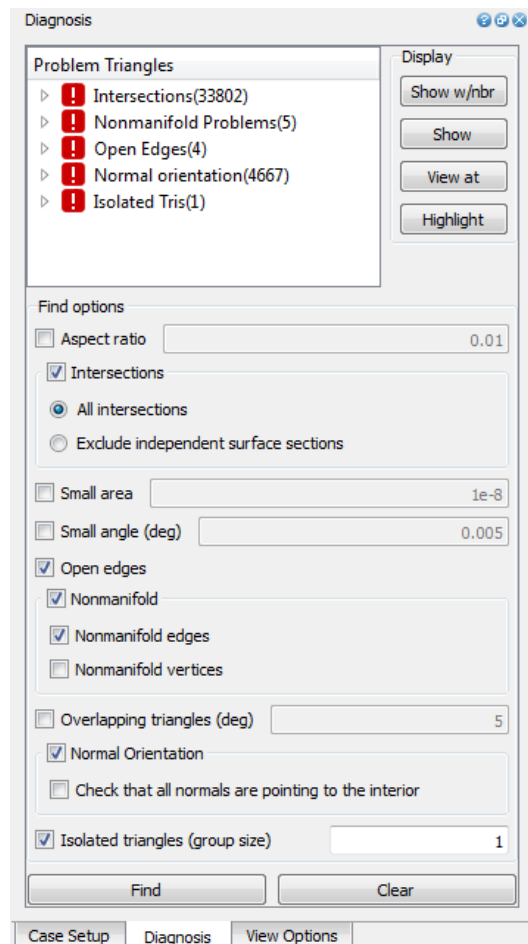



Figura I.4: Ventana de *Diagnosis* con los errores encontrados de la geometría actual

Los errores que aparecen después de lanzar la opción *Find* son: *Intersections*, *Nonmanifold Problems*, *Open edges*, *Normal orientation* e *Isolated Tris*. A continuación, se explica cómo subsanar cada conflicto.

Intersections

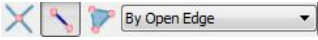
El problema de intersecciones de esta geometría es que el cilindro y el pistón tienen exactamente el mismo diámetro, entonces, no puede haber ningún movimiento. La solución es seleccionar la cabeza y la falda del pistón (piston_01 y piston_02) y reducir su diámetro unos milímetros. Para medir el diámetro de las superficies hay que usar la opción *Geometry>Measure>Direction>Arc Normal* y con la herramienta  marcamos tres vértices de la circunferencia de cada contorno. Da un diámetro de 0.15 metros, entonces rebajamos el diámetro de los *boundaries* Piston_01 y Piston_02 con la opción *Geometry>Transform>Scale*, seleccionamos los dos y reducimos su escala un 0.999. Finalmente, se obtiene un diámetro de 0.15 metros en el Cylinder y un diámetro de 0.14998 metros para el Piston_01 y Piston_02.

Nonmanifold problems

Generalmente, los dos tipos de errores en *nonmanifold problems* son los objetos abiertos y las caras no deseadas. El problema de los objetos abiertos ocurre cuando hay agujeros en los objetos, en cambio, el problema de las caras no deseadas ocurre cuando se han formado caras o bordes extras y distorsionan el objeto.

Para solucionar este error se puede seguir el mismo procedimiento que en *open edges*.

Open edges

El problema de aperturas en una superficie de la geometría se arregla rellenando esos huecos con triángulos nuevos. Para ello, hay que seleccionar *Open Edges* y pulsar *Show*. En pantalla se mostrará los triángulos colindantes a esta apertura. Con la opción  se selecciona la línea que rodea esa apertura. Cuando ya se haya seleccionado, se usa la opción *Geometry>Repair>Patch* y se pulsa *Apply*.

Normal orientation

Para solventar este error hay que cambiar la orientación de los vectores normales para que apunten al eje central del motor maqueta, ya que todas las partes son concéntricas. Para hacer esto hay que ir a *Geometry>Transform>Normal* y seleccionar los triángulos con una orientación errónea y, por último, pulsar *Apply*.

Isolated Tris

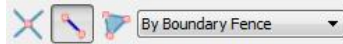
Este problema se basa en que hay zonas aisladas en un *boundary* que no corresponden a este o huecos que no debería haber en una superficie. Este error se suele solventar al reparar los *open edges* y los *nonmanifold problems* al crear triángulos que rellenen los huecos que faltan y asignarles el *boundary* que les toca.

I.4.1 Seals

A parte de solucionar los errores que hay en la geometría, se debe tener en cuenta la particularidad de las lumbreras de admisión y escape en un motor de 2 tiempos. Se necesita usar un sellado cuando el pistón cierra las lumbreras de escape y de admisión. También hay que sellar la cabeza del pistón contra el cilindro para que los fluidos no se salgan de la zona de combustión.

Para poder sellar las zonas que se ha comentado antes hay que usar la opción *Seal* que se sitúa en la ventana *Geometry*. Las dos zonas por sellar son entre el pistón y la camisa del cilindro (*cilinder*) y otro entre los puertos de entrada y salida del flujo por las lumbreras y la falda del pistón (*piston_02*). Seguidamente, se describen los pasos para crear los *sealing*:

1. Crear una nueva fila en *Configuration* pulsando *Add Seal*.
2. Se elige el contorno que se quiere aplicar el sellado en la lista *To Boundary*.
3. Se selecciona los bordes desde los que cada sello compartirá. Estos bordes se tienen que elegir con la opción:



Esta es la única forma de seleccionar los puertos de entrada y salida del cilindro y la circunferencia que une la cabeza y la falda del pistón.

4. Se define la dirección del *seal* a través de los triángulos seleccionados. En *Direction* se selecciona la opción *Orthogonal*, así la dirección del sellado será perpendicular a superficie que se desee. Para el caso de los puertos de entrada y salida de las lumbreras se elige el *cilinder*. Para el otro caso, *piston_02-cilinder*, la superficie a seleccionar es la falda del pistón.

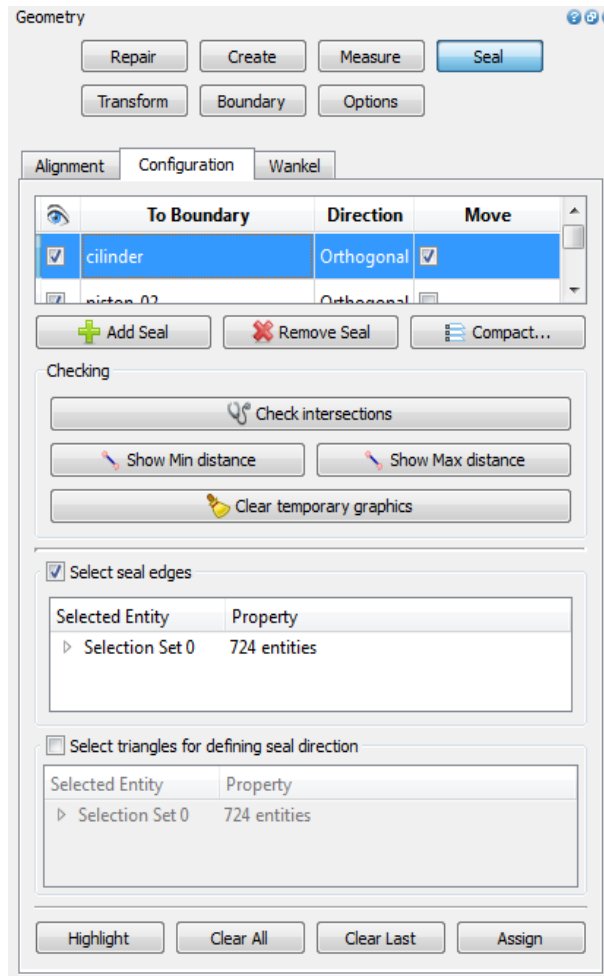


Figura 1.5: Ventana para la creación de seals

Después de haber reparado los posibles fallos de la geometría y haber añadido los sellados pertinentes se debe seguir con la preparación de las condiciones iniciales del estudio para su posterior cálculo

1.5 Preparación de la *Set Up*

La opción *Case Setup* que proporciona Converge permite ajustar las condiciones iniciales que desee el usuario. En este caso se condiciona varios aspectos que se tratan a continuación:

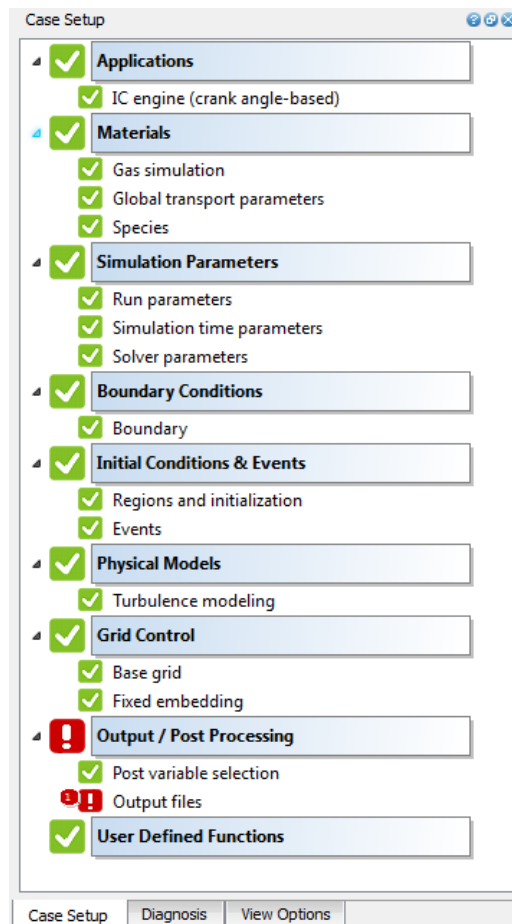


Figura 1.6: Ventana de la Case Setup para la configuración del modelo

1.5.1 Applications

En este apartado se le comunica a Converge cual es la aplicación del estudio. Se puede elegir entre un motor de combustión interna o un chorro generalizado. En este estudio, se trabaja con un motor de combustión interna y los intervalos temporales se desarrollan con el ángulo de avance del cigüeñal. En otro caso que se eligiera el chorro, los intervalos temporales se desarrollarían en segundos.

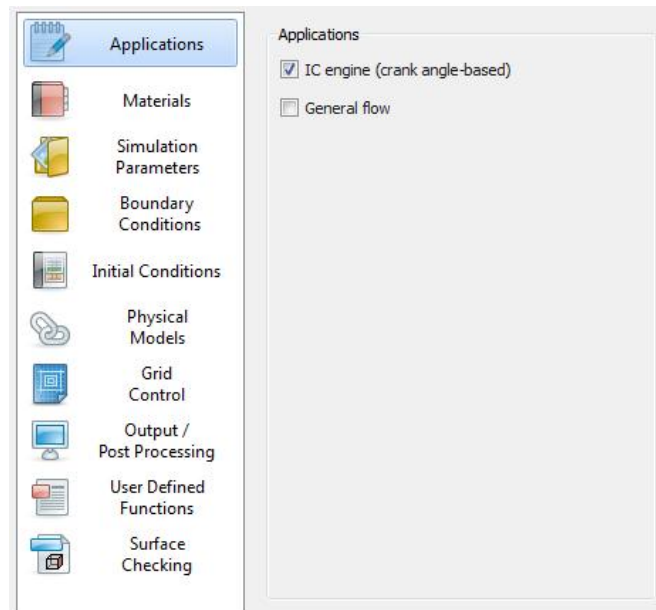


Figura I.7: Ventana para la configuración de applications del modelo

Después de elegir el tipo de intervalo temporal, se tiene que referenciar las superficies del motor. El *piston Surface ID* es el *piston_01* (cabeza del pistón), el *liner ID* es el *cylinder* (superficie del cilindro) y el *head ID* es el *head* (cabeza del cilindro).

I.5.2 Materials

Esta opción se encarga de definir si la simulación se hace con un gas, líquido, sólido o con paquetes. Además, se puede establecer si se trabaja con el mecanismo de reacción, con el mecanismo de reacción de la superficie o la composición de nuestra mezcla. Como en este proyecto no provocamos una combustión en el motor, solo se marca *Species*, donde en los siguientes pasos se especificará la composición.

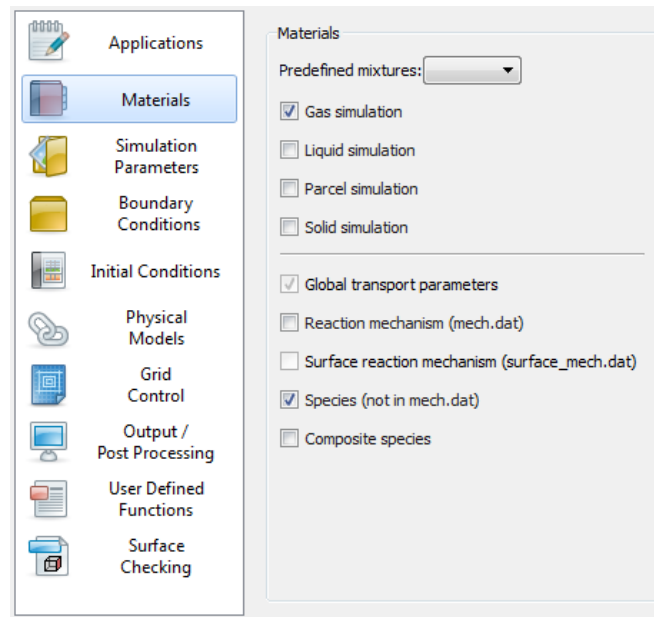


Figura 1.8: Ventana para la configuración de materiales del modelo

Luego hay que especificar las condiciones de cada opción que se haya marcado. En *Gas simulation* se usa la ecuación de estado Redlich-Kwong, la temperatura crítica es 133.0 grados Kelvin y la presión crítica es $3.77e+06$ Pascales.

