



Universitat Rovira i Virgili (URV) y Universitat Oberta de Catalunya (UOC)

Máster en Ingeniería Computacional y Matemática

Trabajo Final de Máster

Título del Trabajo Final

Simulación numérica del flujo y la transferencia de calor en un sistema de refrigeración líquida para un motor eléctrico

Nombre Estudiante: José Ángel Moya Muriana

Nombre Directora: María Dolores Puigjaner Riba

Nombre Codirector: Patricio Bohórquez Rodríguez de Medina

Fecha de Entrega: 13/09/2020

El /a Dr./Dra. (nombre) Dolors Puigjaner Riba, certifica que el/la estudiante (nombre) José Ángel Moya Mariana ha elaborado el trabajo bajo su dirección y autoriza la presentación de esta memoria para la su evaluación.

Firma del director/a:



Esta obra está sujeta a una licencia de Reconocimiento-  
No Comercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## Licencias alternativas

(elegir alguna de las siguientes y sustituir la de la página anterior)

### A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

### B) GNU Free Documentation License (GNU FDL)

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

### C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

# FICHA DEL TRABAJO FINAL DE MASTER

<b>Título del trabajo:</b>	Simulación numérica del flujo y la transferencia de calor en un sistema de refrigeración líquida para un motor eléctrico
<b>Nombre del autor:</b>	D. José Ángel Moya Muriana
<b>Nombre del director/a:</b>	Dra. María Dolores Puigjaner Riba
<b>Fecha (mm/aaaa):</b>	13/09/2020
<b>Titulación:</b>	Máster en Ingeniería Computacional y Matemática
<b>Área del Trabajo Final:</b>	CFD
<b>Idioma del trabajo:</b>	Castellano
<b>Palabras clave:</b> <small>Máximo 3 palabras clave (propuestas por el autor del TFM o en base a listados, tesauros, etc. y validadas por el director)</small>	<ol style="list-style-type: none"> <li>1. Vehículos de movilidad personal (VMP)</li> <li>2. Marco de referencia múltiple (MRF)</li> <li>3. Transferencia de calor</li> </ol>
<b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>El auge de los vehículos eléctricos de movilidad personal (VMP) y su relevancia social es a día de hoy evidente, presentándose como una de las soluciones viables al transporte urbano en grandes ciudades.</p> <p>Independientemente de la naturaleza del mismo, los principios de funcionamiento y la geometría de los actuales motores eléctricos usados para su propulsión presentan la misma deficiencia. Esta es el sobrecalentamiento (así como la pérdida en rendimiento y vida útil que ello supone) ante condiciones de estrés como, por ejemplo, una elevada temperatura ambiental.</p> <p>La principal causa es la carencia de sistemas de refrigeración. Por ello, haciendo uso de un simulador multifísico para transferencia de calor fluido-sólido, se desea proponer un sistema de refrigeración modelo eficiente y caracterizar el flujo de calor resultante. Estará basado en la incorporación de un aceite dieléctrico que acelerará el enfriamiento de la tapa metálica del motor. Teniendo en cuenta los elevados requerimientos computacionales del problema fluidomecánico a resolver, se empleará cálculo en paralelo basado en multicore CPUs para resolver las ecuaciones de Navier-Stokes para la fase fluida y la ecuación de conducción de calor en los elementos sólidos.</p>	

**Abstract (in English, 250 words or less):**

The rise of electric vehicles for personal mobility (PMV) and their social relevance is evident today, presenting itself as one of the viable solutions to urban transport in large cities.

Regardless of its nature, the operating principles and geometry of the current electric motors used for its propulsion, all present the same deficiency. This is the overheating under stress conditions such as, for example, a high ambient temperature or a prolonged period of activity.

The main cause is the lack of cooling systems. Therefore, using a multiphysics simulator for fluid-solid heat transfer, it is desired to propose an efficient model cooling system as well as characterize the resulting heat flow. It will be based on the incorporation of a dielectric oil that will accelerate the cooling of the metal cover of the motor. Taking into account the high computational requirements of the fluidomechanical problem to be solved, parallel calculation based on multicore CPUs will be used to solve the Navier-Stokes equations for the fluid phase and the heat conduction equation in solid elements.

# Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	2
1.5 Breve resumen de productos obtenidos.....	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Antecedentes de la micromovilidad y VMP.....	5
2.1 Introducción.....	5
2.2 La importancia de la micromovilidad.....	6
2.3 Las nuevas ciudades.....	10
2.4 Micromovilidad en la era pos-COVID.....	12
3. Modelo CAD.....	15
3.1 Diseño 3D.....	15
3.2 Simplificación geométrica.....	18
4. Recopilación y tratamiento de datos.....	20
4.1 Introducción.....	20
4.2 Velocidad angular.....	21
4.3 Temperatura.....	24
5. Caso MRFSimpleFoam.....	25
5.1 Introducción.....	25
5.2 Algoritmo simpleFoam.....	26
5.3 Modelo de turbulencia $k-\epsilon$ .....	27
5.4 Marco de Referencia Múltiple (MRF).....	28
5.5 Mallado modelo MRFSimpleFoam.....	29
5.6 Pre-procesado caso MRFSimpleFoam.....	34
5.6.1 Directorio “time”.....	34
5.6.2 Directorio “constant”.....	36
5.6.3 Directorio “system”.....	37
5.7 Resultados caso MRFSimpleFoam.....	41
6. Caso chtMRFMultiRegionFoam.....	47
6.1 Introducción.....	47
6.2 Algoritmo chtMultiRegionFoam.....	48
6.3 Mallado modelo chtMultiRegionFoam.....	50
6.3 Pre-procesado caso chtMultiRegionFoam.....	52
6.3.1 Directorio “0”.....	52
6.3.2 Directorio “constant”.....	54
6.3.3 Directorio “system”.....	56
6.4 Resultados caso chtMultiRegionFoam.....	57
7. Refrigeración líquida directa del motor.....	61
7.1 Introducción.....	61
7.2 Líquidos dieléctricos.....	62
7.3 Propiedades termofísicas.....	63
7.4 Resultados.....	65
8. Conclusiones.....	68

9. Glosario.....	71
9.1 Acrónimos.....	71
9.2 Nomenclatura.....	72
10. Bibliografía.....	75
11. Anexos.....	78

## Lista de figuras y tablas

<i>Figura 1:</i> Distancia de viaje [5].....	5
<i>Figura 2:</i> Datos globales de la NASA temperatura global para 2100 [8].....	6
<i>Figura 3:</i> Distribución según la distancia del viaje en automóvil (US) [15].....	8
<i>Figura 4:</i> Ventas anuales de bicicletas eléctricas vs coches eléctricos [16].....	8
<i>Figura 5:</i> Ventas anuales en la UE de algunos vehículos ligeros [17].....	9
<i>Figura 6:</i> Crecimiento de la población urbana y rural [18].....	10
<i>Figura 7:</i> Incremento anual de la población urbana [19].....	11
<i>Figura 8:</i> Ciudades por tamaño de asentamiento urbano, 2030 [18].....	12
<i>Figura 9:</i> Imagen interactiva 3D del monociclo.....	16
<i>Figura 10:</i> Simplificación geométrica n.º 1.....	18
<i>Figura 11:</i> Simplificación geométrica n.º 2.....	18
<i>Figura 12:</i> Simplificación geométrica n.º 3.....	19
<i>Figura 13:</i> Simplificación geométrica n.º 4.....	19
<i>Figura 14:</i> Ruta GPS recorrida durante el testeo [26].....	20
<i>Figura 15:</i> Representación de datos grabados durante el testeo [27].....	20
<i>Figura 16:</i> Histograma frecuencia relativa de la velocidad de rotación.....	22
<i>Figura 17:</i> Sección en el plano XY para el conjunto motor.....	23
<i>Figura 18:</i> Marcos de referencia estacionario (absoluto) y móvil (relativo).....	28
<i>Figura 19:</i> Imagen interactiva 3D de la geometría caso MRFSimpleFoam.....	31
<i>Figura 20:</i> Sección en el plano XY mostrando las regiones giratorias.....	34
<i>Figura 21:</i> Residuales $\omega=15$ rad/s, $Re=1830$ (flujo laminar).....	41
<i>Figura 22:</i> Residuales $\omega=20$ rad/s, $Re=2440$ (flujo transitorio bajo).....	41
<i>Figura 23:</i> Residuales $\omega=25$ rad/s, $Re=3050$ (flujo transitorio medio).....	42
<i>Figura 24:</i> Residuales $\omega=30$ rad/s, $Re=3660$ (flujo transitorio alto).....	42
<i>Figura 25:</i> Sección contorno de presiones plano YZ para $\omega=30$ rad/s.....	43
<i>Figura 26:</i> Distribución de presiones para $\omega=30$ rad/s.....	43
<i>Figura 27:</i> Líneas de corriente de velocidad para $\omega=30$ rad/s.....	44
<i>Figura 28:</i> Sección YZ contorno de velocidades para $\omega=15$ rad/s.....	44
<i>Figura 29:</i> Sección YZ contorno de velocidades para $\omega=20$ rad/s.....	45
<i>Figura 30:</i> Sección YZ contorno de velocidades para $\omega=25$ rad/s.....	45
<i>Figura 31:</i> Sección YZ contorno de velocidades para $\omega=30$ rad/s.....	45
<i>Figura 32:</i> Sección YZ contorno vorticidad para $\omega=30$ rad/s.....	46
<i>Figura 33:</i> Par frente a temperatura de un motor BLDC genérico [36].....	48
<i>Figura 34:</i> Imagen interactiva 3D de la geometría caso chtMultiRegionFoam.....	51
<i>Figura 35:</i> Condición flujo de calor pared externa.....	53
<i>Figura 36:</i> Evolución de los residuales del dominio fluido.....	57
<i>Figura 37:</i> Evolución de los residuales del cubremotor.....	57
<i>Figura 38:</i> Sección YZ del campo velocidad del fluido, $\omega=16$ rad/s.....	58
<i>Figura 39:</i> Distribución de la temperatura del fluido con líneas de corriente... ..	58
<i>Figura 40:</i> Sección YZ de la distribución de temperatura del motor.....	59
<i>Figura 41:</i> Sección YZ del contorno densidad.....	60
<i>Figura 42:</i> Sección YZ del contorno presión.....	60
<i>Figura 43:</i> Densidad éster natural y ajuste lineal.....	63
<i>Figura 44:</i> Conductividad térmica éster natural y ajuste lineal.....	63
<i>Figura 45:</i> Calor específico éster natural y ajuste cuadrático.....	64
<i>Figura 46:</i> Viscosidad cinemática éster natural y ajuste cúbico.....	64
<i>Figura 47:</i> Evolución de los residuales del dominio fluido.....	65



<i>Figura 48:</i> Evolución de los residuales para el cubremotor.....	66
<i>Figura 49:</i> Sección YZ del campo de velocidad para el éster natural.....	66
<i>Figura 50:</i> Sección YZ para el contorno densidad del éster natural.....	66
<i>Figura 51:</i> Sección YZ de la distribución de temperatura del motor.....	67
<i>Tabla 1:</i> Diagrama de Gantt.....	3
<i>Tabla 2:</i> Tabla de distribución de frecuencias.....	21
<i>Tabla 3:</i> Número de Reynolds asociado a la velocidad angular del rotor.....	23
<i>Tabla 4:</i> Flujo de calor expulsado al exterior a través del cubremotor.....	67

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Las nuevas tecnologías han favorecido la aparición de soluciones de movilidad urbana que favorecen los desplazamientos personales mediante el auxilio de nuevos modelos de vehículos dotados de motor eléctrico. Unido a este desarrollo tecnológico, la micromovilidad se ha hecho patente en las grandes ciudades aportando soluciones de transporte que resuelve el primer y último tramo de los trayectos diarios: el que discurre entre la vivienda del usuario y el medio de transporte primario (como autobús, coche o tren) y el que se realiza entre el medio de transporte primario y el destino.

Un estudio realizado por McKinsey Center for Future Mobility [1], revela que aproximadamente el 60% de los viajes que se hacen en coche en todo el mundo son de menos de ocho kilómetros, por lo que durante estos trayectos se podrían aplicar distintas soluciones de micromovilidad.

Este nuevo concepto de movilidad surge a raíz de la gran evolución que han sufrido las ciudades, a causa de los cambios demográficos y al crecimiento de la población mundial. Según Seat [2], en el año 2030, el número de megaciudades (áreas urbanas con más de diez millones de habitantes) se incrementará hasta 43, es decir, 12 más que ahora. Además, se espera que en el año 2040 el 65% de la población viva en ciudades. Estos datos revelan la importancia de la micromovilidad en el futuro para poder garantizar una movilidad urbana mucho más efectiva, cómoda y que respete el medio ambiente.

Las soluciones de micromovilidad incluyen vehículos ligeros, como patinetes, bicicletas y otros medios de transporte de reducido tamaño, que normalmente son traccionados por motores de carácter eléctrico.

Estos motores encargados de transformar la energía eléctrica procedente de la batería en energía mecánica presentan una limitación y es la eliminación de sus pérdidas en forma de calor durante un uso prolongado o debido a una alta temperatura ambiente. La principal causa de esta insuficiencia en los motores eléctricos viene otorgada por la carencia de sistemas de refrigeración, que permitan una dinámica evacuación de las pérdidas originadas por el motor en forma de calor.

## 1.2 Objetivos del Trabajo

Las etapas concretadas para el desarrollo del trabajo son enumeradas a continuación:

1. Descripción de la importancia de los VMP y la micromovilidad
2. Modelado CAD del monociclo eléctrico

3. Recopilación, procesamiento e interpretación de los datos de funcionamiento del sistema
4. Implementación del marco de referencia múltiple para definir la rotación del aire en el interior de la rueda
5. Análisis de la transferencia de calor del motor refrigerado por aire
6. Estudio de la transferencia de calor en el motor refrigerado con un aceite dieléctrico

### **1.3 Enfoque y método seguido**

Dada la gran cantidad de vehículos que se están comercializando a día de hoy, existe una tipología bastante variopinta de VMP (hoverboards, monociclos, patinetes, segway, bicicletas eléctricas, scooters, etc), así como de electrónica de potencia encargada de su propulsión.

Sin embargo, por su relevancia en las pérdidas energéticas, y por ende en la generación interna de calor, nos centraremos en un vehículo de considerable potencia energética. Y es que existen monociclos que pueden estar dotados de baterías de gran capacidad, aptas para otorgar potencias de hasta 2000 W en valor nominal y 4000 W como valor pico, como es el caso de la rueda eléctrica KINGSONG modelo KS-18XL [3] que estudiaremos.

De modo que en este trabajo tomaremos como hipótesis de partida un producto comercial ya existente en el mercado, que disfruta de una significativa relevancia social entre los seguidores del mundo de los monociclos eléctricos por sus buenas prestaciones y propondremos una mejora del sistema actual de refrigeración, lo cual podría incentivar más aún su uso al hacerlo más eficiente (mayor autonomía) y dotarlo de una mayor vida útil.

### **1.4 Planificación del Trabajo**

Los recursos, tanto de índole tangible como virtuales, empleados para el desarrollo del trabajo constan de:

1. Monociclo KS-XL18 definido como patrón físico de medición y validación
2. El software CATIA para el diseño CAD de la rueda eléctrica
3. El paquete LibreOffice para: el desarrollo del documento, hojas de cálculo, presentación y edición de imágenes
4. Aplicación WheelLog para la monitorización de datos, recolección y tratamiento de información
5. La web Plotly para la representación de datos



Durante el apartado 5, se describen las ecuaciones de Navier-Stokes que gobiernan el flujo confinado enclavado en el sistema de referencia inercial, para después, mediante la determinación de la fuerza de Coriolis, transformar estas ecuaciones de conservación a un marco con movimiento relativo. Por último, el caso MRF es implementado y simulado en OpenFOAM para distintas velocidades angulares del rotor.

En orden a determinar el comportamiento térmico del motor eléctrico, un análisis multifísico añadiendo la ecuación de conservación de la energía a los diversos dominios que lo integran, es desarrollado. Para ello, se define la temperatura máxima en el devanado, conocida de acuerdo al ensayo real.

Finalmente, detectado el coeficiente de convección  $h_c$  como la variable de mayor atractivo a modificar en nuestro sistema, se introducen los líquidos dieléctricos como medios eficientes de refrigeración. Elegido el éster natural como el refrigerante estrella; por su inmejorable biodegradabilidad y capacidad de reciclado, así como, por sus propiedades fisicotérmicas; se demuestra a través de simulación su mayor rendimiento térmico frente al aire.

## 2. Antecedentes de la micromovilidad y VMP

### 2.1 Introducción

El analista Horace Dediu fue la primera persona en acuñar el término Micromovilidad [4], durante el discurso que pronunció en 2017 en la Micromobility Summit en el evento Techfestival en Copenhague, para referirse implícitamente a la capacidad de movimiento (movilidad) a través de medios minimalistas (micro).

Horace asoció el concepto al significado de microcomputación. La microcomputación se definió en torno a la naturaleza singular de su núcleo de procesamiento y nada más. Por analogía, la micromovilidad puede definirse en torno a lo que es singular acerca de su propósito: mover a un ser humano. De modo que su propósito es ofrecer la máxima libertad de movilidad y su minimalismo permite hacerlo de la manera menos impactante.

La micromovilidad aún no tiene una definición consensuada universalmente. Sin embargo, parece ampliamente aceptado el hecho de que se trata de un rango de pequeños y ligeros dispositivos operando a velocidades típicamente inferiores a 25 km/h y usado idealmente para viajes de hasta 10 km [5].

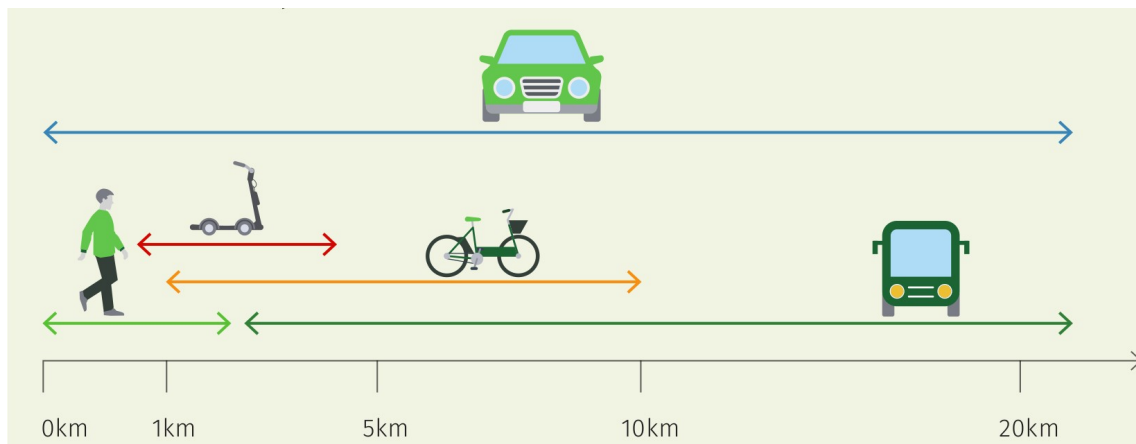


Figura 1: Distancia de viaje [5]

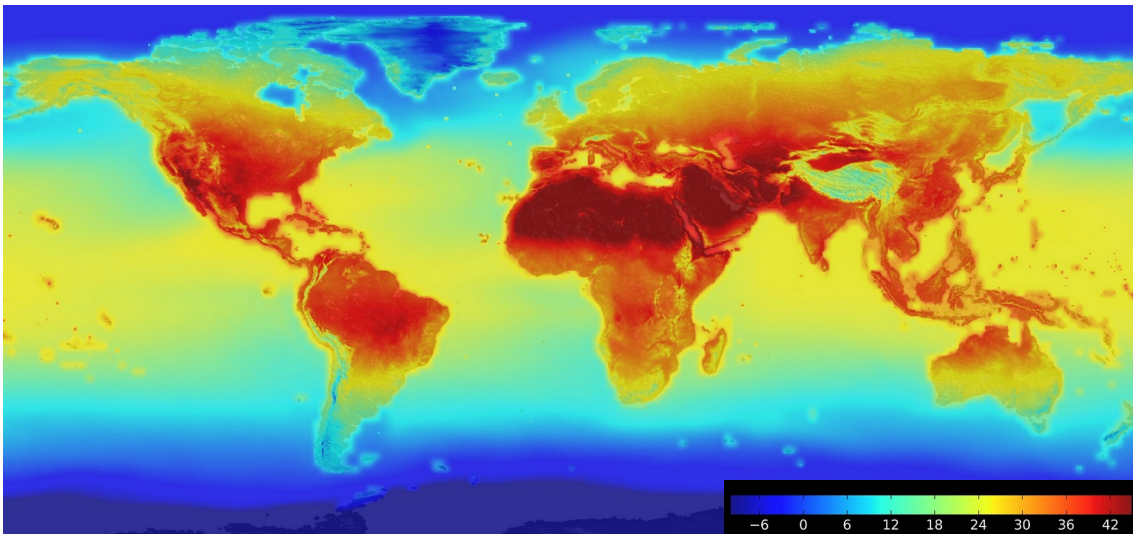
Las definiciones iniciales también establecen que la condición principal para la inclusión en la categoría de micromovilidad es un peso bruto del vehículo de menos de 500 kg. No obstante, la definición ha evolucionado para excluir dispositivos con motores de combustión interna y aquellos con velocidades máximas superiores a 45 km/h.

Los dispositivos de micromovilidad pueden ser humanos o eléctricos, y pueden ser de propiedad privada o disponibles a través de una flota compartida. Destacan como vehículos de movilidad personal o micromovilidad eléctrica: las scotters, bicicletas, monociclos y plataformas.

La introducción de la mayoría de las personas a los vehículos de micro movilidad es el scooter eléctrico. En China y en Japón las bicicletas eléctricas son extremadamente populares. Y aunque todavía no son muchas las start-ups que han intentado monetizar los monociclos y patinetes, están empezando a cobrar bastante importancia debido a su portabilidad [6].

## 2.2 La importancia de la micromovilidad

El mundo se está calentando y científicos de todo el mundo apuntan a que tenemos 10 años para reducir las emisiones anuales a la mitad si queremos evitar un cambio climático catastrófico [7].



*Figura 2: Datos globales de la NASA que combinan mediciones históricas con datos de simulaciones climáticas para proporcionar pronósticos de cómo la temperatura global podrían cambiar para 2100 [8]*

El transporte por carretera es el área de emisiones de más rápido crecimiento [9] y uno de las más difíciles de combatir, pues el transporte sustenta toda la economía y, a medida que avanza, también lo hace la mayoría de los demás.

La respuesta de la industria automotriz ante este problema ha sido trasladar la producción hacia automóviles eléctricos. Estas empresas son expertas en la fabricación y venta de automóviles, por lo que, naturalmente, su solución a la crisis climática es fabricar automóviles que sean lo más limpios posible y con este propósito se han proporcionado miles de millones en subvenciones para ayudar a dicha transición.

También con esta pretensión han surgido nuevas líneas de investigación, innovación y desarrollo energético en el ámbito de la Unión Europea (UE), orientadas a promover la movilidad sostenible, eficiente y accesible para todos. Despuntan instrumentos de financiación como: L'Instrument Financier pour l'Environnement (LIFE), Horizonte 2020 (H2020); Mecanismo Conectar Europa (CEF); Fondo Europeo de Desarrollo Regional (FEDER), entre otros.

Dentro del programa LIFE destaca el proyecto U-MOB (Red Europea de Movilidad Sostenible en la Universidad) cuyo objetivo principal es la creación de una red universitaria que facilite el intercambio y la transferencia de conocimientos sobre buenas prácticas de movilidad sostenible entre las universidades europeas [10].

El desarrollo de un nuevo sistema de transporte inteligente, ecológico e integrado es uno de los siete retos sociales identificados por la Comisión Europea en el programa Horizonte 2020 [11]. El objetivo fundamental de este reto es lograr un sistema europeo de transporte más eficaz en el uso de los recursos, que sea respetuoso con el clima y el medio ambiente y que funcione con seguridad y sin fisuras en beneficio de todos los ciudadanos, la economía y la sociedad.

La finalidad de las convocatorias CEF Transport es ayudar a completar la Red Transeuropea de Transporte (RTE-T) de alto rendimiento, sostenible y eficiente a largo plazo, así como, optimizar la integración e interconexión de los distintos medios de transporte y mejorar su interoperabilidad [12].

Promocionar el transporte sostenible y eliminar los cuellos de botella en las infraestructuras del transporte clave es uno de los once objetivos temáticos de los fondos FEDER [13]. Para ello se marca las siguientes prioridades de inversión:

- Respalda un espacio único europeo del transporte multimodal mediante la inversión en la RTE-T
- Mejorar la movilidad regional mediante la conexión de los nodos secundarios y terciarios a la infraestructura RTE-T
- Desarrollar y mejorar sistemas de transporte respetuosos con el medio ambiente (también de bajo nivel de ruido) e hipocarbónicos.

Sin embargo, a pesar de este esfuerzo por parte de algunas instituciones europeas, incluso las proyecciones más optimistas respecto a la adopción de automóviles eléctricos, emitidas por parte de la Agencia Internacional de Energía (AIE), se sitúan solamente en un 30% de las ventas totales para 2030 [14]. Estas mismas estimaciones calculan que la flota mundial de automóviles no estará totalmente electrificada hasta al menos 2050.

¿Por qué importa esto? Principalmente debido que la industria ha estado analizando este problema de manera incorrecta. Los automóviles se compran para ser utilizados en viajes y si observamos esos viajes se verá que la mayoría de ellos son cortos. Ahí es donde también están las emisiones.

En este contexto, se hace fundamental la aparición de modos de micromovilidad como los VMP eléctricos. Las bicicletas y scooters eléctricos se adaptan perfectamente a viajes pequeños, mientras que solo emiten el 1-2% de lo que hace un automóvil. Dicho de otra manera, 50 paseos en scooters



producen la misma cantidad de emisiones de CO<sub>2</sub> que un solo viaje en automóvil.

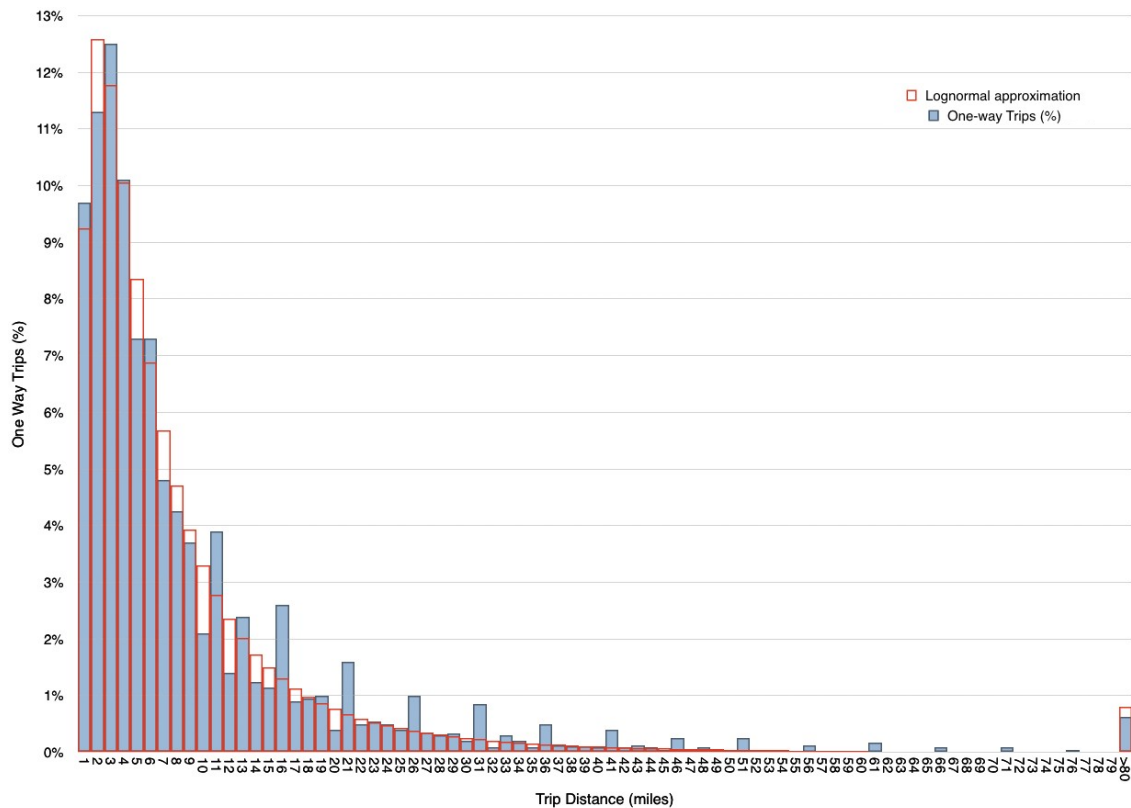


Figura 3: Distribución según la distancia del viaje en automóvil (US) [15]

Los VMP no son perfectos: no son tan seguros, ni tan resistentes a la intemperie, ni tan "prestigiosos" como los automóviles. Aún no. Pero vencieron a los automóviles en otras comparativas importantes: son más rápidos en áreas urbanas congestionadas, más baratos por km y más divertidos en general (tal y como muestran los datos de ventas).