En *Global transport parameters* fijamos el número de Prandtl y Schmidt a 0.9 y 0.78, respectivamente. El número de Prandtl es un número adimensional cociente entre la difusión de cantidad de movimiento (viscosidad) y la difusividad térmica. El número de Schmidt es el cociente entre la viscosidad y la difusión de masa.

Por último, en *Species*, establecemos la composición del fluido del interior del motor, del que entra y sale. En este estudio se usa el N₂ (nitrógeno), ya que no queremos una combustión en el motor.

1.5.3 Simulation Parameters

En los parámetros de simulación no se marca ninguna casilla más de las que están marcadas automáticamente ya que no se usará nada más para el cálculo. Estas son: *Run parameters*, *Simulation time parameters* y *Solver parameters*.

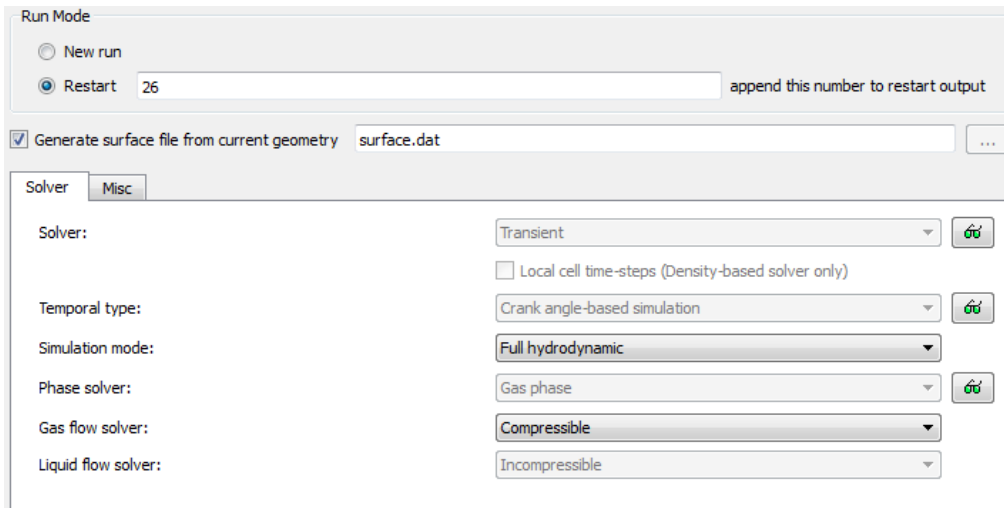


Figura I.9: Menú de los Run Parameters

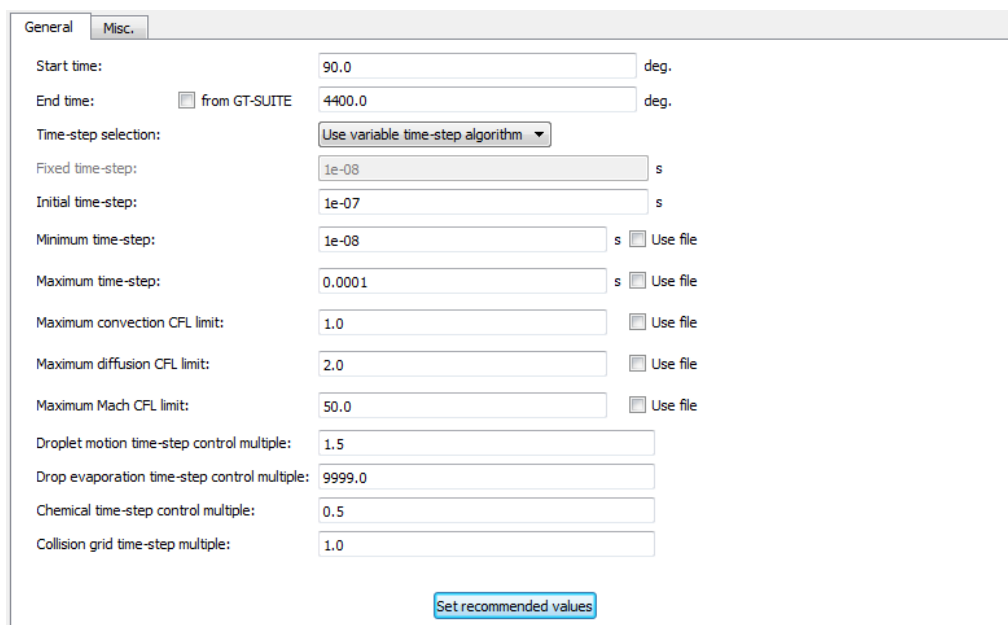


Figura I.10: Menú de los Simulation Time Parameters

En *Run parameters* se elige el tipo de modo de simulación, *Full hydrodynamic*. También se elige el tipo de gas que se trabaja, el cual es compresible.

En *Simulation time parameters* se debe fijar el tiempo de simulación para el posterior cálculo. Se empieza con 90 grados de cigüeñal hasta los 4400 grados, unos 12 ciclos aproximadamente.

En *Solver parameters* no se debe cambiar ninguna condición ya que, para este proyecto, las condiciones predeterminadas son válidas.

I.5.4 Boundary Conditions

En las condiciones de contorno se deja la opción que hay marcada de *Boundary*, ya que para solo se establecerán condiciones al contorno, sin ninguna condición extra.

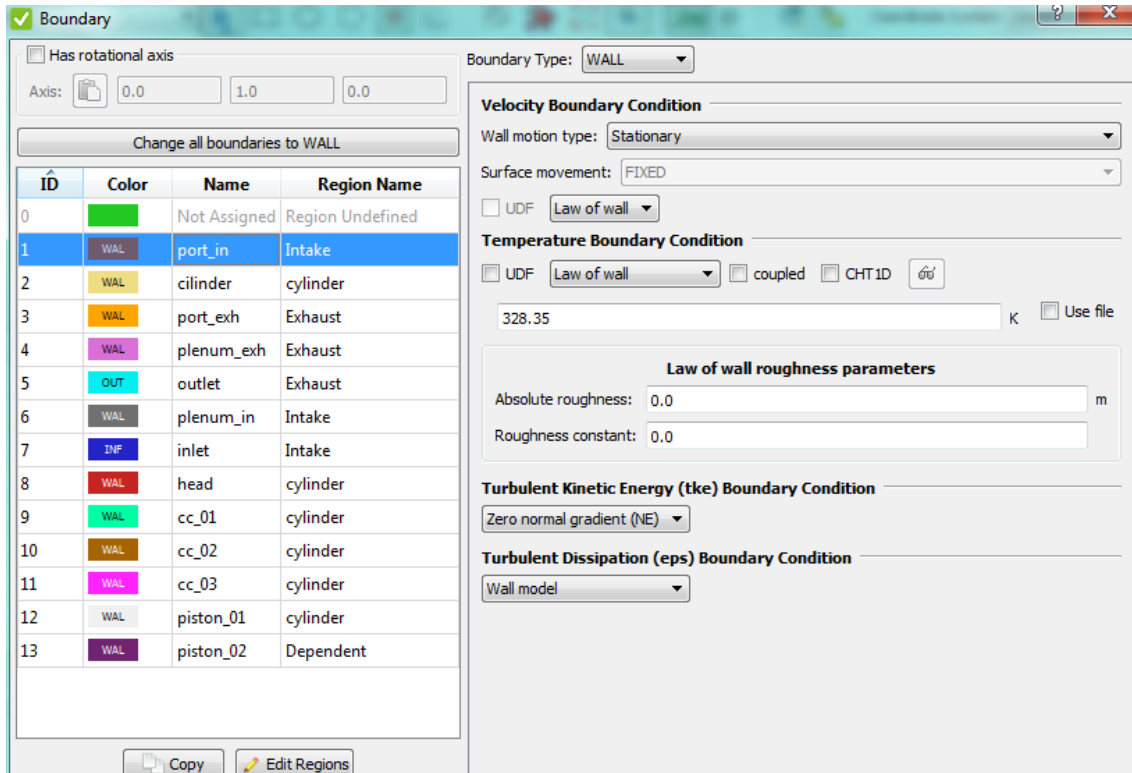


Figura I.11: Menú de los Boundary

En *Boundary* se tiene que condicionar cada contorno que se haya creado. Las superficies de admisión se separan en *Intake*, las superficies de escape en *Exhaust* y las superficies del cilindro en *Cylinder*. Además, la falda del pistón se separa en *Dependent*.

Se debe especificar el tipo de contorno que es cada uno, donde todos menos Outlet e Inlet son *Wall*. *Wall* significa que son una pared donde no tiene que traspasar nada. Inlet es *Inflow*, entrada de flujo. Outlet es *Outflow*, salida de flujo.

En cada *boundary* se define cual va a ser la temperatura de esta y en Inlet y Outlet, se especifica que tipo de flujo entra y sale del motor, en este caso N2.

I.5.5 Initial Conditions & events

En las condiciones iniciales se necesitará, a parte de la ya marcada *Regions and initialization*, la opción *Events*. Esta opción permite especificar las aperturas de zonas o el cierre de estas dependiendo el momento del ciclo. Todas están abiertas, ya que el flujo tiene que circular desde la entrada de la admisión hasta la salida del escape ya que es un ciclo abierto.

En *Regions and initialization* se le da un nombre a cada región, el *cylinder* tiene la ID 0, el *intake* tiene el ID 1 y el *exhaust* tiene el ID 2. Los contornos del cilindro se les nombra con la region 0, los de admisión con la región 1 y los de escape con la región 2 en la opción de *boundary conditions>boundary* en el *Case Setup*. Por otra parte, se define las especies que tiene cada región, que como ya se ha comentado, es el N2.

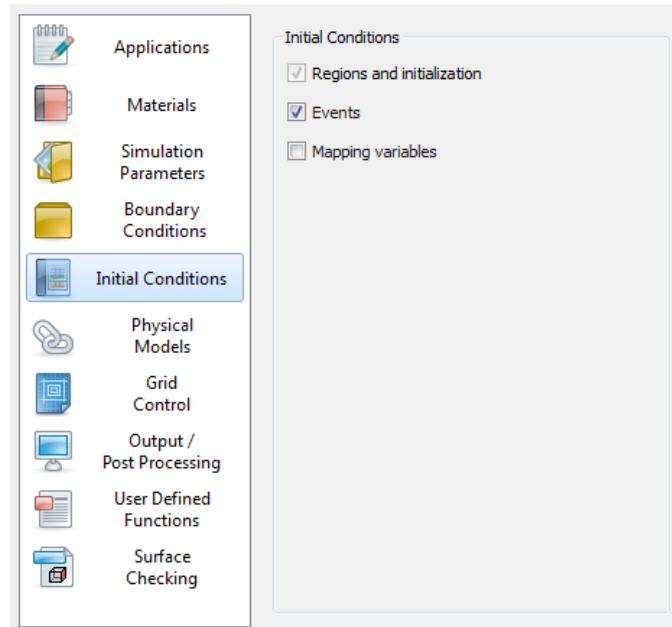


Figura I.12: Ventana para la configuración de Initial Conditions del modelo

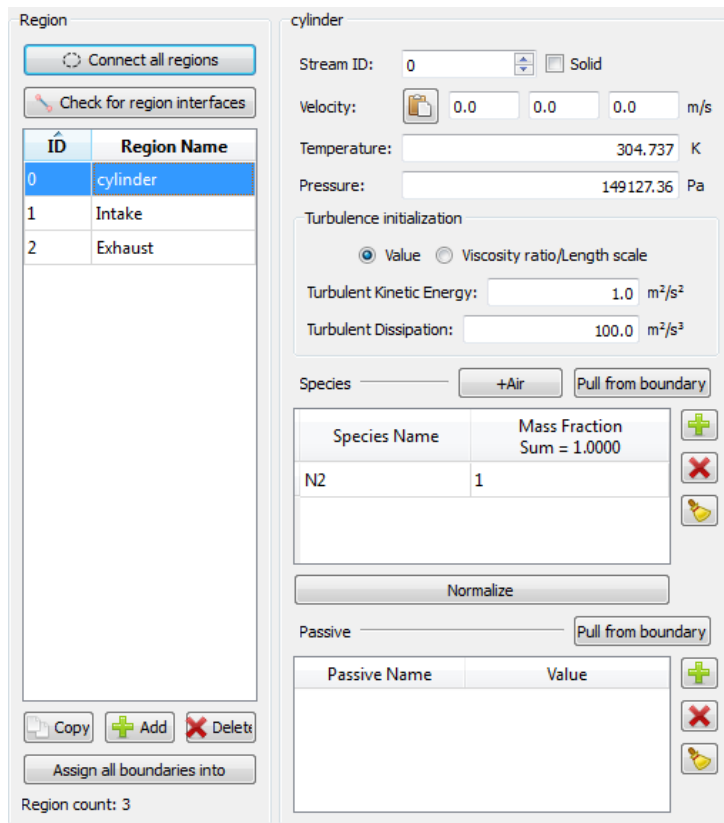


Figura I.13: Menú de los Initial Conditions para cada región

I.5.6 Physical Models

Solo se usa el modelo turbulento que ya está marcado automáticamente. Esto ocurre porque en este estudio no se quiere ver la combustión del ciclo, solo el movimiento del flujo dentro de la cámara de combustión.