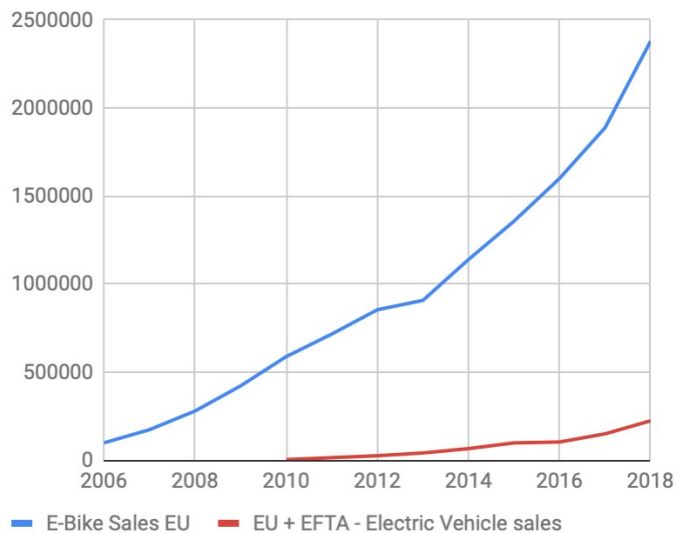


Figura 4: Ventas anuales de bicicletas eléctricas frente a coches eléctricos [16]

Las bicicletas eléctricas ya superan en ventas a los automóviles eléctricos por un factor de 10:1 en Europa y 20:1 en China. Además, la micromovilidad ha sido capaz de lograr esta escala con casi ningún apoyo gubernamental, infraestructura adecuada o incentivos fiscales.

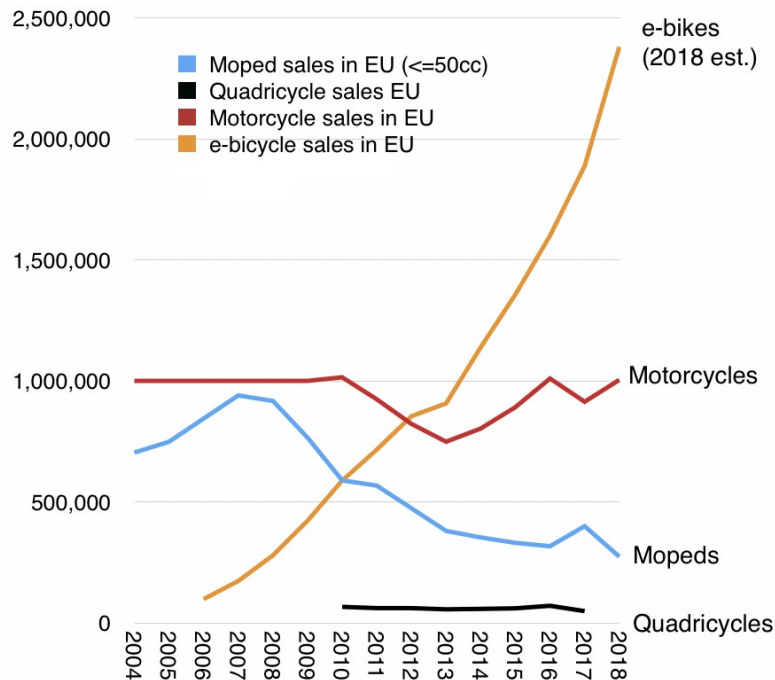


Figura 5: Ventas anuales en la UE de algunos vehículos ligeros [17]

Sin embargo, la micromovilidad sigue siendo ignorada en gran medida por los activistas climáticos que buscan reducir las emisiones. Si los activistas quieren una victoria clara con los gobiernos que buscan opciones de transporte rentables y respetuosas con el clima, deberían adoptar los VMP, así como apoyar la infraestructura de la manera más agresiva posible.

Las inversiones en servicios e infraestructura de transportes benefician directamente a los ciudadanos y a las empresas. La movilidad inteligente, el transporte multimodal, el transporte limpio y la movilidad urbana en general deberían posicionarse como políticas prioritarias para las instituciones gubernamentales durante los próximos años.

La micromovilidad eléctrica tiene un enorme potencial para mejorar los sistemas de transporte urbano sostenible. Los recientes avances tecnológicos, el rápido crecimiento y la adopción de servicios de movilidad compartida han permitido escalar pequeños vehículos. Como resultado, los VMP podrían servir como punto de partida a una red de transporte sostenible más amplia, permitiendo que más personas confíen en caminar, montar en bicicleta y en el transporte público para más viajes.

## 2.3 Las nuevas ciudades

Uno de los aspectos más singulares de la micromovilidad es que es una colección de modos y aplicaciones inherentemente urbanos. Históricamente, el transporte personal motorizado ha sido diseñado para reducir grandes distancias y las primeras aplicaciones de navegación, ferrocarril, carretera y vuelo acortaron las brechas entre ciudades, es decir, los viajes extraurbanos. La micromovilidad es, junto con el tránsito peatonal, una solución para las zonas urbanas.

Es una tendencia constante. La urbanización ha estado aumentando desde que las ciudades fueron concebidas por primera vez y hemos tenido un tremendo crecimiento en el siglo pasado. La población urbana del mundo ha crecido rápidamente de 751 millones en 1950 a 4.200 millones en 2018 y llegará a 6.700 millones a mediados de este siglo.

Las ciudades son grandes lugares a proyectar y todas las ciudades juntas son un concepto que es difícil de visualizar o considerar cómo afrontar. Por lo que parece hacerse necesario un análisis en el que las ciudades se clasifiquen en mercados direccionales, pues la direccionalidad significa que los inversores y los empresarios pueden planificar y actuar sobre patrones.

Es posible establecer una categorización por nivel de ingresos. La ONU ofrece tres niveles de este tipo: ingresos bajos, medios y altos; atendiendo al Ingreso Nacional Bruto (INB) per cápita. Como se señaló anteriormente, la tendencia principal es el aumento de la urbanización. Actualmente, más de la mitad de la población mundial vive en pueblos y ciudades, y la mayor parte del crecimiento futuro de la población se producirá en las zonas urbanas de los países en vías de desarrollo.

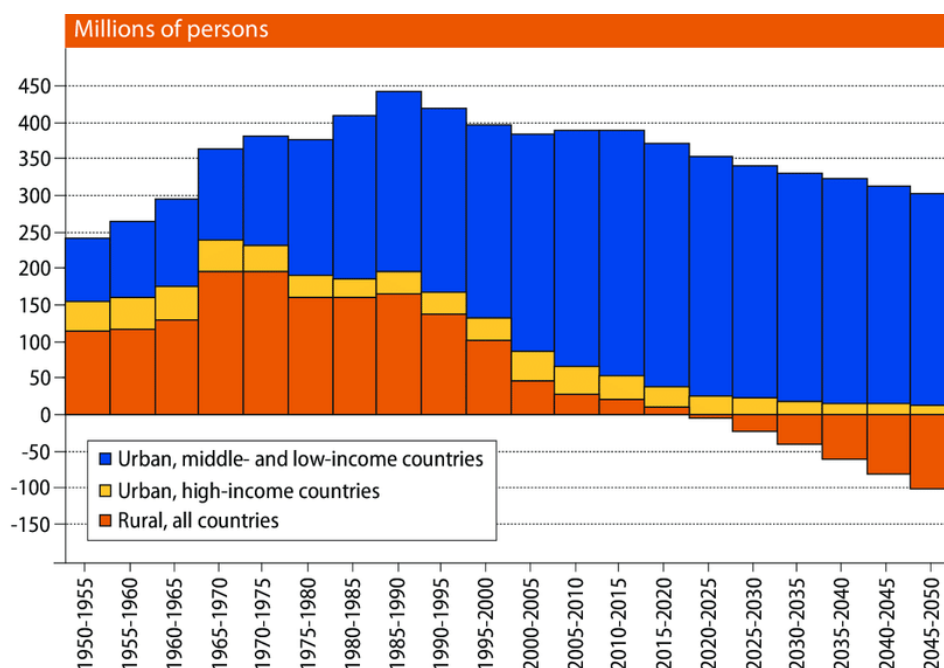
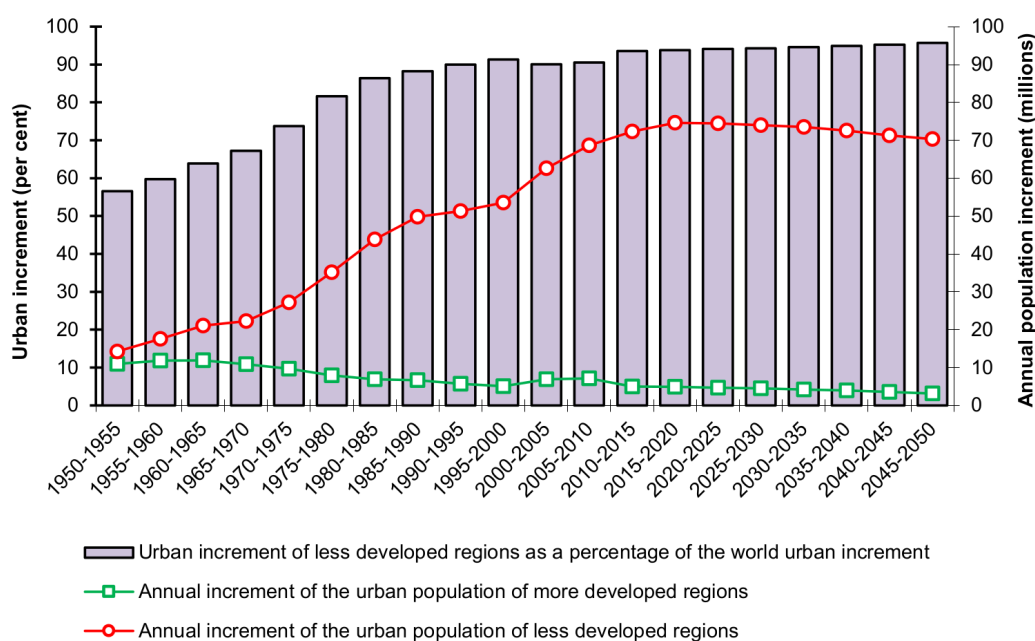


Figura 6: Crecimiento de la población urbana y rural, países de ingresos, altos, bajos y medios [18]

Este punto de vista permite un contraste entre las oportunidades presentes en los niveles de ingresos más bajos en relación con los mercados de apoyo actuales, que parecen centrarse en los mercados de ingresos más altos. La propensión de los servicios de micromovilidad a dirigirse solo a los ingresos altos (hasta la fecha) probablemente se deba a la mayor probabilidad de que los consumidores más ricos estén dispuestos a gastar en este nuevo servicio en comparación con los consumidores más pobres. Esta fue la misma percepción durante los primeros años de los productos de consumo durante el siglo pasado, pues: los electrodomésticos, las tecnologías de entretenimiento, los ordenadores personales y los teléfonos móviles siguieron este patrón.



**Figura 7: Incremento anual de la población urbana en regiones de mayor y menor desarrollo [19]**

Sin embargo, el crecimiento en los países de bajos ingresos eventualmente superó a los países de altos ingresos y las compañías que llegaron a dominar los electrodomésticos, la electrónica de consumo, las máquinas pòrtatiles y los teléfonos móviles están situadas en lo que solían ser países de ingresos medios. De igual modo, parece inevitable que la micromovilidad se expanda significativamente en los países de ingresos medios a medida que creen la mayor parte de la demanda de transporte urbano.

Para aclararlo más, basta con analizar los datos sobre urbanización, ciudad por ciudad. Los datos, también de la ONU, cubren 1860 ciudades o, más precisamente, aglomeraciones urbanas con 300,000 habitantes o más, estimadas para 2030. El mapa muestra el tamaño de población en proporciones relativas al diámetro, en base al cual podemos concluir que:

1. El crecimiento de la urbanización en India y África es asombroso, y debería ser el foco de muchas mentes en la micromovilidad.

2. El crecimiento en los países desarrollados es modesto e incluso negativo. Aunque existe la oportunidad de competir con la automovilidad en esos mercados, y tal vez de crear una nueva demanda, las oportunidades para abordar el nuevo consumo se encuentran en otra parte.
3. Las soluciones para movilizar millones de nuevos usuarios en nuevos mercados serán muy impactantes. Será mucho más fácil proporcionar transporte sostenible para los no consumidores que cambiarlos más tarde. Los mercados de rápida urbanización del mundo en desarrollo son más adecuados para una interrupción de gama baja, como la micromovilidad, que las ciudades más antiguas y ricas.

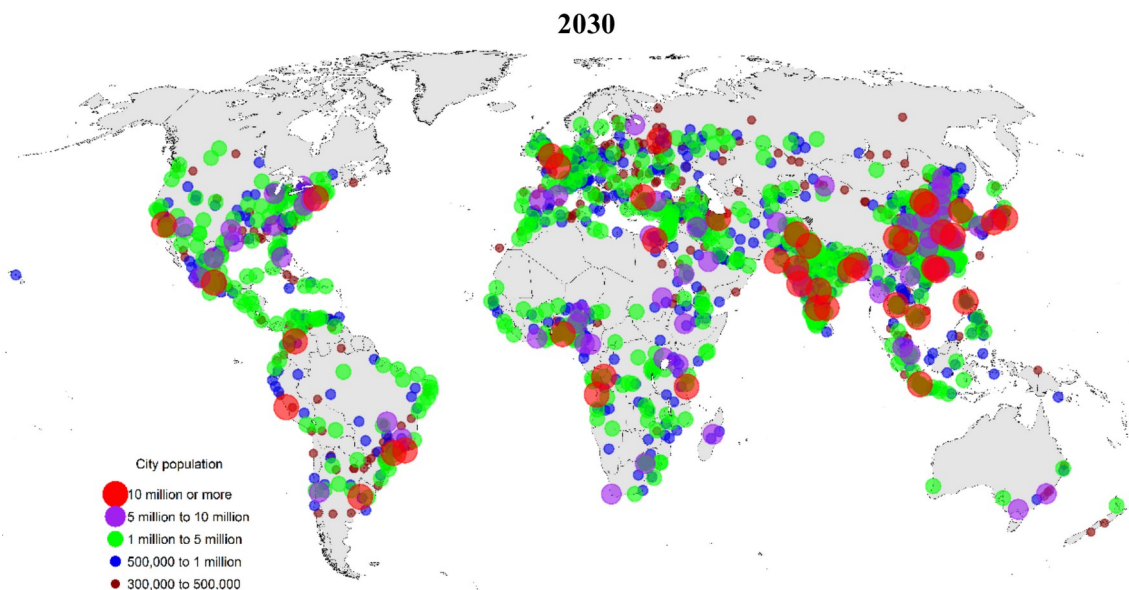


Figura 8: Ciudades por tamaño de asentamiento urbano, 2030 [18]

Luego, la micromovilidad no es una solución para las economías más ricas ni para las ciudades antiguas. Es todo lo contrario: su naturaleza de gama baja implica que es una mejor solución para los menos ricos y para las nuevas ciudades. La aplicación de la micromovilidad en países de bajos y medianos ingresos es inevitable.