I.5.7 Grid Control

Esta opción sirve para remarcar zonas donde se quiere hacer un estudio más intensivo. Para ello, a parte del *Base grid*, también se marca el *Fixed embedding*, que sirve para establecer las zonas que se desee hacer un mallado con unas celdas más pequeñas para poder recibir más datos. Con el *Base grid* se especifica el tamaño de las celdas del mallado en toda la estructura, generalizando.

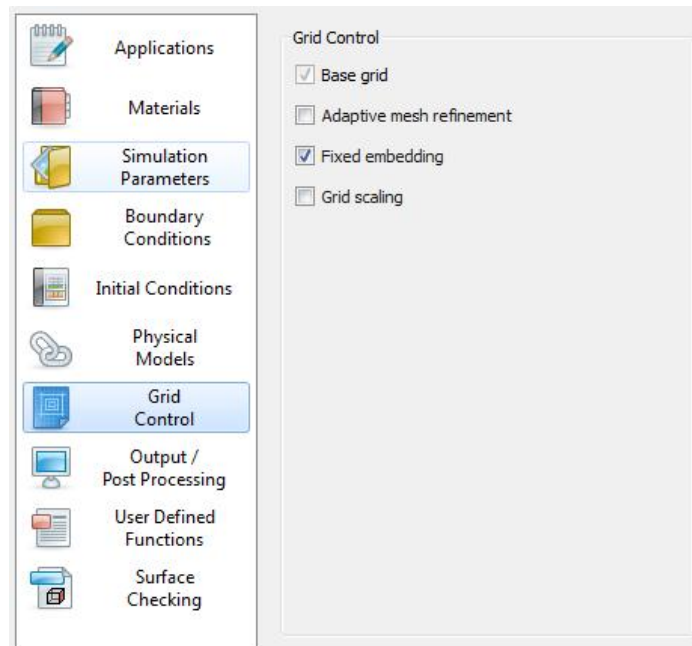


Figura I.14: Ventana para la configuración de Grid Control del modelo

I.5.8 Output/Post Processing

En este apartado no se puede modificar nada. La razón es que no se puede modificar los datos de salida que nos proporciona Converge. En *Post variable selection*, se marca las opciones que el usuario quiera estudiar. En este caso se marca las siguientes variables: densidad, masa, presión, temperatura, velocidad, ejes i, j y k y la región ID.

En *Output files* se especifica cada cuantos grados de cigüeñal se desea obtener datos y donde se escribirán estos.

II Cálculo del modelo

II.1 Introducción

En esta sección se explica brevemente el programa usado, en combinación con Converge, para el cálculo del modelo configurado en el apartado anterior. El programa es el MobaXterm 10.4.

II.2 MobaXterm

El cálculo del modelo se realiza después de la configuración del modelo llevando a cabo el *solver* de Converge. Para proceder al cálculo del caso se ha tenido que usar el programa MobaXterm 10.4, un servidor común con formato Linux que conecta con los 64 procesadores del DMMT para realizar el cálculo con más rapidez.

Una herramienta muy útil de MobaXterm es la capacidad para pausar el cálculo y poder volverlo a lanzar desde el mismo punto o cualquier diferente. Esto puede ahorrar mucho tiempo si se detecta un error en el cálculo, poder solventarlo y no tener que calcular todo el caso de nuevo, solamente unos grados antes del momento del error. Además, los procesadores se suelen saturar ya que hay muchos investigadores lanzando cálculos a diario. En estos casos hay que dar prioridad a los trabajos o estudios más importantes y es necesario parar más de un cálculo para no sobrecargar los procesadores.

Para poder parar el cálculo se debe usar el comando *top* de MobaXterm, copiar la identificación del cálculo en cuestión (su ID) y utilizar el comando *kill* con el ID copiado anteriormente.

Converge tiene la característica de guardar un fichero llamado *restart* durante el cálculo. Este fichero genera un autoguardado de la información del cálculo con el intervalo de tiempo que el usuario haya introducido. Cada vez que Converge genera un fichero *restart*, seguido del nombre del fichero hay un número que marca cuantos ficheros ha creado. Es importante no elegir un intervalo de tiempo de autoguardado muy corto, debido a que son muchos datos que guardar y podría alargar mucho el cálculo.

Después de encontrar un error o solamente querer reanudar el cálculo, hay que escribir el comando *rstrtflg* seguido del número de *restart* que se quiera.

III Post-procesado

III.1 Introducción

Cuando ya se han obtenido los cálculos del modelo, Converge proporciona los datos en ficheros **.out* y los *post*.out*. Los ficheros **.out* muestran valores promediados, en cambio, los *post*.out* presentan valores individuales de las celdas.

Los primeros pueden ser leídos por programas de la índole de Matlab. Antes de empezar el post-procesado de los ficheros *.out con Matlab hay que convertir estos ficheros a formato *.col. Esta conversión se hace con el programa MobaXterm y el comando *post convert*. Con este comando se eligen los ficheros que se quieren convertir y en un breve período de tiempo realiza la acción. Los ficheros *.col son ficheros en formato de columna que permite tratarlos como matrices y se ha usado en Matlab para la comparación de los datos de la técnica PIV y los datos actuales.

Por otro lado, los otros ficheros están en formato binario, esto provoca que no se pueda hacer ningún tipo de post-procesado con ellos. Por ese motivo, hay que usar la opción de *Converge, Post-Processing 3D*. Después de convertir los ficheros en formato binario a un formato que se pueda leer por programas 3D, se usará el programa Ensign para su post-procesamiento y visualización.

III.2 Rutinas de Matlab

En este capítulo se muestran las principales rutinas empleadas en Matlab para la conversión de los resultados en forma numérica obtenidos después del cálculo del modelo en una representación gráfica para tener más facilidad en su análisis. Hay rutinas de todo tipo, como la comparación del modelo obtenido con PIV con las diferentes variables calculadas, la conversión de capturas del flujo en el cilindro en un vídeo o la comparación entre presión y gasto másico del modelo CFD anterior a este y el actual.

III.2.1 Comparación PIV y modelo actual

```
% To post-process several cycles and compare axial velocity maps and axial
% velocity profiles along spray axis with PIV data

% It also includes a comparison with PIV velocity maps

clearvars

% Inputs
ciclos=1:14; % Casos para procesar
main_path='/data/combustion/CONVERGE/TFG_BORJA/2S_real';

plano='C';
Font_Size=15;
magnitud='M'; % Options U (axial component) | M (modulus of velocity)
TKE_factor=2/3;
inj_pos=87.773e-3; % z en metros
CAD_position=-10:1:15;
step=0.5e-3; % [m]
lim_axial_gasto=0:step:inj_pos;
lim_radial_gasto=-0.01:step:0.01;

pathPIV2=['/data/leopacpr/CFD_Datasource/MOMAQ/PIV data/',plano];
packagesize=50;
```

```

ruta2save=[main_path,'/statistics/up1cycle_',num2str(ciclos(end)),'_',plano,'_plane']; %
Folder to save post-processed data

mkdir(ruta2save)
mkdir([ruta2save,'/Velocity_CCV'])
mkdir([ruta2save,'/Velocity_mean'])
mkdir([ruta2save,'/Modeled TKE'])
mkdir([ruta2save,'/Resolved TKE'])
mkdir([ruta2save,'/Velocity_COV'])
mkdir([ruta2save,'/Velocity_std'])

if strcmp(plano,'CC')
    [xi,yi,zi] = meshgrid(0,-0.03:step:0.03,0:step:0.1); % Plano central CC
    xi=squeeze(xi);
    yi=squeeze(yi);
    ri=yi;
elseif strcmp(plano,'C')
    [xi,yi,zi] = meshgrid(-0.03:step:0.03,0,0:step:0.1); % Plano central C
    xi=squeeze(xi);
    yi=squeeze(yi);
    ri=xi;
elseif strcmp(plano,'B')
    [xi,yi,zi] = meshgrid(0.01,-0.03:step:0.03,0:step:0.1);
    xi=squeeze(xi);
    yi=squeeze(yi);
    ri=yi;
elseif strcmp(plano,'F')
    [xi,yi,zi] = meshgrid(-0.01,-0.03:step:0.03,0:step:0.1);
    xi=squeeze(xi);
    yi=squeeze(yi);
    ri=yi;
elseif strcmp(plano,'L')
    [xi,yi,zi] = meshgrid(-0.03:step:0.03,0.01,0:step:0.1);
    xi=squeeze(xi);
    yi=squeeze(yi);
    ri=xi;
elseif strcmp(plano,'R')
    [xi,yi,zi] = meshgrid(-0.03:step:0.03,-0.01,0:step:0.1);
    xi=squeeze(xi);
    yi=squeeze(yi);
    ri=xi;
end

zi=squeeze(zi);

% Mask

if strcmp(plano,'CC') || strcmp(plano,'C')
    mask_data=xlsread('/data/leopacpr/CFD_Datasource/MOMAQ/MOMAQ mask.xlsx','centered');
    radial_mask=mask_data(:,2);
else
    mask_data=xlsread('/data/leopacpr/CFD_Datasource/MOMAQ/MOMAQ mask.xlsx','shifted');
    radial_mask=mask_data(:,1);
end

radial_grid=ri(:,1);
axial_grid=zi(1,:);

```

```

axial_mask=mask_data(:,3);

radial_mask_idx=zeros(1,length(radial_mask)*2);
axial_mask_idx=zeros(length(axial_mask)*2,1);

ii=length(radial_mask)*2;

for i=1:length(radial_mask)
    radial_mask_idx(1,i)=find(radial_grid>=-radial_mask(i,1),1,'first');
    radial_mask_idx(1,ii)=find(radial_grid<=radial_mask(i,1),1,'last');
    axial_mask_idx(i,1)=find(axial_grid>=axial_mask(i,1),1,'first');
    axial_mask_idx(ii,1)=axial_mask_idx(i,1);
    ii=ii-1;
end

mask=poly2mask(axial_mask_idx,radial_mask_idx,size(zi,1),size(ri,2));

```

Sample points

```

alto_cc=abs(min(axial_mask))+abs(max(axial_mask));
ancho_cc=2*abs(max(radial_mask));

axial_sample_points(1,1)=min(axial_mask)+alto_cc/16;
axial_sample_points(1,2)=min(axial_mask)+alto_cc/4;
axial_sample_points(1,3)=min(axial_mask)+alto_cc/2;
axial_sample_points(1,4)=min(axial_mask)+alto_cc*3/4;
axial_sample_points(1,5)=min(axial_mask)+alto_cc/16;
axial_sample_points(1,6)=min(axial_mask)+alto_cc/4;
axial_sample_points(1,7)=min(axial_mask)+alto_cc/2;
axial_sample_points(1,8)=min(axial_mask)+alto_cc*3/4;
axial_sample_points(1,9)=min(axial_mask)+alto_cc/16;
axial_sample_points(1,10)=min(axial_mask)+alto_cc/4;
axial_sample_points(1,11)=min(axial_mask)+alto_cc/2;
axial_sample_points(1,12)=min(axial_mask)+alto_cc*3/4;

radial_sample_points(1,1)=-ancho_cc/4;
radial_sample_points(1,2)=-ancho_cc/4;
radial_sample_points(1,3)=-ancho_cc/4;
radial_sample_points(1,4)=-ancho_cc/4;
radial_sample_points(1,5)=0;
radial_sample_points(1,6)=0;
radial_sample_points(1,7)=0;
radial_sample_points(1,8)=0;
radial_sample_points(1,9)=ancho_cc/4;
radial_sample_points(1,10)=ancho_cc/4;
radial_sample_points(1,11)=ancho_cc/4;
radial_sample_points(1,12)=ancho_cc/4;

% figure
% hold on
% plot(radial_mask,axial_mask,'k-*')
% plot(-radial_mask,axial_mask,'k-*')
% xlim([-max(radial_mask) max(radial_mask)])
% ylim([min(axial_mask) max(axial_mask)])
% line(get(gca,'xlim'),[min(axial_mask)+alto_cc/16 min(axial_mask)+alto_cc/16])
% line(get(gca,'xlim'),[min(axial_mask)+alto_cc/4 min(axial_mask)+alto_cc/4])

```



```

% line(get(gca,'xlim'),[min(axial_mask)+alto_cc/2 min(axial_mask)+alto_cc/2])
% line(get(gca,'xlim'),[min(axial_mask)+alto_cc*3/4 min(axial_mask)+alto_cc*3/4])
% line([0+ancho_cc/4 0+ancho_cc/4],get(gca,'ylim'))
% line([0 0],get(gca,'ylim'))
% line([0-ancho_cc/4 0-ancho_cc/4],get(gca,'ylim'))
% plot(radial_sample_points,axial_sample_points,'r*')
% axis image

```

Datos PIV

```

load([pathPIV2, '/', 'AxialVelocity_data.mat']) % Incertidumbre de la medida PIV

for ik=1:length(CAD_position)

    data=dlmread(strcat(pathPIV2, '/', num2str(CAD_position(ik)), '.dat'), ' ', 1, 0);

    columns=(length(data)-2)/packagesize;

    r=zeros(columns,1);

    x=data(2:packagesize+1,1);

    U=zeros(packagesize,columns);
    V=zeros(packagesize,columns);
    I_U=zeros(packagesize,columns);
    I_V=zeros(packagesize,columns);

    for jk=1:columns
        pos=(jk-1)*packagesize+2;
        r(jk,1)=data(pos,2);
        U(:,jk)=data(pos:pos+packagesize-1,3); % Velocidad axial
        V(:,jk)=data(pos:pos+packagesize-1,4);
        I_U(:,jk)=data(pos:pos+packagesize-1,5); % U turbulence intencity (U std)
        I_V(:,jk)=data(pos:pos+packagesize-1,6); % V turbulence intencity (U std)
    end

    U(U==0 & v==0)=NaN; % Elements with velocity = 0 are changed to NaN
    V(U==0 & v==0)=NaN; % Elements with velocity = 0 are changed to NaN

    axial_vel_PIV{ik}=U;
    rad_vel_PIV{ik}=V;

    Module_PIV=sqrt(U.^2+V.^2);
    Module_CAD_PIV{ik}=Module_PIV;

    TKE_PIV=0.5*(I_U.^2+I_V.^2);
    TKE_PIV(TKE_PIV==0)=NaN;
    TKE_CAD_PIV{ik}=TKE_PIV;

    radial_PIV{ik}=r;
    axial_PIV{ik}=x;

    axis_velocity_PIV{ik}= interp2(r,x,U,0,x);

end

```

Datos CFD

```
% Inicializacion de variables
rad_vel_CFD_ciclos_todos=cell(length(CAD_position),1);
axial_vel_CFD_ciclos_todos=cell(length(CAD_position),1);
axial_vel_CFD_cl_ciclos_todos=cell(length(CAD_position),1);
Mo_CFD_ciclos_todos=cell(length(CAD_position),1);
TKEo_model_CFD_ciclos_todos=cell(length(CAD_position),1);
gasto_ciclos_todos=cell(length(CAD_position),1);

for ii=1:length(CAD_position)

    parfor ik=1:length(ciclos)

        files=dir([main_path, '/output/ciclo_', num2str(ciclos(ik)), '/*.col']); % Lista de
        ficheros de entrada

        data_CFD =
        dlmread([main_path, '/output/ciclo_', num2str(ciclos(ik)), '/', files(ii).name], ',', 2, 0);

        fileID =
        fopen([main_path, '/output/ciclo_', num2str(ciclos(ik)), '/', files(ii).name]);
        C = textscan(fileID, '%s', size(data_CFD, 2), 'HeaderLines', 1);
        fclose(fileID);
        header=C{1,1};

        x_CFD = data_CFD(:, strcmp(header, 'x'));
        y_CFD = data_CFD(:, strcmp(header, 'y'));
        z_CFD = data_CFD(:, strcmp(header, 'z'));
        u_CFD = data_CFD(:, strcmp(header, 'u')); % velocidad radial
        v_CFD = data_CFD(:, strcmp(header, 'v')); % velocidad radial
        w_CFD = data_CFD(:, strcmp(header, 'w')); % velocidad axial
        TKE_model = data_CFD(:, strcmp(header, 'tke')); % TKE modelada
        rho_CFD = data_CFD(:, strcmp(header, 'density')); % Densidad

        F =
        scatteredInterpolant(x_CFD, y_CFD, z_CFD, zeros(size(data_CFD, 1), 1), 'linear', 'none');

        % Campo de TKE modelada
        F.Values=TKE_model;
        TKEo_model = squeeze(F(xi, yi, zi)); % Mapa de TKE
        TKEo_model(mask==0)=NaN;
        TKEo_model_CFD_current(:, :, ik)=TKEo_model;

        % Campo de velocidad axial
        F.Values=w_CFD;
        axial_vel_CFD_cl_current(:, :, ik)=squeeze(F({0, 0, x*1e-3+inj_pos})); % Cambio de
        mm a m
        axial_vel_CFD = squeeze(F(xi, yi, zi)); % Mapa de velocidad axial
        axial_vel_CFD(mask==0)=NaN;
        axial_vel_CFD_current(:, :, ik)=axial_vel_CFD;

        axial_vel_CFD_gasto_current{ii, ik}=squeeze(F({lim_radial_gasto, lim_radial_gasto, lim_axia
        l_gasto}));

        % Campo de velocidad radial
        if strcmp(plano, 'CC') || strcmp(plano, 'B') || strcmp(plano, 'F')
            F.Values=v_CFD;
```

```

elseif strcmp(plano,'C') || strcmp(plano,'L') || strcmp(plano,'R')
    F.Values=u_CFD;
end

rad_vel_CFD = squeeze(F(xi,yi,zi)); % Mapa de velocidad radial
rad_vel_CFD(mask==0)=NaN;
rad_vel_CFD_current(:,:,ik)=rad_vel_CFD;

Module = sqrt(rad_vel_CFD.^2 + axial_vel_CFD.^2); % Mapa del modulo de la
velocidad
Mo_CFD_current(:,:,ik)=Module;

% Campo de densidad
F.Values=rho_CFD;

rho_CFD_gasto_current{ii,ik}=squeeze(F({lim_radial_gasto,lim_radial_gasto,lim_axial_gasto}));

end

TKEo_model_CFD_ciclos_todos{ii}=TKEo_model_CFD_current;
axial_vel_CFD_ciclos_todos{ii}=axial_vel_CFD_current;
axial_vel_CFD_cl_ciclos_todos{ii}=axial_vel_CFD_cl_current;
rad_vel_CFD_ciclos_todos{ii}=rad_vel_CFD_current;
Mo_CFD_ciclos_todos{ii}=Mo_CFD_current;

end

for ii=1:length(CAD_position)

    for ik=1:length(ciclos)

        for jj=1:length(lim_axial_gasto)

rho_CFD_gasto_current_plano=interp3(lim_radial_gasto,lim_radial_gasto,lim_axial_gasto,...
.
rho_CFD_gasto_current{ii,ik},lim_radial_gasto,lim_radial_gasto,lim_axial_gasto(jj));

axial_vel_CFD_gasto_current_plano=interp3(lim_radial_gasto,lim_radial_gasto,lim_axial_gasto,...
sto,...
axial_vel_CFD_gasto_current{ii,ik},lim_radial_gasto,lim_radial_gasto,lim_axial_gasto(jj)
);

gasto_current(jj)=nansum(nansum(rho_CFD_gasto_current_plano.*axial_vel_CFD_gasto_current
_plano*step^2));

end

gasto_current_ciclos(:,:,ik)=gasto_current;

end

gasto_ciclos_todos{ii}=gasto_current_ciclos;

```

```

end

% Inicializacion de variables
TKEo_solved_ciclos_CFD=cell(length(CAD_position),1);
rad_vel_CFD_ciclos_todos_mean=cell(length(CAD_position),1);
axial_vel_CFD_ciclos_todos_mean=cell(length(CAD_position),1);
Mo_CFD_ciclos_todos_mean=cell(length(CAD_position),1);
TKEo_model_CFD_ciclos_todos_mean=cell(length(CAD_position),1);
rad_vel_SamplePoints=zeros(length(CAD_position),length(axial_sample_points),length(ciclos)); % R: CAD C: point M: cycle
axial_vel_SamplePoints=zeros(length(CAD_position),length(axial_sample_points),length(ciclos));
Mo_vel_CFD_SamplePoints=zeros(length(CAD_position),length(axial_sample_points),length(ciclos));

% Calculo de TKE resuelta y promedios acumulados

% Inicializacion de variables
mean_Mo_vel_CFD_SamplePoints=zeros(length(CAD_position),length(axial_sample_points),length(ciclos));

for i=1:length(CAD_position)

    % Calculo de TKE resuelta

    TKEo_solved_ciclos_CFD{i}=0.5*(std(rad_vel_CFD_ciclos_todos{i},1,3).^2+std(axial_vel_CFD_ciclos_todos{i},1,3).^2);

    % Inicializacion de variables

    rad_vel_CFD_ciclos_todos_mean_current=zeros(size(axial_vel_CFD_ciclos_todos{1},1),size(axial_vel_CFD_ciclos_todos{1},2),length(ciclos));

    axial_vel_CFD_ciclos_todos_mean_current=zeros(size(axial_vel_CFD_ciclos_todos{1},1),size(axial_vel_CFD_ciclos_todos{1},2),length(ciclos));

    Mo_CFD_ciclos_todos_mean_current=zeros(size(axial_vel_CFD_ciclos_todos{1},1),size(axial_vel_CFD_ciclos_todos{1},2),length(ciclos));

    TKEo_model_CFD_ciclos_todos_mean_current=zeros(size(axial_vel_CFD_ciclos_todos{1},1),size(axial_vel_CFD_ciclos_todos{1},2),length(ciclos));

    % Calculo de las sumas acumuladas
    rad_vel_CFD_ciclos_todos_suma_currrent=cumsum(rad_vel_CFD_ciclos_todos{i},3);
    axial_vel_CFD_ciclos_todos_suma_currrent=cumsum(axial_vel_CFD_ciclos_todos{i},3);
    Mo_CFD_ciclos_todos_suma_currrent=cumsum(Mo_CFD_ciclos_todos{i},3);
    TKEo_model_CFD_ciclos_todos_suma_currrent=cumsum(TKEo_model_CFD_ciclos_todos{i},3);

    % Calculo de los promedios
    for j=1:length(ciclos)

        rad_vel_CFD_ciclos_todos_mean_current(:,:,j)=rad_vel_CFD_ciclos_todos_suma_currrent(:,:,j)/j;

        axial_vel_CFD_ciclos_todos_mean_current(:,:,j)=axial_vel_CFD_ciclos_todos_suma_currrent(:,:,j)/j;
    end
end

```

```

Mo_CFD_ciclos_todos_mean_current(:,:,j)=Mo_CFD_ciclos_todos_suma_currrent(:,:,j)/j;

TKEo_model_CFD_ciclos_todos_mean_current(:,:,j)=TKEo_model_CFD_ciclos_todos_suma_currrent(:,:,j)/j;

    % Evaluacion de convergencia del campo de velocidad promedio

mean_Mo_vel_CFD_SamplePoints(i,:,j)=interp2(zi,ri,Mo_CFD_ciclos_todos_mean_current(:,:,j),axial_sample_points,radial_sample_points);

    % Evaluacion de convergencia del campo de velocidad

rad_vel_SamplePoints(i,:,j)=interp2(zi,ri,rad_vel_CFD_ciclos_todos{i}(:,:,j),axial_sample_points,radial_sample_points);

axial_vel_SamplePoints(i,:,j)=interp2(zi,ri,axial_vel_CFD_ciclos_todos{i}(:,:,j),axial_sample_points,radial_sample_points);

Mo_vel_CFD_SamplePoints(i,:,j)=interp2(zi,ri,Mo_CFD_ciclos_todos{i}(:,:,j),axial_sample_points,radial_sample_points);

    end

    rad_vel_CFD_ciclos_todos_mean{i}=rad_vel_CFD_ciclos_todos_mean_current;
    axial_vel_CFD_ciclos_todos_mean{i}=axial_vel_CFD_ciclos_todos_mean_current;
    Mo_CFD_ciclos_todos_mean{i}=Mo_CFD_ciclos_todos_mean_current;

TKEo_model_CFD_ciclos_todos_mean{i}=TKEo_model_CFD_ciclos_todos_mean_current*TKE_factor;

end

for i=1:length(CAD_position)
    for j=2:length(ciclos)
        COV_Mo_CFD_ciclos_todos{i,1}(:,:,j-1)=100*std(Mo_CFD_ciclos_todos{i}(:,:,1:j),1,3)/mean(Mo_CFD_ciclos_todos{i}(:,:,1:j),3);
        ;
        sdt_Mo_CFD_ciclos_todos{i,1}(:,:,j-1)=std(Mo_CFD_ciclos_todos{i}(:,:,1:j),1,3);
    end
end
end

```

Plot mass flow rate along chamber axis

```

mkdir([ruta2save,'/Mass flow rate'])

for i=1:length(CAD_position)

    f1=figure('color','w'); % PIV frente a los primeros 4 promedios del CFD
    set(f1,'visible','off','units','normalized','position',[0 0 1 1])
    hold on

    for j=1:length(ciclos)
        plot(lim_axial_gasto-inj_pos,gasto_ciclos_todos{i}(:,:,j),'--b','Linewidth',1.5)
    end

    title(strcat(num2str(CAD_position(i)), ' CAD'))
    grid on
end

```

```

% Legend('PIV', 'CFD mean cycle')
xlabel('Axial coordinate [m]');
ylabel('Mass flow rate [kg/s]');
xlim([-0.08 0]);
ylim([-0.05 0.1]);
set(gca, 'XDir', 'reverse', 'FontSize', Font_Size, 'FontWeight', 'bold')

export_fig(f1, [ruta2save, '/Mass flow rate', '/', num2str(i), '_Mass flow
rate', '_', num2str(CAD_position(i)), '_CAD'])
close(gcf)

end