## 2.4 Micromovilidad en la era pos-COVID

Con una pandemia que causa estragos en la vida diaria a nivel mundial, muchas cosas están cambiando, incluida la forma en que las personas se mueven. La micromovilidad ofrece muchos beneficios en este sentido, como es una forma personalizable de moverse mientras se mantiene la distancia social.

Si reflexionamos acerca de eventos pasados que reflejan un patrón similar: después del brote de SARS en China, las ventas de bicicletas eléctricas y el número de usuarios se disparó a medida que la gente buscaba alternativas a opciones de transporte público bastante concurridas. A lo que hay que añadir

que la movilidad urbana también tiene el potencial de apoyar la recuperación a largo plazo de la economía, especialmente entre las empresas locales.

Durante los primeros días de la pandemia, la ansiedad por el coronavirus impulsó el mayor aumento en las ventas de bicicletas desde la crisis del petróleo de 1970, ya que los viajeros buscaban alternativas seguras que permitiesen el distanciamiento social. NPD Group cifra el aumento en las ventas minoristas de bicicletas eléctricas en Estados Unidos en un 85% respecto al mes de marzo del año anterior [20].

A medida que la pandemia se prolonga, esa explosión inicial de entusiasmo se ha transformado en una buena racha de meses para la industria de las bicicletas eléctricas. El Covid está consolidando la curva de adopción y la población en general está cambiando rápidamente a lo que sin duda es la mejor forma de transporte urbano personal sostenible y saludable.

Esto es importante no solo en relación al COVID-19, sino también para la capacidad de recuperación general de una ciudad. Ya sea una pandemia global, tormentas severas, mala calidad del aire u otros efectos del cambio climático, las ciudades seguirán enfrentándose y tendrán que perseverar ante las interrupciones. Para mantener las ciudades en movimiento, necesitamos que todas las opciones estén disponibles: autobús, metro, taxis compartidos, caminar y, por supuesto, la creciente micromovilidad debe ser parte del plan de resiliencia de cada ciudad.

La ciudad de Nueva York está agregando más espacio para ciclistas y usuarios de micromovilidad, para apoyar el cambio repentino a pequeños modos de transporte individual en sus calles. Bogotá (Colombia) ha agregado 76 kilómetros de carriles bici prácticamente de la noche a la mañana para acomodar a más ciclistas y facilitar el distanciamiento social. Ciudades como Ciudad de México y Londres están viendo los beneficios de muchos años dedicados al crecimiento de sus redes de ciclismo, y se están moviendo para hacer que las medidas temporales de ciclismo sean permanentes. En Seattle, Milán, Montreal y París, los funcionarios estaban ocupados ampliando la infraestructura de ciclismo para alentar a los residentes a salir del encierro sobre dos ruedas en lugar de al volante de un automóvil. Respecto a España, Madrid ha planificado la construcción de nuevos carriles bici y ha peatonalizado 20 km; mientras en Barcelona el ayuntamiento ha iniciado la ampliación de 12 km de aceras y de la red de itinerancia ciclista, con 21 nuevos kilómetros.

Si algo ha dejado claro la crisis del coronavirus es que la micromovilidad en las grandes ciudades es clave en la desescalada, y que bicis, patinetes o motos eléctricas, compartidas o en propiedad, son una alternativa de transporte limpio y necesario en estos días de distancias obligadas.

A todo esto hay que añadir que el 91% de la población mundial vive en lugares donde la calidad del aire excede los límites de la salud [21], mientras estudios recientes muestran evidencia directa de que la presencia de partículas finas en el aire hace que el coronavirus sea más mortal [22].

Según José Luis Zimmermann, portavoz de Smart Mobility, *"son muchos los aspectos de nuestras vidas que esta emergencia global va a cambiar, y uno de ellos es cómo nos desplazamos dentro de las ciudades. La micromovilidad es una alternativa para evitar la congestión del transporte público, que podría elevar el riesgo de contagio, o el uso masivo del coche privado, que aumenta la presencia de partículas contaminantes en el aire y facilita la propagación del virus"* [23].

Los analistas del Instituto Coordinadas [24] señalan dos elementos que impulsarán el crecimiento de la micromovilidad. Por un lado, la mayoría de los lanzamientos se hacen en entornos propicios, ya que vienen a solucionar problemas acuciantes para los que de momento no hay alternativa. Esto facilita una clara aceptación ciudadana hacia el modelo, por lo que las novedades y nuevas iniciativas "tienen amplias posibilidades de posicionarse". Por otro lado, la economía de la micromovilidad proporciona "buenas oportunidades de negocio a las empresas", como demuestra el hecho de que la inversión para crear una flota de motos eléctricas en una ciudad media tiene un coste de menos de 400 euros por vehículo.

## 3. Modelo CAD

### 3.1 Diseño 3D

Como punto de partida para la representación geométrica se ha tomado como referencia un monociclo KingSong modelo KS-18XL, propiedad del autor, del cual se han obtenido las dimensiones de los principales elementos que lo conforman.

Más concretamente, nos estamos refiriendo a uno de los monociclos más exclusivos y versátiles de la marca KINGSONG conocido como KS-18XL, descrito por las siguientes especificaciones técnicas:

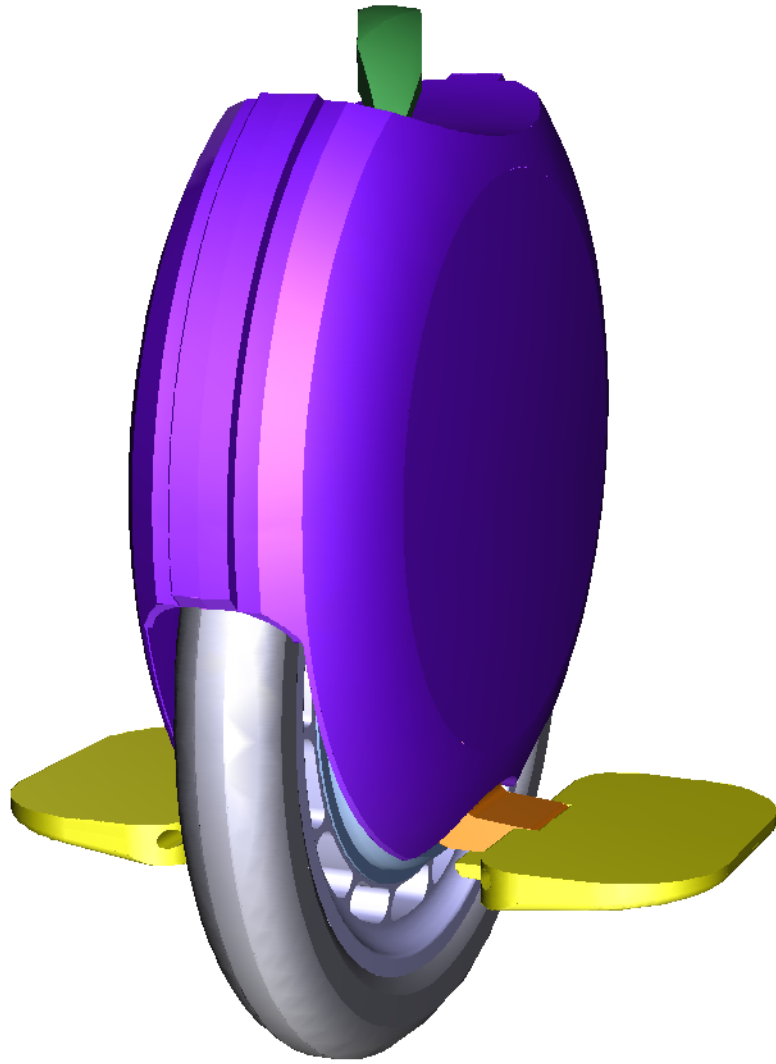
- Diámetro del neumático: 18"
- Peso: 24kg
- Dimensiones: 495 mm x 590 mm x 180 mm
- Carga máxima: 120kg
- Pendiente máxima:  $\approx 35^\circ$
- Autonomía (máxima): 140km
- Batería: 1554Wh, 84V 1.5A Lithium-ion
- Tiempo de carga batería: 7-14h
- Velocidad máxima: 50km/h
- Potencia nominal: 2000W
- Máxima potencia: 4000W
- Tipo de motor: brushless (BLDC)
- Mínima temperatura de trabajo:  $-10^\circ\text{C}$
- Máxima temperatura de trabajo:  $60^\circ\text{C}$

Para la definición e implementación tridimensional de cada una de las piezas integrantes de nuestro conjunto se ha hecho uso de la herramienta de diseño asistido por ordenador CATIA, software del sector que cuenta con potentes herramientas de modelado paramétrico y características avanzadas para superficies.

Entre las operaciones de diseño utilizadas para la representación CAD del producto destacan las enumeradas a continuación:

- Pad: extrusión de un perfil o superficie en una o dos direcciones
- Pocket: extruir un perfil o una superficie y retirar el material resultante de la extrusión
- Shaft: barrido de rotación de un perfil abierto o cerrado. La rotación se basa en un ángulo de barrido deseado alrededor de un eje
- Groove: operación de revolución que elimina material de las operaciones existentes
- Stiffener: crea un nervio rigidizador a partir de la dirección de creación





- **Mirror:** refleja la entidad seleccionada, esto es, la duplica simétricamente con respecto a un plano
- **Multi-sections:** generación de un volumen cerrado barriendo una o más curvas de sección planas lo largo de una o más curvas guía
- **Assemble:** operación booleana que combina dos cuerpos en uno solo
- **Remove:** elimina la interferencia de un cuerpo respecto a otro cuerpo mediante una operación booleana
- **Swept:** superficie generada barriendo un perfil en planos normales a una curva mientras se tiene en cuenta otros parámetros como curvas guía u otros elementos de referencia
- **EdgeFillet:** curva de radio constante o variable que es tangente y que une a dos superficies
- **CircPattern:** para creación de patrones circulares, es decir, duplicar un elemento original en la ubicación y número elegido de acuerdo con una disposición circular
- **Extract:** extracción de superficies para la descomposición de una superficie o lista de superficies que encierra un volumen sólido en varios subelementos sin eliminar el elemento de partida

Como resultado un total de 13 componentes distintos son los integrantes de nuestro ensamblaje general:

1. Asa (handle)
2. Carcasa (housing)
3. Pedales (pedals)
4. Soportes pedales (pedal supports)
5. Llanta (rim)
6. Neumático (tire)
7. Cubremotor (cap motor)
8. Rodamientos (bearings)
9. Eje (shaft)
10. Estátor (stator)
11. Nucleo bobinado (stator\_coil)
12. Bobina (coil)
13. Imanes permanentes (magnets)

De todos los componentes que integran el conjunto rueda monociclo, no todos serán de consideración en nuestro estudio multifísico. Básicamente, nos centraremos en el comportamiento del motor eléctrico, por lo que el número de piezas diferentes que manejar se reducirá de 13 a 7 (cubremotor, rodamientos, eje, estátor, núcleo del bobinado, bobina e imanes permanentes).

Por otro lado, sólo resta decir que nuestro dominio fluido será el aire confinado en el motor, el cual no requiere ser representado geoméricamente para su discretización en volúmenes finitos, pues, el algoritmo de mallado identificará el volumen que encierra el motor de acuerdo a las superficies que lo envuelve o limitan.

### 3.2 Simplificación geométrica

Con el objetivo de obtener una malla de calidad aceptable y reducir el tiempo de cálculo, 4 simplificaciones o idealizaciones han sido implementadas:

1. Eliminación del redondeo interior y exterior de las aletas del cubremotor con objeto de incurrir en un menor número de celdillas para el volumen fluido colindante, así como, en una mayor calidad de los elementos.

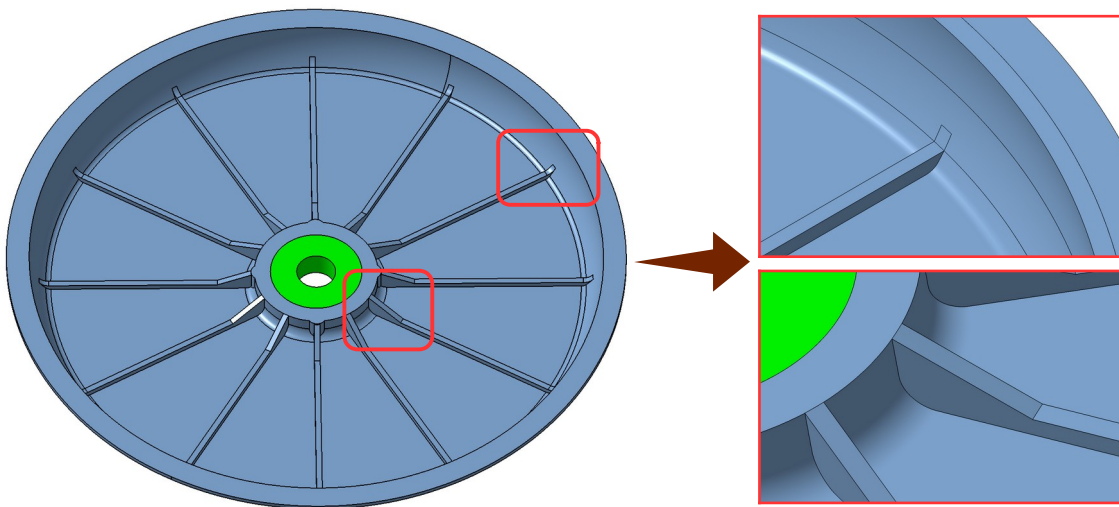


Figura 10. Simplificación geométrica n.º 1

2. Simplificación del devanado y su núcleo a una configuración de diseño equivalente, pero de mayor regularidad y simplicidad geométrica.

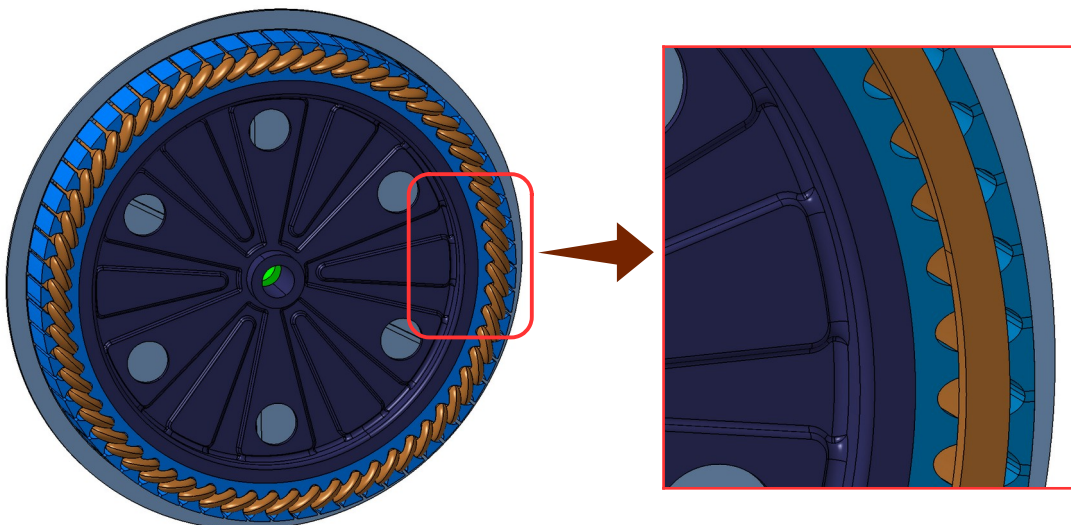


Figura 11. Simplificación geométrica n.º 2

3. Eliminación del insignificante gap ( $\sim 0.5$  mm) que separa el núcleo de la bobina del estátor de los imanes permanentes del rotor, y que debido a efectos de la capa límite se puede considerar despreciable el paso de aire a su través.

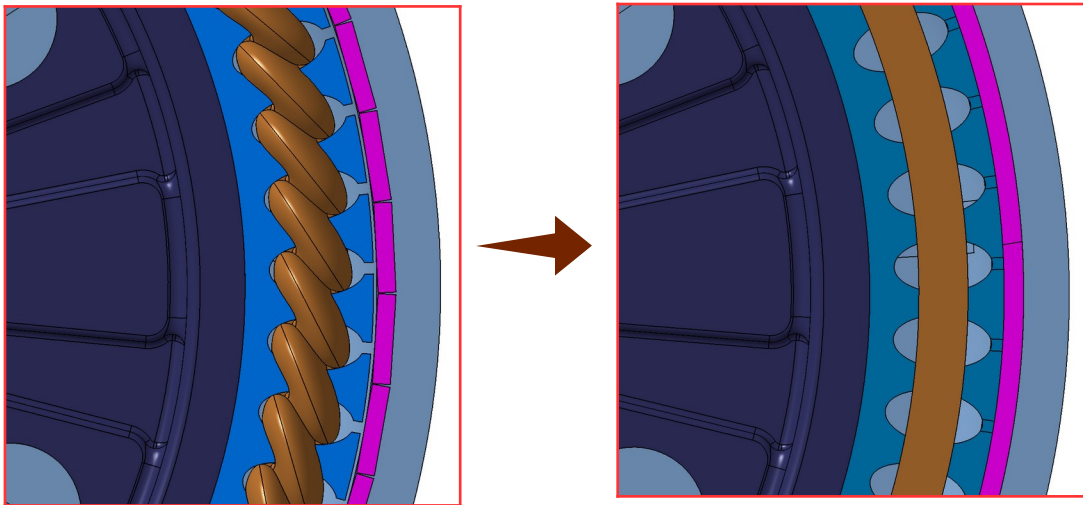


Figura 12. Simplificación geométrica n.º 3

4. Supresión de las lengüetas o chavetas mecanizadas en el eje, las cuales carecen de interés en nuestro estudio.

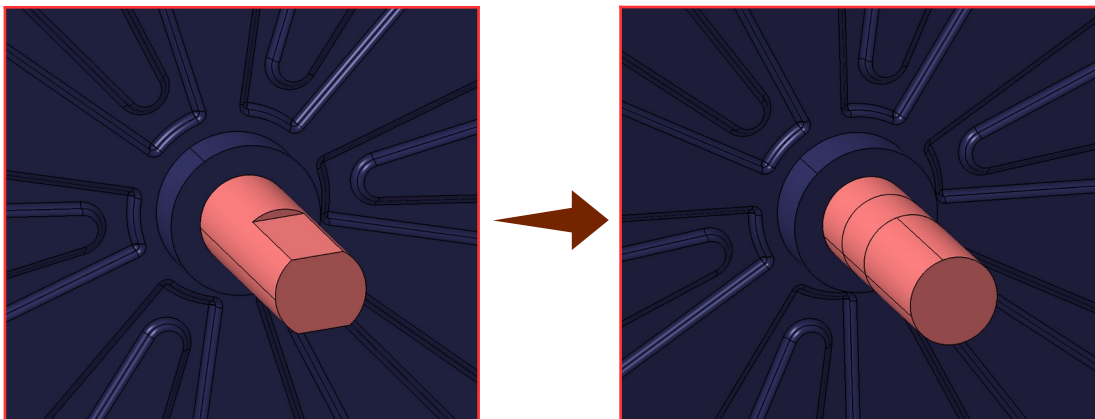


Figura 13. Simplificación geométrica n.º 4

Por último, la geometría ha sido dividida en diversas regiones, acorde a: material, componente o condición de contorno que será aplicada durante la configuración del caso de estudio. Regiones que podrán ser exportadas desde CATIA como superficie triangulada 2D, para su subsiguiente discretización 3D y simulación en OpenFOAM.

## 4. Recopilación y tratamiento de datos

### 4.1 Introducción

En orden a obtener la información necesaria a nuestra investigación y reproducir el comportamiento real de nuestro sistema, se ha llevado a cabo un test en campo abierto del monociclo eléctrico. El instrumento usado para tal fin es la aplicación (app) WheelLog [25], aplicación que permite la monitorización y telemetría de datos medidos por los sensores durante el trayecto, así como, el registro de los mismos para su posterior tratamiento cuantitativo.

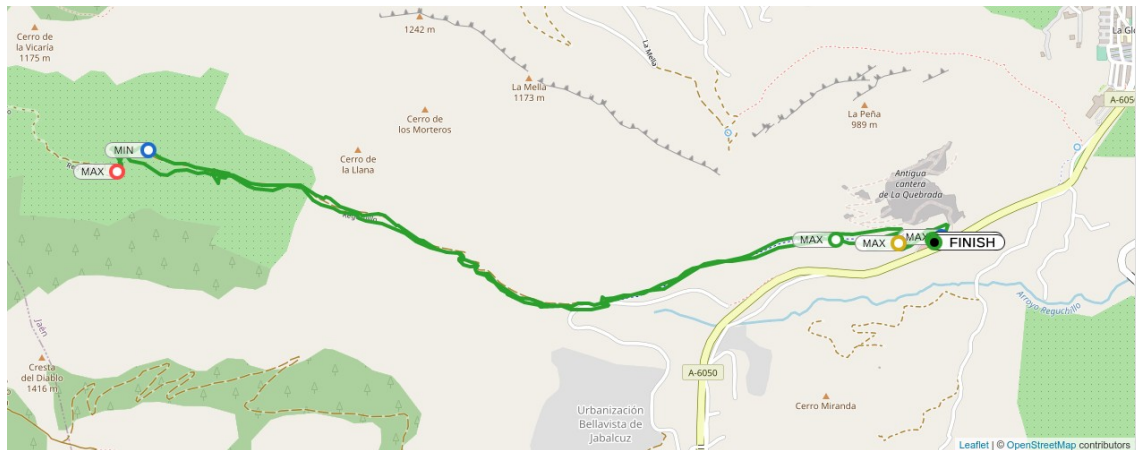


Figura 14: Ruta GPS recorrida durante el testeo [26]

Esta app puede actuar como una “caja negra” y ofrece la posibilidad de controlar la velocidad, temperatura, tensión, intensidad, potencia, energía de carga y descarga, nivel de batería, distancia recorrida y otros parámetros vitales para un apropiado funcionamiento de la rueda. Además, admite la configuración de alarmas (activar la bocina, encender y apagar las luces, etc) para advertir sobre la proximidad a alcanzar algún límite prefijado, como por ejemplo, una temperatura máxima.

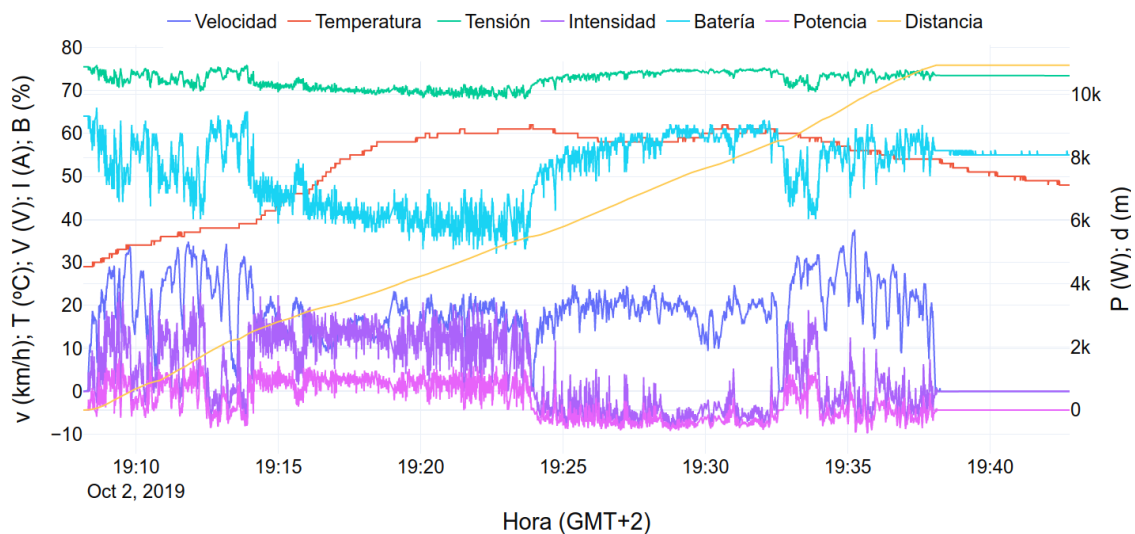


Figura 15. Representación de datos grabados durante el testeo [27]

De la multitud de variables capaz de coleccionar la app nos centraremos en aquellas de mayor relevancia para nuestro estudio, como son la velocidad y la temperatura.

## 4.2 Velocidad angular

Conocida la velocidad lineal  $v$ , basta con determinar el radio de la rueda  $R$  ( $R=0.235$  m), para poder vincular esta con la velocidad angular  $\omega$  mediante la siguiente expresión:

$$\omega = \frac{v}{R}$$

Transformación que nos permitirá calcular la velocidad de rotación en función del tiempo (fecha y hora) para el conjunto de datos de testeo del que disponemos. Con ello se hace factible un análisis estadístico de distribución de frecuencias como procedimiento descriptivo de la variable velocidad angular. Básicamente, se pretende obtener una herramienta que evidencie las velocidades de rotación que son preferencia o tendencia.

Dado el carácter de variable unidimensional y continua de la velocidad de giro, la frecuencia relativa puede ser evaluada de forma sencilla. En este supuesto, los datos pueden disponerse agrupándolos o clasificándolos en intervalos (clases) mutuamente excluyentes, e indicando el número de datos que caen dentro de cada intervalo.

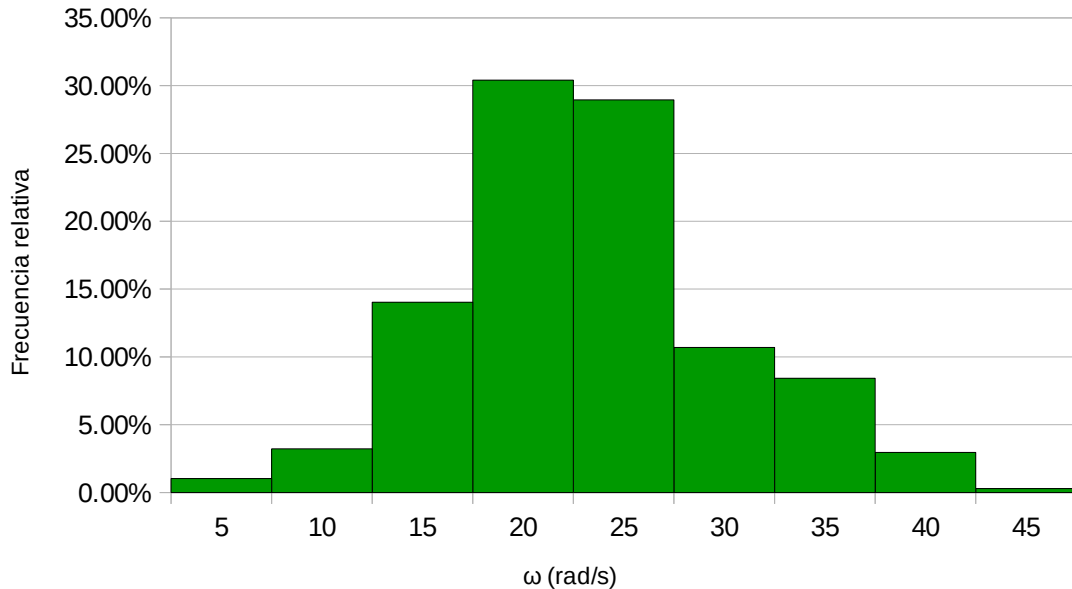
Considerando una amplitud de intervalo de 5 rad/s y un recorrido de 45 rad/s (iniciación a 2.5 rad/s y finalización en 47.5 rad/s) tendremos un total de 9 categorías. El motivo por el que se elige comenzar a un cierto offset de 0 es debido a que se prefiere obviar velocidades nulas o quasi-estáticas, pues carecen de interés por la irrelevancia que suscitan velocidades tan bajas en el patrón de comportamiento fluido.