```

Plot axial velocity profiles

```

mkdir([ruta2save, '/Axial velocity profiles'])

for i=1:length(CAD_position)

    f1=figure('color','w'); % PIV frente a los primeros 4 promedios del CFD
    set(f1, 'visible', 'off', 'units', 'normalized', 'position', [0 0 1 1])
    hold on

    plot(axial_PIV{i}*1e-3, axis_velocity_PIV{i}, '-k', 'Linewidth', 3)
    plot(axial_PIV{i}*1e-3, mean(axial_vel_CFD_c1_ciclos_todos{i}, 3), '-b', 'Linewidth', 3)
    for j=1:length(ciclos)
        plot(axial_PIV{i}*1e-3, axial_vel_CFD_c1_ciclos_todos{i}(:, :, j), '--
b', 'Linewidth', 1.5)
    end

    Axialcoord_PIV=(PIV_data(i).Axialcoord_ms)*1e-3; % Cambio de mm a m
    unc_sup=PIV_data(i).AxialVel_ms+PIV_data(i).unc;
    unc_inf=PIV_data(i).AxialVel_ms-PIV_data(i).unc;

    fill(vertcat(Axialcoord_PIV, flipud(Axialcoord_PIV)), vertcat(unc_sup, flipud(unc_inf)), 'k'
, 'edgecolor', 'none', 'FaceAlpha', 0.2)

    title(strcat(num2str(CAD_position(i)), ' CAD'))
    grid on
    legend('PIV', 'CFD mean cycle')
    xlabel('Axial coordinate [m]');
    ylabel('Axial velocity [m/s]');
    xlim([-0.08 0]);
    ylim([-15 30]);
    set(gca, 'XDir', 'reverse', 'FontSize', Font_Size, 'FontWeight', 'bold')

    export_fig(f1, [ruta2save, '/Axial velocity profiles', '/', num2str(i), 'axial
velocity', '_', num2str(CAD_position(i)), '_CAD'])
    close(gcf)

end

```

Plots Sample points meeting 2017-05-08

```

for k=1:length(axial_sample_points)

```

```

mkdir([ruta2save, '/SamplePoints/Meeting_20170508/SamplePoint ', num2str(k)])

for i=1:length(CAD_position)

    % Velocity for each cycle and cumulative mean velocity

    f1=figure('Color','w');
    set(f1, 'visible', 'off', 'units', 'normalized', 'position', [0 0 1 1]);

    hold on
    plot(ciclos,squeeze(Mo_vel_CFD_SamplePoints(i,k,:)), '-
sr', 'Linewidth', 3, 'MarkerSize', 12) % Modulo de la velocidad para cada ciclo
    plot(ciclos,squeeze(mean_Mo_vel_CFD_SamplePoints(i,k,:)), '-
ob', 'Linewidth', 3, 'MarkerSize', 12)
    box on
    grid on
    set(gca, 'FontSize', Font_Size, 'FontWeight', 'bold')
    xlim([ciclos(1) ciclos(end)])
    ylim([0 25])
    set(gca, 'xtick', 2:ciclos(end))
    ylabel(['Velocity module [m/s] at ' num2str(CAD_position(i)) ' CAD'])
    xlabel('Cycle')
    legend('Module at each cycle', 'Module cumulative moving
average', 'Location', 'NorthWest')
    set(gca, 'FontSize', 20, 'FontWeight', 'bold')
    axes('position', [.77 .70 .2 .2]) % create a new pair of axes inside current
figure
    hold on
    plot(radial_mask, axial_mask-inj_pos, 'k', 'Linewidth', 2)
    plot(-radial_mask, axial_mask-inj_pos, 'k', 'Linewidth', 2)
    plot(radial_sample_points(k), axial_sample_points(k)-
inj_pos, 'k*', 'MarkerSize', 15)
    xlim([-max(radial_mask) max(radial_mask)])
    ylim([min(axial_mask) max(axial_mask)])
    axis image
    set(gca, 'XDir', 'reverse')
    set(gca, 'FontSize', 15, 'FontWeight', 'bold')
    box on

    export_fig(f1, [ruta2save, '/SamplePoints/Meeting_20170508/SamplePoint ', ...
num2str(k), '/', num2str(i), '_Vel_SP_', num2str(k), '_', num2str(CAD_position(i)), '_CAD'])
    close(gcf)

end
end

```

Plots variables ciclo a ciclo

```

cd([ruta2save, '/Velocity_CCV'])

% Modulo de la velocidad --> PIV vs todos los ciclos
vel_plots(radial_PIV, axial_PIV, rad_vel_PIV, axial_vel_PIV, Module_CAD_PIV, ...
ri, zi, Mo_CFD_ciclos_todos, rad_vel_CFD_ciclos_todos, axial_vel_CFD_ciclos_todos, plano)

```

Plots variables promediadas

```
cd([ruta2save, '/Velocity_mean'])

% Promedio del modulo de la velocidad --> PIV vs todos los ciclos
mean_vel_plots(radial_PIV,axial_PIV,rad_vel_PIV,axial_vel_PIV,Module_CAD_PIV,...

ri,zi,Mo_CFD_ciclos_todos_mean,rad_vel_CFD_ciclos_todos_mean,axial_vel_CFD_ciclos_todos_
mean,plano)

cd([ruta2save, '/Modeled TKE'])

% Promedio de la TKE modelada --> PIV vs todos los ciclos --> el CFD va multiplicado por
el escalar al final de la funcion
mean_TKE_modeled(radial_PIV,axial_PIV,TKE_CAD_PIV,ri,zi,TKEo_model_CFD_ciclos_todos_mean
,plano,3)

cd([ruta2save, '/Resolved TKE'])

% TKE resuelta
PIVvsCFD_TKE_resolved_plots_all(radial_PIV,axial_PIV,TKE_CAD_PIV,ri,zi,TKEo_solved_ciclo
s_CFD,plano)

cd([ruta2save, '/Velocity_COV'])
COV_plots(ri,zi,COV_Mo_CFD_ciclos_todos,plano)

cd([ruta2save, '/Velocity_std'])
std_plots(ri,zi,sdt_Mo_CFD_ciclos_todos,plano)
```

Guardar variables

```
save([main_path, '/statistics/up1cycle_', num2str(ciclos(end)), '_CC_plane/RawData_statisti
cs_COV.mat'], 'CAD_position', 'ciclos'...
, 'Mo_vel_CFD_SamplePoints', 'mean_Mo_vel_CFD_SamplePoints', 'radial_sample_points', 'axial_
sample_points', 'inj_pos'...
, 'radial_mask', 'axial_mask', 'radial_PIV', 'axial_PIV', 'rad_vel_PIV', 'axial_vel_PIV', 'Modu
le_CAD_PIV', 'ri', 'zi'...
, 'Mo_CFD_ciclos_todos', 'rad_vel_CFD_ciclos_todos', 'axial_vel_CFD_ciclos_todos', 'plano', '
Mo_CFD_ciclos_todos_mean'...
, 'rad_vel_CFD_ciclos_todos_mean', 'axial_vel_CFD_ciclos_todos_mean', 'TKE_CAD_PIV', 'TKEo_m
odel_CFD_ciclos_todos_mean'...
, 'TKEo_solved_ciclos_CFD', 'TKEo_model_CFD_ciclos_todos', 'axis_velocity_PIV', 'axial_vel_C
FD_c1_ciclos_todos',...
'lim_axial_gasto', 'gasto_ciclos_todos', 'COV_Mo_CFD_ciclos_todos', 'sdt_Mo_CFD_ciclos_todo
s')
```

III.2.2 Comparación modelo CFD anterior y actual

```
clearvars

ciclos_raw=1:15; % Ciclos originales
```



```

ciclos_cfd=ciclos_raw(1:end-1); % Eliminar el primer ciclo que llamamos ciclo ini
visibility='off';
save_fig='on';
folder2save='/data/combustion/CONVERGE/TFG_BORJA/2S_real/statistics/up1cycle_14_C_plane/
Compare Plots Real_Simple';
Font_Size=20;
Line_Width=3;

% Primero siempre el caso antiguo y después el nuevo
rutas{1}='/data/leopacpr/RIGEL/full_cycle/2S'; titles{1}='SIMPLE'; colors{1}=[0 0 1];
rutas{2}='/data/combustion/CONVERGE/TFG_BORJA/2S_real'; titles{2}='REAL'; colors{2}=[1 0
0];

CAD_TDC=720:360:360*ciclos_raw(end);

```

Experimental data

```

data_exp=xlsread('/data/leopacpr/CFD_Datasource/Caracterizaciones/PuntoPIV_presion.xls')
;

CAD_exp=data_exp(:,1);
CAD_exp_1=(-180:0.5:179.5+(360*15))';

P_exp=data_exp(:,3);
P_exp_1= repmat(P_exp,[16 1]);

P_exp_TDC=interp1(CAD_exp,P_exp,0);

```

Leer datos

```

for i=1:length(rutas)

    cd(rutas{i})

    % Pcam Region0 Cylinder
    list_thermoR0=dir('*thermo**region0*.out'); % Data is imported from thermo.out
    [~,idx_thermoR0] = sort([list_thermoR0.datenum]);
    thermoR0=dlmread(list_thermoR0(idx_thermoR0(1)).name, ',',5,0);
    fileID_thermoR0 = fopen(list_thermoR0(idx_thermoR0(1)).name);
    C_thermoR0 = textscan(fileID_thermoR0, '%s', size(thermoR0,2)+1, 'HeaderLines', 2); % +1
    porque la primera columna tiene el caracter #
    fclose(fileID_thermoR0);
    header_thermoR0=C_thermoR0{1,1};
    header_thermoR0(1)=[]; % Elimina el caracter # de la cabecera
    if length(list_thermoR0) > 1 % To concatenate thermo.out files
        for ii=2:length(list_thermoR0)

thermoR0=vertcat(thermoR0,dlmread(list_thermoR0(ii).name, ',',6,0));
            end
            empty=0;
            while empty==0 % Delete repeated instants of time
                check=diff(thermoR0(:,1));
                repeated=find(check<=0);
                empty=isempty(repeated);
                thermoR0(repeated,:)=[];
                clear check repeated
            end
        end
    end
end

```

```

end

cad_thermoR0{i}=thermoR0(:,strcmp(header_thermoR0,'Crank')); % [CAD]
P_thermoR0{i}=thermoR0(:,strcmp(header_thermoR0,'Pressure'))*10; % 10 [MPa] a [bar]
Tmax_thermoR0{i}=thermoR0(:,strcmp(header_thermoR0,'Max_Temp')); % [K]
Tmin_thermoR0{i}=thermoR0(:,strcmp(header_thermoR0,'Min_Temp')); % [K]
Pmax_thermoR0{i}=thermoR0(:,strcmp(header_thermoR0,'Max_Pres'))*10; % 10 [MPa] a
[bar]
mass_thermoR0{i}=thermoR0(:,strcmp(header_thermoR0,'Mass'))*1000; % 1000 [kg] a [g]

P_thermoR0_TDC{i}=interp1(cad_thermoR0{i},P_thermoR0{i},CAD_TDC);
mass_thermoR0_TDC{i}=interp1(cad_thermoR0{i},mass_thermoR0{i},CAD_TDC);

% Valores promedios en adm y esc
list_avg=dir('*area_avg_flow*.out'); % Data is imported from thermo.out
[~,idx_avg] = sort([list_avg.datenum]);
avg=dlmread(list_avg(idx_avg(1)).name, '',5,0);
fileID_avg = fopen(list_avg(idx_avg(1)).name);
C_avg = textscan(fileID_avg, '%s', size(avg,2)+1, 'HeaderLines',2); % +1 porque la
primera columna tiene el caracter #
fclose(fileID_avg);
header_avg=C_avg{1,1};
header_avg(1)=[]; % Elimina el caracter # de la cabecera
if length(list_avg) > 1 % To concatenate thermo.out files
    for ii=2:length(list_avg)
        avg=vertcat(avg,dlmread(list_avg(idx_avg(ii)).name, '',6,0));
    end
    empty=0;
    while empty==0 % Delete repeated instants of time
        check=diff(avg(:,1));
        repeated=find(check<=0);
        empty=isempty(repeated);
        avg(repeated,:)=[];
        clear check repeated
    end
end
cad_avg{i}=avg(:,strcmp(header_avg,'Crank')); % [CAD]
if i==1 % Caso viejo
mass_flow_rate_inflow{i}=avg(:,find(strcmp(header_avg,'Mass_Flow_Rate')==1,1,'first'));
% [Kg/s]

mass_flow_rate_outflow{i}=avg(:,find(strcmp(header_avg,'Mass_Flow_Rate')==1,1,'last'));
% [Kg/s]
    vel_inflow{i}=avg(:,find(strcmp(header_avg,'Avg_Velocity')==1,1,'first')); %
[m/s]
    vel_outflow{i}=avg(:,find(strcmp(header_avg,'Avg_Velocity')==1,1,'last')); %
[m/s]
    P_inflow{i}=avg(:,find(strcmp(header_avg,'Static_Pres')==1,1,'first'))/1e5; %
1e5 [Pa] a [bar]
    P_outflow{i}=avg(:,find(strcmp(header_avg,'Static_Pres')==1,1,'last'))/1e5; %
1e5 [Pa] a [bar]
elseif i==2 % Caso nuevo

mass_flow_rate_inflow{i}=avg(:,find(strcmp(header_avg,'Mass_Flow_Rate')==1,1,'last')); %
[Kg/s]

mass_flow_rate_outflow{i}=avg(:,find(strcmp(header_avg,'Mass_Flow_Rate')==1,1,'first'));

```

```

% [Kg/s]
    vel_inflow{i}=avg(:,find(strcmp(header_avg, 'Avg_Velocity')==1,1, 'last')); %
[m/s]
    vel_outflow{i}=avg(:,find(strcmp(header_avg, 'Avg_Velocity')==1,1, 'first')); %
[m/s]
    P_inflow{i}=avg(:,find(strcmp(header_avg, 'Static_Pres')==1,1, 'last'))/1e5; % 1e5
[Pa] a [bar]
    P_outflow{i}=avg(:,find(strcmp(header_avg, 'Static_Pres')==1,1, 'first'))/1e5; %
1e5 [Pa] a [bar]
    end

    clear list_thermoR0 list_avg

end

cd([folder2save])

```

Plot P at TDC

```

f1=figure('Color','w');
set(f1,'visible',visibility,'units','normalized','position',[0 0 1 1]);

hold on
for j=1:length(rutas)

plot(ciclos_cfd,P_thermoR0_TDC{j},'Linewidth',Line_Width,'color',colors{j},'marker','o')
end
line(get(gca,'Xlim'),[P_exp_TDC P_exp_TDC],'color','k','LineStyle','--',
'Linewidth',Line_Width)
ylim([32 35.5])
xlim([1 15])
box on
grid on
xlabel('Cycle')
ylabel('Pressure at TDC [bar]')
legend(titles,'location','southwest')
set(gca,'FontSize',Font_Size,'FontWeight','bold')

if strcmp(save_fig,'on')
    export_fig(f1,'Pcam_TDC')
    set(f1,'visible','on')
    savefig(f1,'Pcam_TDC.fig');
    close(gcf)
end

```

Plot mass at TDC

```

f1=figure('Color','w');
set(f1,'visible',visibility,'units','normalized','position',[0 0 1 1]);

hold on
for j=1:length(rutas)

plot(ciclos_cfd,mass_thermoR0_TDC{j},'Linewidth',Line_Width,'color',colors{j},'marker','o')
end
ylim([2.8 3.5])

```

```

xlim([1 15])
box on
grid on
xlabel('Cycle')
ylabel('Trapped mass [g]')
legend(titles, 'location', 'southwest')
set(gca, 'FontSize', Font_Size, 'FontWeight', 'bold')

if strcmp(save_fig, 'on')
    export_fig(f1, 'Trapped_mass')
    set(f1, 'visible', 'on')
    savefig(f1, 'Trapped_mass.fig');
    close(gcf)
end

```

Plot Pcam

```

f1=figure('Color','w');
set(f1,'visible',visibility,'units','normalized','position',[0 0 1 1]);

hold on
for j=1:length(rutas)
    plot(cad_thermoR0{j},P_thermoR0{j},'Linewidth',Line_width,'color',colors{j})
end
plot(CAD_exp_1,P_exp_1,'k','Linewidth',Line_width)
xlim([410 5500])
ylim([1 2])
box on
grid on
xlabel('[CAD]')
ylabel('Pcam [bar]')
legend(titles, 'location', 'southwest')
set(gca, 'FontSize', Font_Size, 'FontWeight', 'bold')

if strcmp(save_fig, 'on')
    export_fig(f1, 'Pcam_zoom')
    ylim([1 35])
    export_fig(f1, 'Pcam')
    set(f1, 'visible', 'on')
    savefig(f1, 'Pcam.fig');
    close(gcf)
end

```

Plot mass flow rate

```

%Plot inflow mass flow rate
f1=figure('Color','w');
set(f1,'visible','on','units','normalized','position',[0 0 1 1]);

%subplot(1,2,1)
hold on
for j=1:length(rutas)
    plot(cad_avg{j},mass_flow_rate_inflow{j},'Linewidth',Line_width,'color',colors{j})
end
xlim([410 5500])
ylim([-0.4 1])
box on

```

```

grid on
xlabel('[CAD]')
ylabel('Mass flow rate [Kg/s]')
title('Average at inflow')
legend(titles,'location','SouthWest')
set(gca,'FontSize',Font_Size,'FontWeight','bold')

if strcmp(save_fig,'on')
    export_fig(f1,'Inflow mass flow rate')
    set(f1,'visible','on')
    savefig(f1,'Inflow mass flow rate.fig');
    close(gcf)
end

%Plot outflow mass flow rate
f1=figure('Color','w');
set(f1,'visible','on','units','normalized','position',[0 0 1 1]);

%subplot(1,2,2)
hold on
for j=1:length(rutas)
    plot(cad_avg{j},mass_flow_rate_outflow{j},'Linewidth',Line_width,'color',colors{j})
end
xlim([410 5500])
ylim([-0.4 1])
box on
grid on
xlabel('[CAD]')
title('Average at outflow')
legend(titles,'location','SouthWest')
set(gca,'FontSize',Font_Size,'FontWeight','bold')

if strcmp(save_fig,'on')
    export_fig(f1,'Outflow mass flow rate')
    set(f1,'visible','on')
    savefig(f1,'Outflow mass flow rate.fig');
    close(gcf)
end

```

III.2.3 Unión imágenes varios planos

```

clearvars

folder2save='/data/combustion/CONVERGE/TFG_BORJA/2S_real/statistics/png2video';

SpacingHoriz=0.01;

ruta_1='/data/combustion/CONVERGE/TFG_BORJA/2S_real/statistics/Frames_MotorMaqueta/Plano
_X';

list_1=dir([ruta_1,'/*.png']);

```

```

for i=1:length(list_1)
    name_x=list_1(i).name;
    name_1(i)=str2double(name_x(1:end-4));
end
[~,idx_1]=sort(name_1);

ruta_2='/data/combustion/CONVERGE/TFG_BORJA/2S_real/statistics/Frames_MotorMaqueta/Plano
_Y';

list_2=dir([ruta_2,'/*.png']);

for i=1:length(list_2)
    name_x=list_2(i).name;
    name_2(i)=str2double(name_x(1:end-4));
end
[~,idx_2]=sort(name_2);

for i=1:length(list_1)

    f1=figure('Color','w');
    set(f1,'visible','on','units','normalized','position',[0 0 1 1])

    subaxis(1,2,1,'SpacingHoriz',SpacingHoriz) % (x,y,z) filas x, columnas y, cuadrante
z
%     subplot(1,2,1)
    imagen = imread([ruta_1,'/',list_1(idx_1(i)).name]);
    imagen(200:end,1:150,1)=256;
    imagen(200:end,1:150,2)=256;
    imagen(200:end,1:150,3)=256;
    imshow(imagen(:,20:900,:))
%     imshow(imagen)

    subaxis(1,2,2,'SpacingHoriz',SpacingHoriz)
%     subplot(1,2,2)
    imagen= imread([ruta_2,'/',list_2(idx_2(i)).name]);
    imagen(200:end,1:70,:)=1;
    imshow(imagen(:,400:end,:))
%     imshow(imagen)

    hold on

    export_fig(gcf,[folder2save,'/',num2str(i)])
    close(gcf)

end

```

III.2.4 Imágenes a vídeo

```

clearvars
sort_files='Y'; % Sort files by creation date. Not useful if images were created with a
parfor loop
%path='/data/leopacpr/ECN5/Chemical_mechanisms/pfc_clustercmt03/SA_Nara_185K_minus/Unste
ady/Ignition/Post-processing_YC_CMT/A4';
path='/data/combustion/CONVERGE/TFG_BORJA/2S_real/statistics/png2video';

```

```

name_png=path(find(path=='/',1,'last')+1:end);

cd(path)

%list_png=dir(['*',name_png,'*.png']);
list_png=dir('*.png');

for i=1:length(list_png)
    name_x=list_png(i).name;
    name_1(i)=str2double(name_x(1:end-4));
end
[~,idx]=sort(name_1);

imageNames = {list_png.name}';
% [~,idx] = sort([list_png.datenum]);

outputVideo=videowriter(fullfile(path,[name_png,'_Video_sequence.avi']));
outputVideo.FrameRate=1;
open(outputVideo)

for i=1:length(imageNames)
    i
    if strcmp(sort_files,'Y')
        img_1 = imread(fullfile(path,imageNames{idx(i)}));
    elseif strcmp(sort_files,'N')
        img_1 = imread(fullfile(path,imageNames{i}));
    end
    writeVideo(outputVideo,img_1)
end

close(outputVideo)

```

III.3 EnSight

III.3.1 Introducción

EnSight es un programa de post-procesamiento que facilita en gran medida la lectura de resultados de un flujo. Tiene la capacidad de mostrar el flujo en una geometría en 2D y 3D. Además, se puede elegir que variables mostrar, como la velocidad, la temperatura, entre otros. Se tiene la elección de los contornos en el que se quieren visualizar las variables y de qué forma hacerlo.

En los siguientes apartados se mostrarán los diferentes pasos a seguir para conseguir visualizar el flujo del interior del motor en 2D en diferentes planos. Estos planos son los planos ortogonales a X e Y.

III.3.2 Primeros pasos en EnSight

Para poder abrir un archivo de Converge con EnSight, primero hay que convertir los archivos a .case. Hay una opción en Converge, *post-processing*, que nos facilita esta conversión. Esto lo podemos encontrar en Converge abajo a la izquierda, un botón de opciones que dice *Post-Processing 3D*. En esta opción marcaremos que variables se quieren transformar para poder visualizarlas en el entorno 3D de EnSight.

Después de obtener el archivo .case, también nos aparece para cada intervalo de tiempo del estudio diferentes archivos que corresponden a la geometría (el archivo más pesado) y para las variables que hemos elegido.

Cuando abrimos EnSight se debe elegir la opción *Open* y buscar donde se tenga guardado el archivo .case. Cuando este archivo se abra aparece la geometría del modelo, distinguiendo todos los contornos que se delimitaron en Converge. A la izquierda de la pantalla se encuentran diferentes opciones:

- Arriba está la opción de elegir manualmente en que momento de la simulación se quiere estar o arrancar la simulación y que muestre la simulación como si fuera un vídeo.
- En el medio diferencia todos los *boundaries* o contornos. Si se quiere mostrar o ocultar un contorno se tiene que pinchar la "X" que este en la misma línea.
- Abajo se encuentran diferentes opciones, todas de visualización. Entre ellas están la visualización del ángulo del cigüeñal, el tiempo, colores de las partes de la simulación, formato de la leyenda, etc.

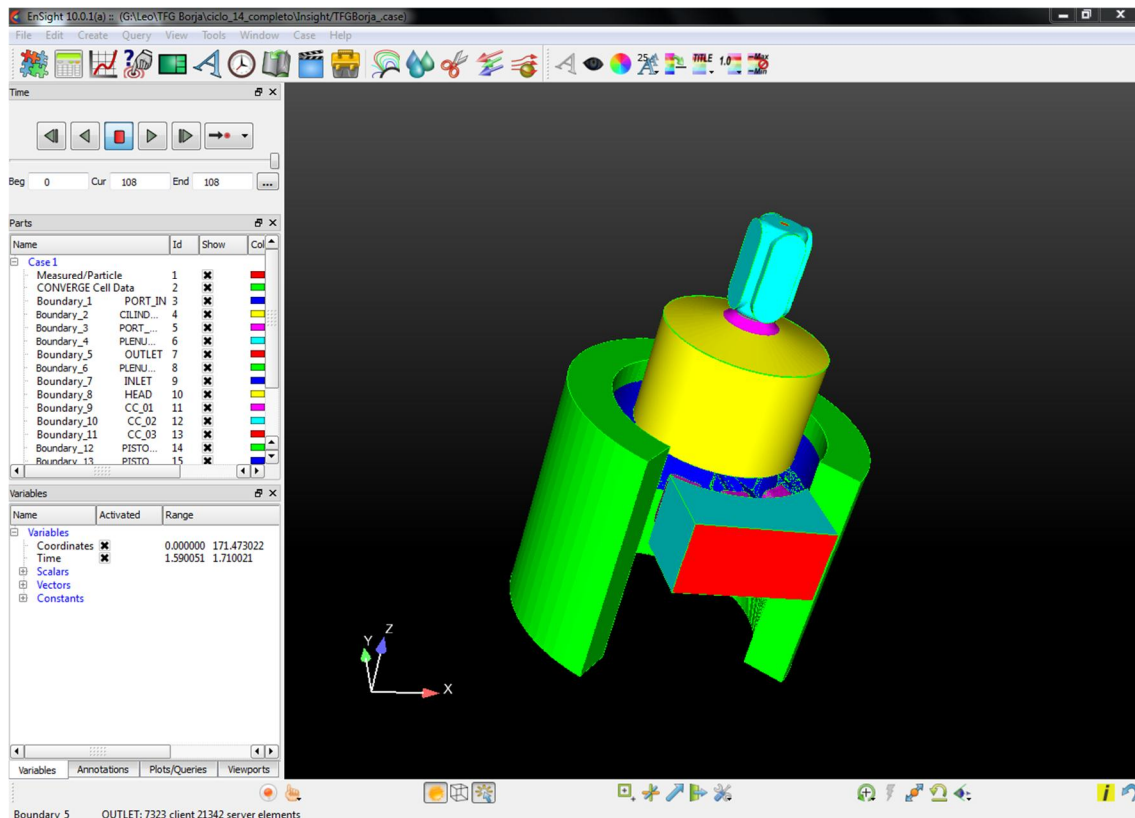


Figura III.0.1: Representación de la geometría actual en EnSight

III.3.3 Creación de *clips*

Para poder realizar un estudio más sencillo de la simulación se suele hacer planos que corten el motor en toda su superficie, es decir, planos ortogonales al X, Y y, si fuera necesario, el plano Z. Para poder crear estos planos (en inglés, *clips*) se tienen que seguir una serie de pasos bastante accesibles.