La tabla de frecuencias, teniendo presente que el límite inferior de una clase pertenece al intervalo, pero el límite superior no pertenece al intervalo, es para la velocidad angular:

Clase	Marca de clase	Frecuencia absoluta	Frecuencia relativa
[2.5,7.5)	5	90	1.02%
[7.5,12.5)	10	283	3.22%
[12.5,17.5)	15	1232	14.03%
[17.5,22.5)	20	2671	30.41%
[22.5,27.5)	25	2543	28.95%
[27.5,32.5)	30	939	10.69%
[32.5,37.5)	35	740	8.43%
[37.5,42.5)	40	260	2.96%
[42.5,47.5)	45	25	0.28%

Tabla 2: Tabla de distribución de frecuencias

Un gráfico asociado a la tabla de frecuencias de variables cuantitativas continuas son los histogramas. Un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.



*Figura 16: Histograma frecuencia relativa de la velocidad de rotación*

Extrapolando estos resultados a un uso genérico del monociclo quedan patentes los espectros de velocidad angular más característicos y con una mayor probabilidad de reproducirse comúnmente. Esto nos ayudará a tratar de forma estacionaria el estudio CFD del comportamiento del flujo en el interior del motor, imponiendo para ello, velocidades de giro discretas en cada simulación. Más concretamente, nos concentraremos en los intervalos de mayor frecuencia (>10%), fijando como velocidad angular la correspondiente a su marca de clase (punto medio del intervalo) en cada caso.

Conocedores de los regímenes de giro comprendidos dentro de los rangos de uso más destacados, analizaremos 4 velocidades de giro distintas acotadas entre los 15 rad/s y 30 rad/s con un incremento escalonado de 5 rad/s ([15:5:30]). Con el propósito de determinar el fenómeno de transporte que rige el movimiento del fluido para cada velocidad de rotación, trabajaremos con el valor adimensional conocido como número de Reynolds (Re).

El Re se define como la relación entre las fuerzas inerciales y las fuerzas viscosas dentro de un fluido que está sujeto a un movimiento interno relativo debido a diferentes velocidades del fluido. Movimiento relativo que genera una fricción en el fluido y que es un factor de desarrollo de un flujo turbulento. Por otro lado, contrarrestando este efecto está la viscosidad del fluido, que tiende a inhibir la turbulencia. La expresión del número de Reynolds [28] adaptada a nuestro caso rotativo viene dada por:

$$Re = \frac{uL}{\nu} = \frac{\omega r_m w_{max}}{\nu}$$



donde  $\nu$  es la viscosidad cinemática del aire a temperatura ambiente ( $1.5 \cdot 10^{-5} \text{ m}^2/\text{s}$ ),  $r_m$  es el radio medio del dominio fluido (122 mm) y  $w_{\min}$  es la longitud característica correspondiente a la mínima distancia entre el rotor y el estátor (15 mm). Dimensión, esta última, que representa la sección de paso más reducida (exceptuando los huecos existentes entre la bobina y su propio núcleo, donde la velocidad del fluido se puede considerar depreciable), es decir, el sector donde el desarrollo del frente de flujo será más importante, pudiéndose alcanzar primeramente el régimen de turbulencia.

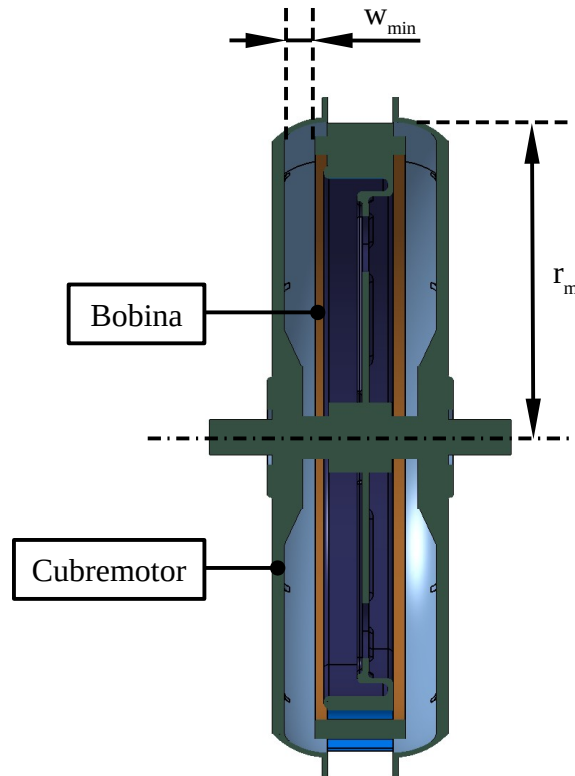


Figura 17. Sección en el plano XY para el conjunto motor

Los modelos de comportamiento fluido quedan así divididos en: a) flujo laminar, ocurre a bajos  $Re$  ( $<2300$ ), donde las fuerzas viscosas prevalecen, y se caracteriza por un movimiento fluido constante y suave; b) flujo turbulento, dado para altos  $Re$  ( $>4000$ ), está dominado por fuerzas inerciales, que tienden a producir remolinos caóticos, vórtices y otras inestabilidades de flujo; c) flujo transicional, si el número de Reynolds se encuentra entre valores superiores a 2300 e inferiores a 4000.

Velocidad angular (rad/s)	Número de Reynolds	Comportamiento
15	1830	Laminar
20	2440	Transitorio (bajo)
25	3050	Transitorio (medio)
30	3660	Transitorio (alto)

Tabla 3: Número de Reynolds asociado a la velocidad angular del rotor



Como consecuencia de la magnitud de los valores del Re de la tabla, al menos en términos de la dinámica de fluidos, analíticamente el comportamiento del flujo asociado a la primera velocidad de rotación deberá ser asimilado a un caso de flujo laminar, mientras que para las demás velocidades se requerirá de un modelo de transporte turbulento. Lo que se tratará de desarrollar en mayor profundidad en el apartado 5.

### 4.3 Temperatura

En relación a la otra variable influyente en el análisis de nuestro sistema, la temperatura, reduciremos a un único valor característico y representativo su evolución a lo largo de todo el itinerario seguido durante la fase de prueba. Para ello, utilizaremos el valor máximo absoluto de temperatura como situación térmica más desfavorable, y por tanto, de mayor interés investigativo.

Una vez contabilizada, del total de datos medidos, la máxima magnitud alcanzada por la variable temperatura del sistema ( $T_{\max}=62\text{ °C}$ ), estimaremos su velocidad angular media asociada, velocidad que vendrá calculada como el promediado de aquel conjunto finito de datos cuya temperatura instantánea coincide con el valor pico  $T_{\max}$ .

Llegados a este punto, y teniendo en cuenta que la viscosidad del aire, aunque sutilmente, aumenta conforme lo hace la temperatura, podemos afirmar que el mayor número de Re para el fluido, por ser inversamente proporcional, vendrá determinado por la viscosidad cinemática más pequeña, o lo que es lo mismo, por la viscosidad correspondiente a la temperatura ambiente.

Por tanto, conocida la velocidad de rotación media ( $\omega_m=16\text{ rad/s}$ ), y que la viscosidad cinemática más apropiada analíticamente sigue siendo la usada hasta el momento, se hace evidente el cálculo del Re para esta  $\omega_m$ . Dado que resulta que  $Re=1950$ , estamos en posición de afirmar que el comportamiento del flujo a alta temperatura será el descrito como régimen laminar.

Sabiendo que la medición térmica en el sistema se lleva a cabo sobre el punto más cálido, esto es, en las inmediaciones al devanado, nos encontramos ya en disposición de realizar un estudio multifísico del motor fundamentado en la imposición de esta temperatura máxima ( $T_{\max}=62\text{ °C}$ ) sobre la bobina y su correspondiente velocidad ( $\omega_m=16\text{ rad/s}$ ) sobre el fluido en rotación. Lo cual será el objetivo del apartado 6 de este trabajo.

## 5. Caso MRFSimpleFoam

### 5.1 Introducción

Hay una amplia variedad de problemas en el campo de la ingeniería donde la rotación de alguno de los componentes y sus efectos están presentes. Las turbomáquinas (turbinas, compresores, bombas, ventiladores), conductos de refrigeración, así como, las ruedas eléctricas y los motores de buje entre otros, son un claro ejemplo donde este fenómeno debería ser abordado.

Existen diferentes vertientes teóricas a la hora de tratar los efectos de la rotación en los campos de flujo. Una posibilidad es introducir directamente las fuerzas generadas debido a la rotación, utilizando el término fuente  $\mathbf{f}$  de la ecuación del momento. Sin embargo, esto requiere que el usuario conozca las magnitudes y distribuciones de las fuerzas a priori, lo cual generalmente no es factible. Por lo tanto, esta técnica generalmente es solo aproximada.

Otro enfoque consiste en resolver un problema transitorio de malla en movimiento. Esto tiene la ventaja de que es exacto, al menos en el nivel de aproximación impuesto por el tamaño de elemento, el paso de tiempo, los esquemas numéricos, etc, es decir, no hay nada inherentemente aproximado en el tratamiento de los efectos rotacionales: las paredes se mueven, generan gradientes de presión, que inducen al fluido a moverse. Sin embargo, el enfoque de malla en movimiento transitorio tiene importantes inconvenientes. En primer lugar, es más complejo y requiere que el usuario defina eventos (unión/separación de celdillas), y use o cree un movimiento de malla consistente con estos eventos. En segundo lugar, requiere una solución dependiente del tiempo, incluso si al usuario solo le interesa el comportamiento promediado en el tiempo. Las soluciones transitorias son computacionalmente más costosas ya que se debe de obtener una solución convergente en cada paso de tiempo.

Un tercer enfoque es el uso de múltiples marcos de referencia (MRF). Esta técnica puede considerarse como un compromiso que combina las mejores características de los dos enfoques anteriores: mejor precisión que el método de término fuente, con menos coste computacional y de configuración que la técnica de malla móvil. Aunque es algo aproximada en muchos casos, nos permite, al menos en principio, resolver cualquier problema de flujo de fluidos en cualquier marco de referencia que elijamos, siempre y cuando definamos nuestras condiciones de contorno de manera consistente. El punto importante es que debemos optar por resolver el problema en el marco de referencia que tenga más sentido.

Una de las grandes ventajas de este último enfoque es que podemos resolver el problema estacionario. Esto implica un coste computacional considerablemente menor y también facilita el pos procesamiento. Pero, ¿qué sucede en la interfaz entre las partes estacionarias y giratorias del modelo? Las velocidades en el lado giratorio de la interfaz deben transformarse en un marco de referencia estacionario y viceversa. En OpenFoam de todo esto se encarga el solver simpleFoam.

## 5.2 Algoritmo simpleFoam

El algoritmo SIMPLE (Semi-Implicit Method for Pressure Linked Equations) es un procedimiento numérico frecuentemente utilizado para resolver las ecuaciones de Navier-Stokes. Este enuncia que, si un problema de estado estacionario se resuelve de manera iterativa, no se hace necesario determinar completamente el acoplamiento lineal presión-velocidad. Cada ciclo de iteración implica, en primer lugar, el cálculo de un campo de velocidad intermedio que satisface las ecuaciones de momento linealizado para una distribución de presión estimada: luego se invoca el principio de conservación de la masa para ajustar las velocidades y presiones, de modo que todas las ecuaciones queden en equilibrio [29].

Las ecuaciones de Navier-Stokes [30] que gobiernan un flujo monofásico en estado estacionario con densidad y viscosidad constantes quedan reducidas a:

$$\text{Ecuación de continuidad: } \nabla \cdot \mathbf{u} = 0$$

$$\text{Ecuación del momento: } \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) = -\nabla p_k + \mathbf{g} + \mathbf{f}$$

donde  $\mathbf{u}$  es el vector velocidad,  $p_k$  la presión cinemática,  $\mathbf{g}$  el vector fuerza de la gravedad y  $\mathbf{f}$  el término fuerzas externas.

El algoritmo SIMPLE resuelve una serie cíclica de operaciones de conjetura y corrección, en las que la velocidad se calcula en primer lugar mediante la ecuación del momento para un campo de presión supuesto, y luego este último, y a continuación las velocidades, se ajustan para satisfacer la continuidad. Estos pasos básicos pueden ser resumidos como [31]:

1. Se obtiene una aproximación del campo de velocidades resolviendo la ecuación del momento. El término gradiente de presión se calcula utilizando la distribución de presión de la iteración anterior o una estimación inicial.
2. Se formula y se resuelve la ecuación de presión para obtener la nueva distribución de presión.
3. Se corrigen las velocidades y se calcula un nuevo conjunto de flujos conservativos.

La ecuación de momento discretizado y la ecuación de corrección de presión se resuelven implícitamente, mientras que la corrección de velocidad se resuelve explícitamente. Esta es la razón por la que se le llama "Método Semi-implícito".

En OpenFoam el archivo simpleFoam.C (an. 11.1) contiene el bucle para resolver la ecuación de momento Ueqn.H (an. 11.2) y presión pEqn.H (an. 11.3). En concreto, pEqn.H comprende la resolución y relajación de la presión, así como la corrección de la velocidad.

Por otro lado, el archivo createFields.H (an. 11.4) contiene los campos ( $\rho$  y  $\mathbf{U}$ ) que serán resueltos para el campo fluido.

### 5.3 Modelo de turbulencia k- $\epsilon$

En dinámica de fluidos, la turbulencia es un régimen de flujo que goza de cierta irregularidad, aleatoriedad y que se caracteriza por cambios caóticos e incluye baja difusión de momento, alta convección de momento y rápida variación en el espacio y el tiempo. Hay varios modelos para tener en cuenta la turbulencia como: a) Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) y c) Reynolds-Average Simulation (RAS).

RAS es un enfoque de modelado para predecir flujos turbulentos promediando en tiempo las ecuaciones de Navier-Stokes. Este es bastante robusto y el más comúnmente utilizado en aplicaciones industriales, así como, por su coste computacional será el idóneo a implementar en nuestro análisis de tipo rotativo. En particular, nos decantaremos por su subgrupo, el modelo k- $\epsilon$ .

Este último es un modelo estándar de turbulencia [32] para su uso con flujos incompresibles y basado en la teoría de distorsión rápida (RDT) [33] para su uso con flujos compresibles. Consta de las dos siguientes ecuaciones de transporte:

- La ecuación de energía cinética turbulenta,  $k$

$$\frac{D}{Dt}(\rho k) = \nabla \cdot (\rho D_k \nabla k) + P_k - \rho \epsilon$$

siendo  $\rho$  la densidad,  $D_k$  la difusividad efectiva para  $k$  y  $P_k$  la tasa de producción de energía cinética turbulenta.