Primero, hay que pinchar la línea que corresponde a CONVERGE Cell Data. Está línea se encuentra en el mismo lugar que las correspondencias con los contornos. Después de haber pinchado esa línea, hay que dirigirse a la barra de herramientas en la parte de arriba de la pantalla y buscar un símbolo que representa unas tijeras (*clips*). Al elegir esa herramienta aparece el menú para crear el plano que quiera el usuario. Dentro de este menú se puede modificar las siguientes variables:

- Nombre del *clip*.
- Los planos de referencia que se usarán en la creación del plano.
- El plano al que será ortogonal el *clip*.
- Distancia del origen del plano de referencia al que se quiere el *clip*.

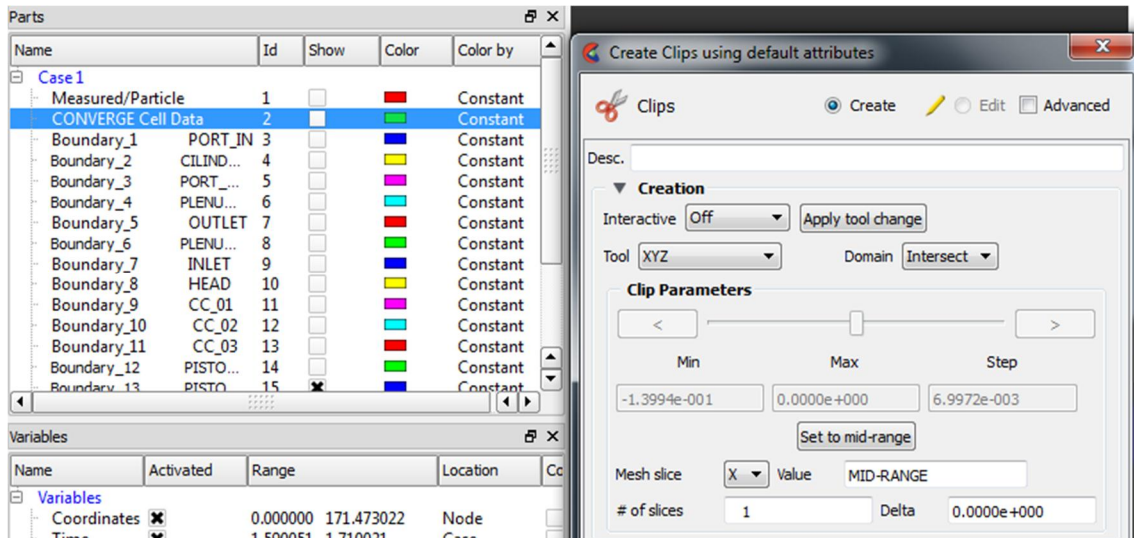


Figura III.0.2: Ventanas para la creación de un plano de corte

Al elegir todo lo que el usuario necesite se pincha *Create*. Para poder visualizar y diferenciar el nuevo plano que se ha creado es necesario ocultar todos los *boundaries* y límites de cada uno y, así, poder estudiar el nuevo plano sin ninguna interferencia. El nuevo *clip* se verá gris, sin ninguna información, solamente la forma que tiene. Para obtener información de este plano se debe elegir una variable, donde esas opciones se encontrarán abajo a la izquierda de la pantalla. Cuando ya se sepa que variable se puede estudiar se arrastrará esta a la línea del *clip* que se encuentra con el resto de las líneas de los *boundaries*. Al hacer eso, el nuevo plano adquirirá colores, siendo el azul el valor más bajo y el rojo el más alto. Además, abajo a la derecha de la pantalla aparecerá una leyenda con el nombre y los valores de la variable elegida.

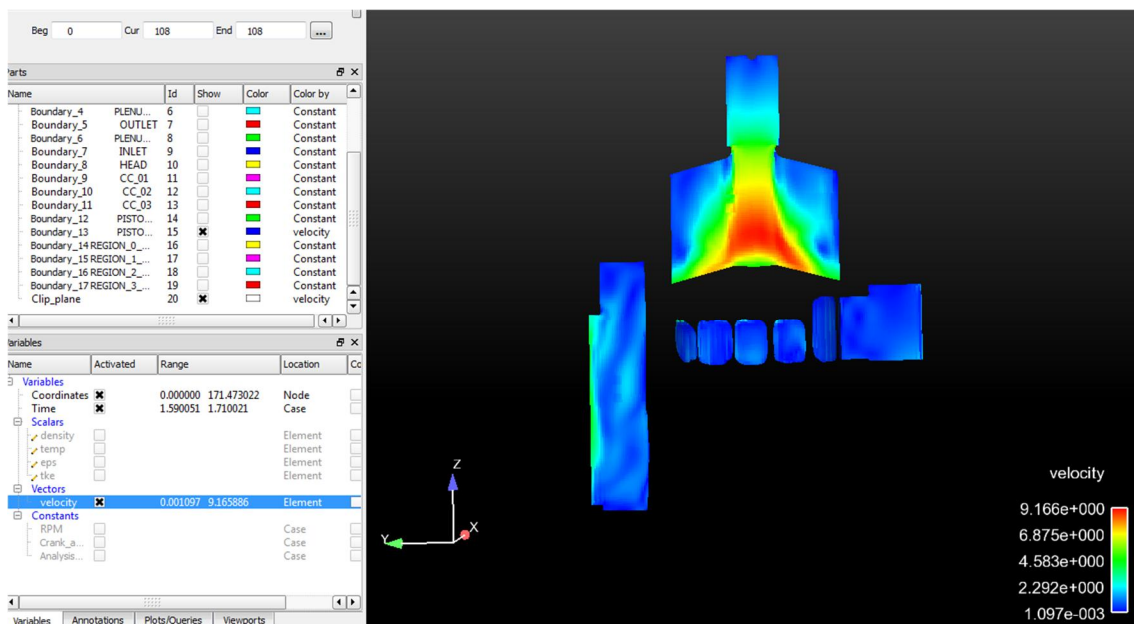


Figura III.0.3: Representación de la velocidad en el plano de corte ortogonal a X

III.3.4 Configuración de la representación

A partir de haber creado los planos necesarios, solo queda representar la información de una forma clara. Esto significa mostrar ángulos de cigüeñal, tiempo de simulación, colores claros para la comprensión de las variables, entre otros.

Para mostrar variables como el tiempo o el ángulo hay que dirigirse abajo a la izquierda, donde se ubican todas las variables del estudio. Ahí se debe cambiar de pestaña y elegir *Annotations*. En esa pestaña aparecen el tiempo y el *Crank Angle* (ángulo de cigüeñal) y con un click derecho elegir la opción de mostrar la variable y aparecerán las variables que se quieran.

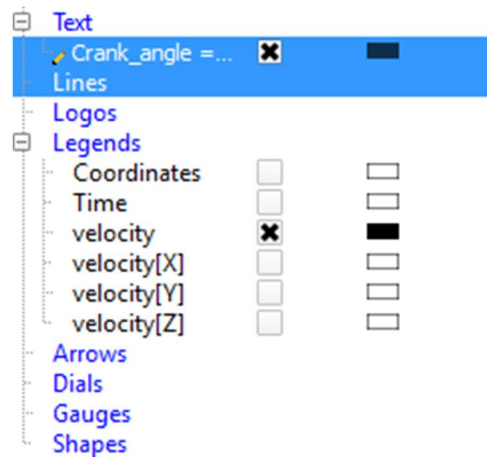


Figura III.0.4: Menú de las variables representadas en el modelo

RPM = 500.00
Crank_angle = 5130.06

Figura III.0.5: Representación de los valores de RPM y ángulo de cigüeñal (CAD)

Otra cosa necesaria es cambiar el color de fondo de la simulación, ya que dejar el color predeterminado de Ensign (negro) puede inducir confusión. Para cambiar el color se tiene que elegir la pestaña *Viewports* donde se ubican las variables. Hacer clic derecho y elegir la opción de *Create/edit Viewports*. Aparecerá un menú con diferentes opciones de configuración, las más importantes son la elección de un color sin degradados, en el programa *Constant* y la otra elegir el color. Para ello hay que dirigirse a *Mix color...* y seleccionar el color que se desee.

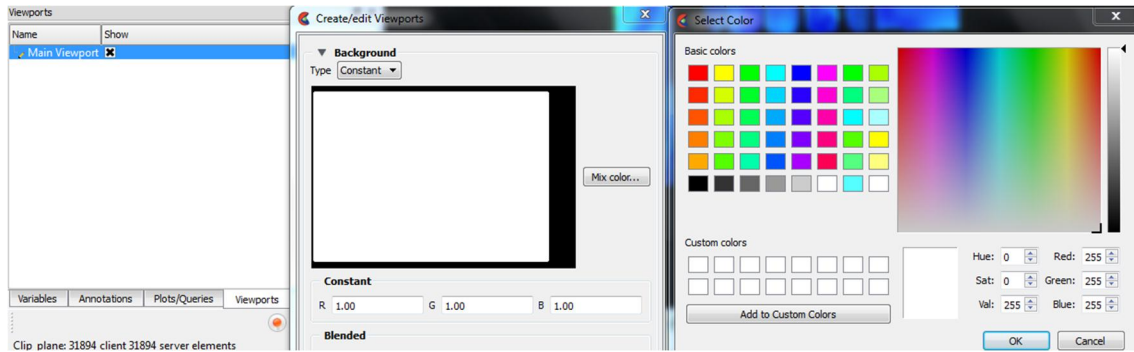


Figura III.0.6: Ventanas para la modificación del color de fondo

Para la modificación de las leyendas es un procedimiento muy parecido al anterior. En vez de ir a la pestaña de *Viewports*, se irá a la pestaña de *Plots*. En esa pestaña hay que hacer clic derecho en la variable y seleccionar *Edit palettes*. Cuando aparece el menú se puede modificar los colores de la intensidad de la variables y los colores de las letras y bordes de la leyenda. A parte se puede editar los valores mínimo y máximo que se representen en los planos, en esta opción se eligen números entero para dar facilidad a otros usuarios a comprender mejor el estudio.

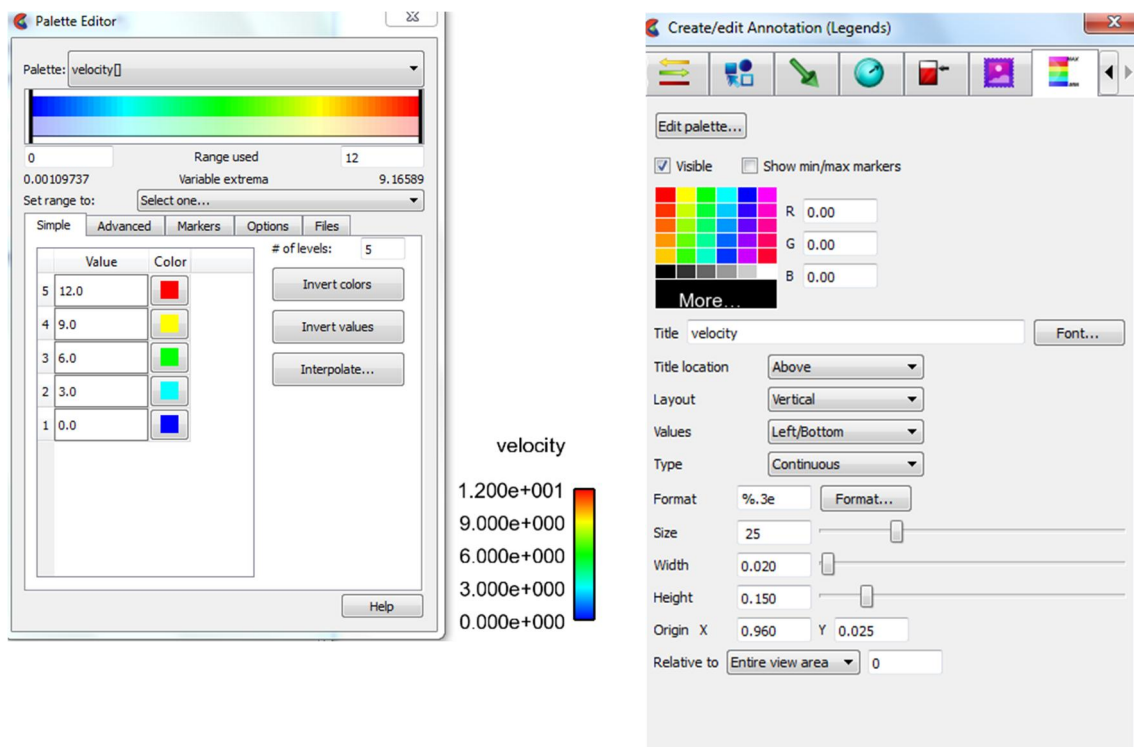


Figura III.0.7: Ventanas para la modificación del formato de las leyendas de las variables

III.3.5 Grabación de la simulación en imágenes

Para una mayor facilidad en el formato y el tratamiento de las imágenes y vídeos, se ha elegido hacer capturas de la simulación por grados y representarlas en imágenes en formato PNG. Gracias a una rutina de Matlab mostrada en apartados anteriores podremos juntar capturas de diferentes planos, pero del mismo ángulo de cigüeñal y mostrar estas capturas grado a grado formando un vídeo.