- La ecuación de la tasa de disipación de energía cinética turbulenta,  $\epsilon$

$$\frac{D}{Dt}(\rho \epsilon) = \nabla \cdot (\rho D_\epsilon \nabla \epsilon) + \frac{C_1 \epsilon}{k} \left( P + C_3 \frac{2}{3} k \nabla \cdot \mathbf{u} \right) - C_2 \rho \frac{\epsilon^2}{k}$$

donde  $D_\epsilon$  es la difusividad efectiva para  $\epsilon$  y  $C_1$ ,  $C_2$ ,  $C_3$  son constantes del modelo.

Finalmente, la viscosidad turbulenta  $\nu_t$ , viene dada como la relación existente entre las dos ecuaciones anteriores:

$$\nu_t = C_\mu \frac{k^2}{\epsilon}$$

con  $C_\mu$  coeficiente del modelo para la viscosidad turbulenta.

Si el flujo turbulento interacciona con contornos sólidos, es necesario especificar el tratamiento cerca de la pared. Para ello se cuenta con la opción de resolver la subcapa viscosa o emplear funciones de pared.

## 5.4 Marco de Referencia Múltiple (MRF)

En la implementación MRF, el dominio computacional se divide en dos subdominios que se pueden asumir como en rotación o traslación en referencia al marco inicial. Las ecuaciones que rigen cada dominio se obtienen del marco de referencia correspondiente al subdominio.

Consideremos un sistema de coordenadas que se traslada con una velocidad lineal  $\mathbf{u}_T$  y gira con una velocidad angular  $\omega$  relativa a un sistema de referencia estacionario (inercial). El origen del sistema en movimiento se ubica mediante un vector de posición  $\mathbf{r}_0$ .

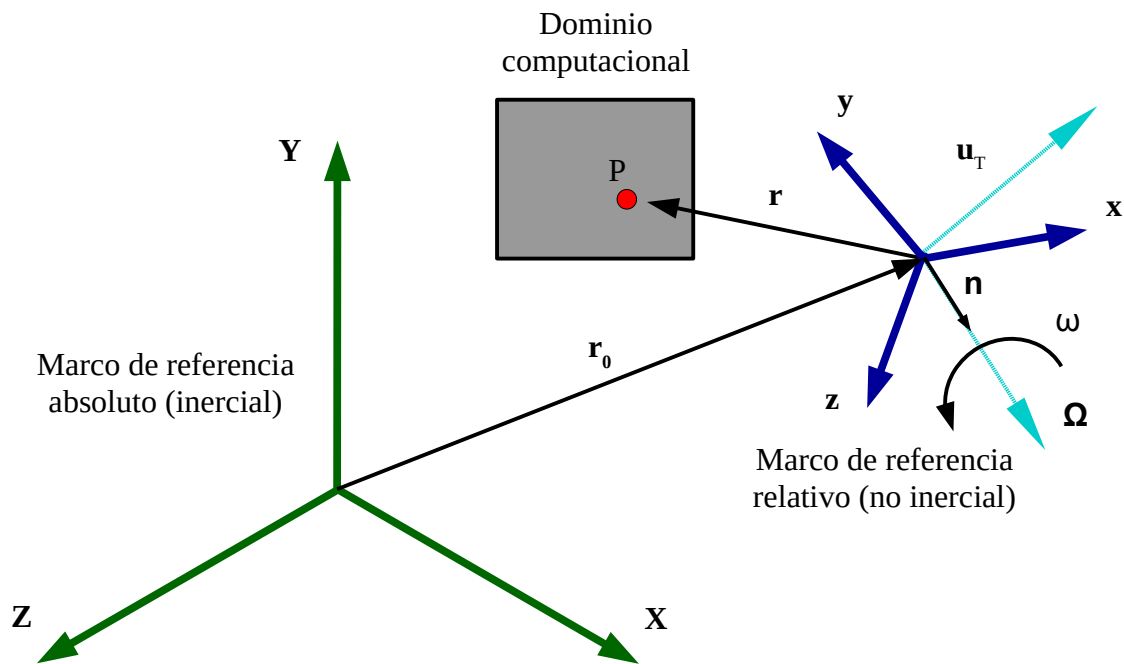


Figura 18. Marcos de referencia estacionario (absoluto) y móvil (relativo)

El dominio computacional se define con respecto al marco móvil de tal manera que un punto arbitrario  $P$  en el dominio está localizado por un vector de posición  $\mathbf{r}$  desde el origen del marco en movimiento. Las velocidades del fluido se pueden transformar desde el marco en movimiento (no inercial) al marco estacionario (inercial) utilizando la siguiente relación:

$$\mathbf{u}_I = \mathbf{u}_R + (\boldsymbol{\Omega} \times \mathbf{r}) + \mathbf{u}_T$$

donde  $\mathbf{u}_I$  es la velocidad en el marco de referencia absoluto inercial,  $\boldsymbol{\Omega}$  el eje de rotación ( $\boldsymbol{\Omega} = \omega \mathbf{n}$ , siendo  $\mathbf{n}$  el vector de dirección unitario),  $\mathbf{u}_R$  la velocidad relativa (velocidad vista desde el marco en movimiento) y  $\mathbf{u}_T$  la velocidad de traslación del marco. Expresión que, acondicionada a nuestro caso, en el que solo contamos con rotación del marco en movimiento, se reduce a:

$$\mathbf{u}_I = \mathbf{u}_R + (\boldsymbol{\Omega} \times \mathbf{r})$$

donde si no existiese rotación, es decir,  $\boldsymbol{\Omega} = \mathbf{0}$ , se cumpliría que  $\mathbf{u}_I = \mathbf{u}_R$ .

Retomando las ecuaciones de conservación presentadas anteriormente y despreciando las fuerzas gravitacionales y de origen externo, las ecuaciones de Navier-Stokes en un marco de referencia inercial pueden ser descritas por:

$$\text{Ecuación de continuidad en marco inercial: } \nabla \cdot \mathbf{u}_I = 0$$

$$\text{Ecuación del momento en marco inercial: } \nabla \cdot (\mathbf{u}_I \otimes \mathbf{u}_I) = -\nabla p_k + \nu \nabla \cdot \nabla \mathbf{u}_I$$

Ecuaciones que escritas en términos de la velocidad absoluta en un marco de referencia en rotación quedan como:

$$\text{Ecuación de continuidad en marco rotativo: } \nabla \cdot \mathbf{u}_I = 0$$

$$\text{Ecuación del momento en marco rotativo: } \nabla \cdot (\mathbf{u}_R \otimes \mathbf{u}_I) + \boldsymbol{\Omega} \otimes \mathbf{u}_I = -\nabla p_k + \nu \nabla \cdot \nabla \mathbf{u}_I$$

donde el término  $\boldsymbol{\Omega} \times \mathbf{u}_I$  hace referencia a la fuerza de Coriolis.

En OpenFOAM, como se puede deducir de las dos últimas líneas de código del archivo de cabecera createFields.H (an. 11.4), los efectos de rotación son ya considerados mediante la llamada a createMRF.H (an. 11.5). El cual crea el objeto MRF invocando la clase IOMRFZoneList, y que contendrá los datos del modelo relativo al dominio fluido en rotación. Adicionalmente, el cálculo de la fuerza de Coriolis se lleva a cabo a través del archivo MRFZone.C (an. 11.6).

## 5.5 Mallado modelo MRFSimpleFoam

En orden a analizar el flujo en fluidos, los dominios computacionales se dividen en subdominios y las ecuaciones discretizadas de Navier-Stokes son resueltas en cada celdilla o elemento. El proceso para abordar la discretización del dominio se llama generación de malla y se remonta a los primeros días de CFD cuando la mayoría de los análisis en la práctica eran 2D. La generación de mallas es un paso importante para trabajar con CFD y emplea diferentes algoritmos para proporcionar análisis cualitativos de los campos fluidos.

Generalmente, las celdillas se pueden dividir en dos categorías: celdillas estructuradas y no estructuradas. Las estructuradas se identifican por una conectividad regular. Son adecuadas para estudios de capa límite y consisten en un cuadrilátero para dominios 2D o hexaedro para 3D. Por otro lado, las celdillas no estructuradas muestran una conectividad irregular y los elementos se distribuyen arbitrariamente a través del dominio computacional. Para un dominio 2D, el tipo de cuadrícula son elementos cuadriláteros o triangulares, mientras que para uno 3D los elementos son prismas, tetraedros, hexaedros o poliedros.

Existen varias utilidades para la generación, manipulación y conversión de malla disponibles en OpenFOAM, destacando en el campo de generación de malla: blockMesh para la creación de mallas hexaédricas estructuradas muy adecuada para geometrías que puedan describirse con unos pocos bloques; y snappyHexMesh para la generación de malla hexagonal compleja de celdas

divididas automáticamente a partir de geometrías de superficie trianguladas, garantizando una calidad de malla mínima.

La utilidad blockMesh crea mallas paramétricas con la división de bordes haciendo uso de un diccionario llamado blockMeshDict ubicado en el directorio system del caso. BlockMesh lee este diccionario, genera la malla y escribe los datos de la malla en el directorio constant.

El principio detrás de la utilidad blockMesh es descomponer la geometría del dominio en un conjunto de 1 o más bloques hexaédricos. Los bordes de los bloques pueden ser líneas rectas, arcos o splines. La discretización de malla se manifiesta como un número de celdas en cada dirección del bloque.

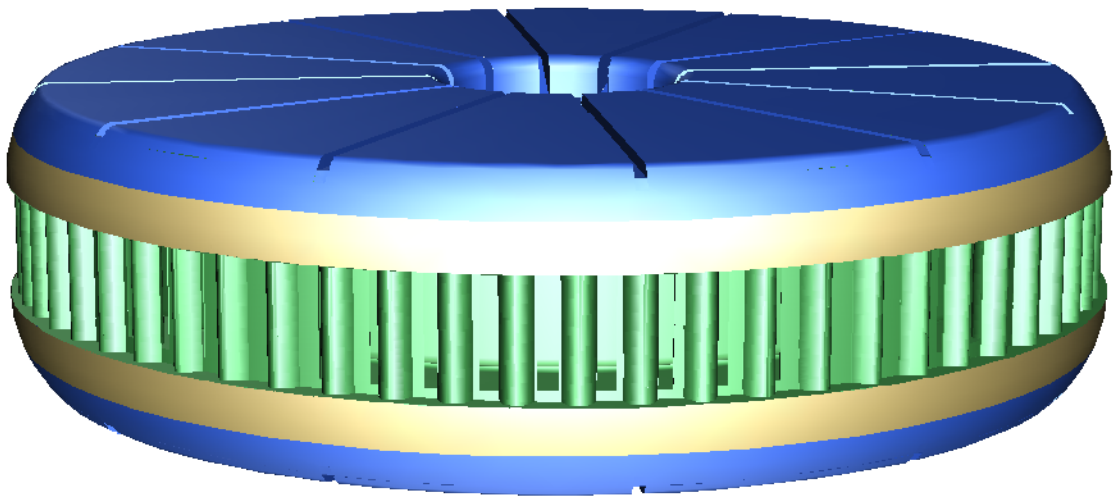
Para la definición de nuestro dominio computacional basta con un solo bloque (an. 11.7) dividido en las siguiente secciones: a) vértices, lista estándar de las coordenadas de los 8 puntos o esquinas del hexaedro que envuelve el motor eléctrico; b) bordes, que por defecto se corresponden con líneas rectas uniendo 2 vértices; c) bloques, lista de vértices ordenados, vector especificando el número de celdas en cada dirección y una expansión uniforme de 1 otorgada a cada dirección (sin refinamiento); contornos, parches o regiones donde establecer las condiciones de contorno y que podemos sortear por tratarse este de un dominio exterior al motor.

Empleando la malla estructurada del dominio en bruto resultante de la utilidad blockMesh, snappyHexMesh se encarga de esculpirla siguiendo unos parámetros de refinamiento y ajustando la malla a la geometría proporcionada.

Para la definición de la geométrica de partida se ha optado por la técnica de triangulización de las superficies implicadas, mediante una aproximación STL de la representación CAD. Esta malla 2D de tipo no estructura puede ser exportada directamente desde CATIA y contiene información sobre la posición de los tres vértices que conforman cada triángulo de la superficie y la normal a la faceta.

Dado que en el desarrollo de nuestro análisis MRF solamente entran en juego las superficies en contacto directo con el fluido (aire) interno del motor, será suficiente con aplicar la estereolitografía a las caras internas del rotor y externas del estátor. Lo que dará como resultado la división de la geometría en 3 regiones: a) fixedWall, pared exterior del estátor y que se mantendrá inmóvil respecto al marco de referencia absoluto; b) rotatingWall, pared interior al rotor que se encuentra girando a una velocidad de rotación  $\omega$  en referencia al marco estacionario inercial; c) rotatingZoneWall, pared interna del rotor que es vista en reposo desde el marco de referencia relativo en rotación.

Un inciso de gran relevancia para que snappyHexMesh trabaje de forma adecuada durante la generación de la malla está en la constatación de que el sentido de las normales de los elementos sea el apropiado. Como regla general las normales a las superficies deben apuntar hacia el dominio fluido.





Previamente a disponer de snappyHexMesh, y una vez que los archivos de geometría STL se encuentran disponibles en la carpeta constant/triSurface del caso, se hace uso de la utilidad surfaceFeatures. Esta lee la configuración de su archivo de diccionario system/surfaceFeaturesDict (an. 11.8), asumiendo la extracción de bordes afilados de acuerdo a un criterio de ángulo y que pueda suscitar problemas en el mallador snappy.