Primero, se selecciona la opción *Export* dentro de *File* arriba a la derecha de la pantalla. Después de elegir esa opción aparece un menú donde se tiene que seleccionar *Play* y *Reset* en *Solution Time* para poder iniciar la simulación y vaya haciendo capturas a todos los grados. En la opción avanzada se puede elegir la resolución de las imágenes, donde se opta por un HD 1080p. Por último, se selecciona el formato de imagen que se desee. Para ello hay que pinchar en *Prefix select...* y, del menú que se abre, elegir el formato de imagen PNG.

Es importante que antes de darle el OK al *Save animation* hay que arrancar la simulación para que vayan pasando todos los intervalos de tiempo, es decir, todos los grados del cigüeñal y así Ensignht pueda ir guardando todos los intervalos en imágenes.

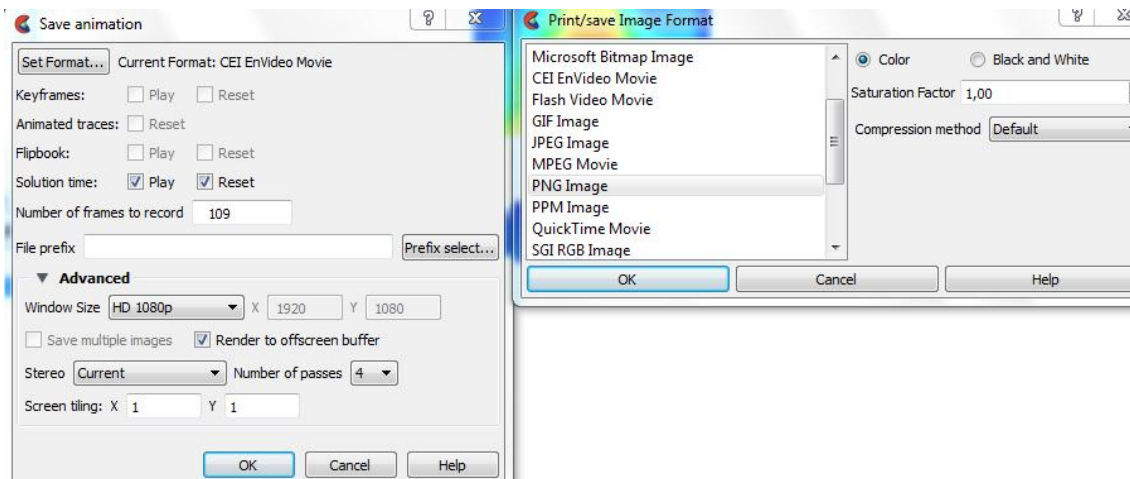


Figura III.0.8: Ventanas para elegir el formato de la imagen

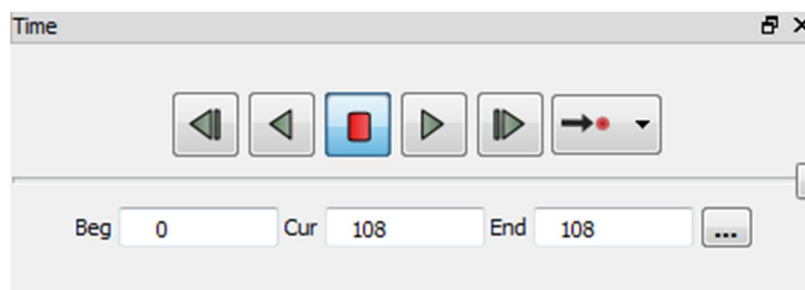


Figura III.0.9: Ventana para la elección del momento de representación del modelo o de ver la simulación como un vídeo

PRESUPUESTO

A. Introducción

En el último documento de este trabajo se expone el presupuesto del proyecto para su realización. Este presupuesto es un poco peculiar porque hay que considerar que no hay la confección de un producto, solamente el estudio del desarrollo de una investigación.

Hay que considerar la mano de obra que ha participado en este proyecto, dándole prioridad al Ingeniero de Proyecto, es decir, al alumno que ha desarrollado todos los pasos para alcanzar unos resultados. Además, hay que contar con el coste de las licencias de los programas usados, mantenimiento de los aparatos y el uso de unas instalaciones.

B. Gastos

B.1 Gastos en la mano de obra

En este apartado se designan los costes de los ingenieros que trabajan en el actual proyecto. Este equipo de ingenieros lo componen el Ingeniero de Proyecto, alumno que ha realizado la mayoría de la investigación, el Ingeniero Tutor del Proyecto, persona encargada de guiar e inspeccionar el trabajo que vaya desarrollando al alumno y el Ingeniero Director del Proyecto, quien supervisa y dictamina los pasos que tiene que seguir el equipo.

Los gastos en el sueldo del Ingeniero del Proyecto se emplean el mismo que si fuera un Ingeniero Mecánico Junior. El Ingeniero Tutor del Proyecto se emplea los gastos de un Director de Proyectos Senior. El sueldo en bruto del Ingeniero Director de Proyecto puede ser un sueldo parecido al del Ingeniero del Proyecto.

La Tabla [B.3] muestra el coste anual de un Ingeniero Mecánico Junior:

Salario bruto anual	25000 €
Seguridad social (23.6%)	5900 €
Desempleo (5.5%)	1375 €
Form. Profesional (0.6%)	150 €
Form. Gar. Profesional (0.3%)	75 €
Coste anual empresa	32500 €

Tabla B.1: Coste anual del salario de un Ingeniero Mecánico Junior

Considerando doce meses de trabajo y 22 días laborales por mes, con jornadas diarias de 8 horas, se alcanza un valor de 15.39 €/h

La Tabla [B.4] muestra el coste anual de un Ingeniero Tutor de Proyecto:

Salario bruto anual	37500 €
Seguridad social (23.6%)	8850 €
Desempleo (5.5%)	2062.50 €
Form. Profesional (0.6%)	225 €
Form. Gar. Profesional (0.3%)	112.50 €
Coste anual empresa	48750 €

Tabla B.2: Coste anual del salario de un Ingeniero Tutor de Proyecto

Considerando doce meses de trabajo y 22 días laborales por mes, con jornadas diarias de 8 horas, se alcanza un valor de 23.08 €/h.

La Tabla [B.5] muestra el coste anual de un Director de Proyectos Senior:

Salario bruto anual	50000 €
Seguridad social (23.6%)	11800 €
Desempleo (5.5%)	2750 €
Form. Profesional (0.6%)	300 €
Form. Gar. Profesional (0.3%)	150 €
Coste aneal empresa	65000 €

Tabla B.3: Coste anual del salario de un Ingeniero Director de Proyecto

Considerando doce meses de trabajo y 22 días laborales por mes, con jornadas diarias de 8 horas, se alcanza un valor de 30.78 €/h.

B.2 Gastos en licencias

El coste de las licencias lo conforman el precio de las licencias en el momento de su compra y la amortización que le da el trabajador que las emplee. Como ya se ha comentado anteriormente, los programas usados en este proyecto son Converge CFD, EnSight y Matlab.

El cálculo del coste total se tiene en cuenta el coste, la vida útil, la amortización y su dedicación. Todas estas variables se relacionan a través de la siguiente fórmula:

$$\text{Coste total} = \text{Coste} \cdot \frac{\text{Amortización (\%)}}{100} \cdot \frac{\text{Vida útil}}{12} \cdot \frac{1}{\text{Vida útil}} \quad (\text{B.1})$$

En la Tabla [B.1] se indican los valores de todas las variables y el coste final:

Licencia	Coste anual [€/año]	Amortización [meses]	Dedicación [%]	Coste total [€]	
Matlab	2000	4	100	666.70	
EnSight	10000	4	50	1666.70	
Converge	30000	4	50	5000	
Coste total				7333.4	

Tabla B.4: Coste, vida útil, amortización, dedicación y coste total de las licencias usadas en este proyecto

Se puede deducir a partir de la Tabla [B.1] que el coste total de las licencias de los programas usados es 7333.4€ durante el tiempo que ha durado en el proyecto, que ha sido cuatro meses. De esta forma, suponiendo que el equipo trabaja 8 horas al día, 22 días al mes, se puede llegar al resultado de que el coste es 10.42 €/h aproximadamente.

B.2 Gastos en equipos informáticos

Se hace un cálculo idéntico al apartado de gastos en licencias, donde ahora se quiere averiguar el coste total de los aparatos informáticos. Se tiene en cuenta el coste, la vida útil, la amortización y su dedicación. Los aparatos informáticos usados en este proyecto son un ordenador y una estación de cálculo formada por varios procesadores.

En la Tabla [B.2] se indican los valores de todas las variables y el coste final:

Licencia	Coste [€]	Vida útil [años]	Amortización [meses]	Dedicación [%]	Coste total [€]
Ordenador	800	3	4	100	88.90
Estación de cálculo	50000	5	4	50	1666.70
Coste total					1755.6

Tabla B.5: Coste, vida útil, amortización, dedicación y coste total del equipo informático usado en este proyecto

Si cogemos los valores obtenidos en la Tabla [B.2] se puede llegar a la conclusión que, si los ingenieros encargados del proyecto trabajan durante 22 días al mes durante 4 meses, 8 horas al día, dividiendo el coste total entre las horas trabajadas en total, se obtiene un coste de 0.25 €/h.

B.3 Gastos energéticos

En este apartado se asignan los gastos de energía, principalmente electricidad, que producen el uso de ordenador e instalaciones del centro. Para el gasto energético hay que valorar la potencia que necesita cada aparato y su rendimiento.

La Tabla [B.6] muestra la potencia y rendimiento de cada sistema:

Sistema	Rendimiento	Potencia [kW]
Ordenador	0.5	0.5
Climatización	50	5
Iluminación	0.2	0.2
Total	-	5.7

Tabla B.6: Rendimiento y potencia [kW] de los sistemas eléctricos

C. Cuadro de precios descompuestos

C.1 Introducción

En el cuadro de precios descompuestos se detalla el reparto de costes por trabajador, es decir, el coste del Ingeniero Director del Proyecto, el Ingeniero Tutor del Proyecto y el Ingeniero del Proyecto.

C.2 Ingeniero Director del Proyecto

Concepto	Ud.	Descripción	Rto. [u/h]	Precio unitario [€/ud]	Importe [€/h]
Ingeniero de proyecto	H	Hora trabajada del ing.	1	30.78	30.78
Equipos	H	Amortización de equipos	1	2.50	2.50
Energía	kWh	Energía eléctrica consumida por los aparatos	5.7	0.14	0.80
TOTAL [€/h]					34.08

Tabla C.1: Precio desglosado del coste total para el Ingeniero Director del Proyecto

C.3 Ingeniero Tutor de Proyecto

Concepto	Ud.	Descripción	Rto. [u/h]	Precio unitario [€/ud]	Importe [€/h]
Ingeniero de proyecto	H	Hora trabajada del ing.	1	23.08	23.08
Energía	kWh	Energía eléctrica consumida por los aparatos	5.7	0.14	0.80
TOTAL [€/h]					23.88

Tabla C.2: Precio desglosado del coste total para el Ingeniero Tutor del Proyecto

C.4 Ingeniero de Proyecto

Concepto	Ud.	Descripción	Rto. [u/h]	Precio unitario [€/ud]	Importe [€/h]
Ingeniero de proyecto	H	Hora trabajada del ing.	1	15.39	15.39
Equipos	H	Amortización de equipos	1	2.50	2.50
Licencias	H	Amortización de licencias	1	10.42	10.42
Energía	kWh	Energía eléctrica consumida por los aparatos	5.7	0.14	0.80
TOTAL [€/h]					29.11

Tabla C.3: Precio desglosado del coste total para el Ingeniero Mecánico Junior

D. Cuadro de mediciones

Nº Orden	Descripción	Medición [h]	Precio [€/h]	Importe [€]
01	Ingeniero de Proyecto	650	29.11	18921.5
02	Ingeniero Director del Proyecto	100	34.08	1408
03	Ingeniero Tutor del Proyecto	50	23.88	1194
Total				21523.50

Tabla D.1: Coste total del proyecto sin IVA

E. Resumen del presupuesto

En el resumen del presupuesto se extiende la Tabla [anterior] con el beneficio industrial (6%) y con el IVA (impuesto de valor añadido) del 21%.

Total presupuesto de ejecución material	21523.50 €
6% Beneficio industrial	1291.41 €
Total presupuesto de inversión	22814.91 €
21% de IVA	4791.13 €
TOTAL PRESUPUESTO BASE DE LICITACIÓN	27606.04 €

Tabla E.1: Coste total del proyecto con IVA

El presupuesto base de licitación aumenta a la cantidad de **VEINTISIETE MIL SEISCIENTOS SEIS EUROS CON CUATRO CÉNTIMOS**.