En primer lugar y a nivel superior el directorio snappyHexMeshDict (an. 11.9) incluye conmutadores que controlan las distintas etapas del proceso: a) castellatedMesh, situado en “on”, activa la creación de la malla tipo castellated; b) snap, dispuesto a “on”, activa la etapa de ajuste de superficie; c) addLayers, cambiado a “off”, desactiva la inserción de la capa de superficie. Por otro lado, cuenta con subdirectorios individuales para cada proceso:

- a) Geometría: describe todas las superficies trianguladas de entrada. Cada superficie y región de una superficie se convierte en un parche separado en la malla final.
- b) CastellatedMeshControls: partiendo de la malla hexagonal pura nativa proporcionada por blockMesh, aplica la división y refinamiento de celdas. La estrategia consiste en refinar la malla cerca de las entidades y la superficie geométrica para facilitar la fase de ajuste posterior y, al mismo tiempo, permitir la menor distorsión de malla posible. El refinamiento es especificado acorde a bordes (nivel 3) y superficies (desde mínimo nivel de refinamiento 3 a máximo 5). A través de la función locationInMesh se identifica una ubicación en la malla final, desde la cual snappyHexMesh marcará y mantendrá todas las celdas conectadas. Para controlar que los cambios de tamaño de celda sean suaves y no existan cambios consecutivos de nivel de refinamiento se establece un valor de 3 para el número de capas intermedias de celdillas (nCellsBetweenLevels) entre niveles sucesivos de refinamiento. Otros parámetros son presentados en el archivo de configuración snappyHexMeshDict.cfg (an. 11.10) usado.
- c) SnapControls: esta fase de ajuste implica proyectar y mover los vértices de los elementos sobre la superficie geométrica como proceso iterativo para eliminar irregularidades de malla. Si las adaptaciones invalidan los criterios de calidad de la malla, el paso se deshace y se intenta de nuevo utilizando parámetros modificados. La secuencia asegura que la malla resultante alcance una calidad mínima, a expensas de la complejidad de la geometría. Valores predeterminados (nSmoothPatch=3, tolerance=2, nSolverIter=100 y nRelaxIter=5) han sido seleccionados para el método, tal y como viene reflejado por el archivo de configuración. Además, la función explícita ha sido elegida, la cual autoriza a ajustar la malla a las líneas características extraídas a través de la utilidad surfaceFeatures.
- d) AddLayersControls: si la adición de capas de malla fuese imprescindible, capas de celdillas prismáticas podrían ser insertadas en los huecos originados al contraer la malla en la última etapa del proceso.

La calidad de la malla juega un papel fundamental cuando se trata de precisión, estabilidad y eficiencia de la solución. Por ello, meshQualityDict (an. 11.11) es usado como diccionario para definir los valores límite que pueden ser encontrados durante el proceso de mallado para garantizar que la malla resultante tenga la calidad suficiente para el cálculo posterior. Los límites afectan predominantemente a las regiones con mayor probabilidad de incurrir en distorsión de malla local. Si la malla viola alguno de los valores límite, intenta volver a aplicar los cambios ofensivos con configuraciones actualizadas para ciclos adicionales. Brevemente, algunos de los aspectos de calidad más destacados son:

- a) Ortogonalidad: acota el ángulo máximo de no ortogonalidad de la cara, es decir, el ángulo formado entre la línea que conecta dos centros de celda y la normal de su cara común.
- b) Oblicuidad: relacionado generalmente con caras pequeñas, y se define como la medida de la distancia entre la intersección de la línea que conecta dos centros de celda con su cara común y el centro de esa cara.
- c) Relación de aspecto: está relacionado con la cuadratura de las celdas, por ejemplo, para un cubo, la relación de aspecto es igual a uno. En el caso de una cara mide la relación entre la longitud más larga y la más corta, mientras que para una celda mide la relación entre las áreas de la caja delimitadora de la celda y el volumen de la celdilla.

Una vez finalizada la generación de malla mediante snappyHexMesh, dispondremos del archivo cellLevel ubicado en la carpeta contant/polyMesh, y que comprende el nivel de refinamiento del conjunto de celdillas que conforman el dominio fluido. Por el momento contamos con un único grupo de celdillas que pertenecen al marco de referencia inercial. Para el listado de un nuevo subconjunto de celdas concerniente al marco de referencia en rotación se hace necesario la creación de cellZones (regiones particulares en la malla a las que aplicar propiedades únicas en simulación) mediante el uso del diccionario para tal efecto topoSet (an. 11.12).

A través de topoSet seleccionamos el conjunto de elementos del dominio fluido cuyo centro de celdilla se encuentra contenido en el interior del cubo delimitador indicado, creando una zona (rotatingZone-1 y 2) por cada conjunto elegido. De esta manera queda concretado el grupo de celdillas rotatingZone-1 (y su grupo simétrico rotatingZone-2) que se hallan en rotación, pero que desde el punto de vista del sistema de referencia no inercial se encuentra en reposo pues se encuentran girando al unísono.

Los límites que determinan la región giratoria (rotatingZone) son aquellos que envuelven el volumen que resultaría de una revolución completa de las aletas del cubremotor, es decir, aquel cilindro de radio, la dimensión exterior de las aletas ( $r_{\max}=126$  mm) y longitud, la máxima anchura ( $w_{\max}=10$  mm) dada para la base de las aletas.

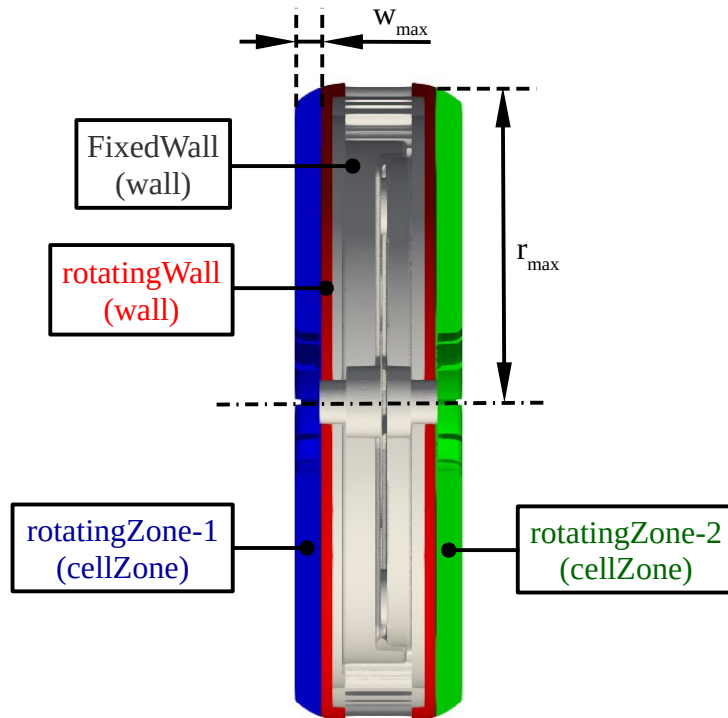


Figura 20. Sección en el plano XY mostrando las regiones giratorias

## 5.6 Pre-procesado caso MRFSimpleFoam

La estructura básica de un caso estándar de OpenFOAM normalmente está formada por un conjunto de 3 directorios: time (0), constant y system, que contiene el conjunto de archivos necesarios para ejecutar una aplicación. En el directorio de tiempo 0, se establece las condiciones iniciales y de contorno; en constant se proporciona información sobre la malla y propiedades del fluido; y system incluye el control de iteraciones, esquemas numéricos y parámetros de los solvers [34].

A continuación, describiremos las peculiaridades de cada uno de estos 3 directorios aplicado a nuestro caso MRF.

### 5.6.1 Directorio “time”

En OpenFOAM los campos tensoriales (datos de velocidad, presión, etc) siempre deben inicializarse, incluso cuando la solución no lo requiera estrictamente, como es el caso de los problemas en estado estacionario. Para ello, hacemos uso de la entrada internalField, que al ser elegida de tipología uniforme, un único valor (escalar o vectorial) es asignado a todos los elementos contenidos dentro del campo.

Primeramente, debemos considerar que, con el propósito de aplicar condiciones de contorno, estas generalmente se dividen en un conjunto de parches. Un parche puede incluir una o más áreas encerradas de la superficie límite que no necesariamente necesitan estar conectadas físicamente. Además se asigna un tipo a cada parche como parte de la descripción de la malla y que

es definido en términos de la geometría. Por nuestra parte la función tipo wall es asignada a los 3 parches (fixedWall, rotatingWall y rotatingZoneWall) en el archivo constant/polyMesh/boundary (an. 11.13).

BoundaryField es un diccionario que contiene un conjunto de entradas cuyos nombres corresponden a cada uno de los parches referenciados en el archivo boundary. La entrada obligatoria, tipo, describe la condición de campo del parche especificado: a) básica o primitiva, y b) derivada. Las condiciones de contorno primitivas son entre otras, fixedValue, que es relevante para la condición de contorno de Dirichlet; zeroGradient, que es la condición de Neumann; y la tipo calculated, que se calcula a partir de otros campos. La condición de contorno derivada es una condición de parche compleja, derivada del tipo primitivo (noslip, inletOutlet, ...). Las entradas restantes corresponden al tipo de condición de campo del parche seleccionado y normalmente pueden incluir datos de campo que especifican las condiciones iniciales.

Nos encontraremos con 2 archivos de campo, p y U, debido al carácter incompresible, estacionario y laminar de nuestro flujo; mientras este número aumenta a 5 al tratarse de un fluido en régimen turbulento (p, U, k, epsilon y nut).

Dando comienzo a un flujo en régimen laminar, tanto la velocidad (an. 11.14) de naturaleza vectorial como la presión (an. 11.15) de índole escalar son inicializadas a un valor nulo. Por otro lado, en relación a la clase de condiciones de contorno concedidas:

- Para la velocidad
  1. rotatingZoneWall: la básica fixedValue por su condición estacionaria en referencia al marco de referencia en movimiento.
  2. rotatingWall: la derivada rotatingWallVelocity por su condición rotativa respecto al marco de referencia absoluto. Las propiedades origin, axis y omega son requeridas; y marcan el origen de rotación en coordenadas cartesianas, el eje de rotación y la velocidad angular (rad/s) del marco no inercial, respectivamente.
  3. fixedWall: la derivada noslip, la cual fija la velocidad a cero en las paredes.
- Para la presión: la condición de contorno zeroGradient es aplicada a las tres entidades, asignando una condición de gradiente cero desde el campo interno del parche a las caras del parche.

Adicionalmente, en un análisis propio turbulento, en la carpeta 0 se nos presentará: a) k (an. 11.16), la energía cinética turbulenta; b) epsilon (an. 11.17), la tasa de disipación de energía turbulenta y c) nut (an. 11.18), la viscosidad turbulenta. Las condiciones iniciales  $k=0.375$ ,  $\epsilon=0.54$  y  $\nu_t=0$  son establecidas. Y en los tres campos los siguientes modelos de función pared son definidos como sigue:

- k: kqRWallFunction, proporciona una envoltura simple alrededor de la condición de gradiente cero, adecuada para el caso de un flujo de alto número de Reynolds utilizando funciones de pared.
- epsilon: epsilonWallFunction, proporciona una restricción de pared en epsilon para modelos de turbulencia de número de Reynolds alto.
- nut: nutWallFunction, condición para flujos de alto número de Reynolds basados en energía cinética de turbulencia cercana a la pared.

El solver crea los subdirectorios de tiempo correspondientes para cada solución de tiempo resuelta. El nombre de cada directorio de tiempo se basa en el tiempo simulado en el que se escriben los datos.

### 5.6.2 Directorio “constant”

Este contiene una descripción completa de la malla en el subdirectorio polyMesh. La descripción de polyMesh se basa en caras, las caras internas conectan 2 celdas y las caras exteriores se dirigen a una celda y un parche. Por lo tanto, a cada cara se le asigna una celda propietario y una celda vecino para que la conectividad a través de una cara determinada pueda ser descrita simplemente mediante estas etiquetas propietario y vecino. En los límites, la celdilla conectada es la propietaria y al vecino se le asigna la etiqueta -1. Los principales archivos integrantes son:

- points: lista indexada de vectores describiendo los vértices de las celdas.
- faces: lista de caras, cada cara a su vez es una lista de 4 elementos de índices a vértices, de la lista de puntos anterior.
- owner: lista de etiquetas de celda propietario, donde el índice de entrada está directamente relacionado con el índice de la cara, de modo que la primera entrada en la lista es la etiqueta de propietario para la cara 0.
- neighbour: lista de etiquetas de celdillas vecinas.
- boundary: ya explicado con anterioridad, trata de una lista de parches, que contiene una entrada de diccionario para cada parche.

En OpenFOAM, los solvers que no incluyen la transferencia de calor, introducen una biblioteca de modelos para viscosidad cinemática  $\nu$ . Conocido tratarse de un fluido newtoniano (aire bajo condiciones normales de presión y temperatura), la viscosidad puede considerarse constante y designada a través de la variable  $\nu = 1.5 \cdot 10^{-5} \text{ m}^2/\text{s}$  en el archivo transportProperties (an. 11.19).

El directorio momentumTransport introduce el modelado de turbulencia. En el cual se encuentra la palabra clave simulationType, que controla el tipo de modelado de turbulencia que se utilizará, ya sea: a) laminar (an. 11.20), no utiliza modelos de turbulencia o b) RAS (an. 11.21), usa el modelado Reynolds-

Average Simulation. Para la puesta en marcha del modelo RAS se requieren de varias entradas:

- `model`: detalla el modelo de turbulencia k-epsilon empleado.
- `turbulence`: activación de la resolución del modelado de turbulencias.
- `printCoeffs`: muestra los coeficientes del modelo por terminal al iniciar la simulación.

La región en el que el enfoque MRF tiene efecto se define a través de la función `cellZone` del directorio `MRFProperties` (an. 11.22). Donde el nombre de cada zona viene establecido por el apelativo (`rotatingZone-1` o `rotatingZone-2`), empleado en `topoSetDict` durante la conversión del conjunto de celdillas a una `cellZone`. Por otro lado, la entrada `nonRotatingPatches` puntualiza, en caso de existir, los parches pertenecientes a la `cellZone` que no se hallan en rotación. Otros parámetros; coincidentes con los ya comentadas para la condición de contorno `rotatingWallVelocity`; son las propiedades `origin`, `axis` y `omega`.

El archivo de usuario `initialConditions` (an. 11.23) es solamente dedicado a la definición de la variable velocidad angular, comúnmente utilizada por varios directorios en el caso.

### 5.6.3 Directorio “system”

OpenFOAM comienza todas las ejecuciones configurando una base de datos. La base de datos controla las entradas/salidas y, dado que la salida de datos generalmente se solicita a intervalos de tiempo durante el cálculo, el tiempo es una parte inextricable de la base de datos. El diccionario `controlDict` (an. 11.24) establece los siguientes parámetros de entrada:

- `application`: señala que se usará el solver `simpleFoam`.
- `startFrom`, `startTime`: controlan que la iteración de inicio de la simulación coincida con la de aquel directorio más recientemente escrito. Por defecto, el instante de partida será la carpeta 0.
- `stopAt`, `endTime`: controla que el tiempo de simulación acabe para la iteración final de 5000 si previamente no se ha logrado la convergencia.
- `deltaT`: paso iterativo unitario.
- `writeControl`, `writeInterval`: sincroniza el almacenamiento de resultados para que se produzca cada 200 iteraciones.
- `purgeWrite`: representa el número máximo de directorios temporales que serán almacenados durante el cálculo.
- `writeFormat`, `writeFormat`, `writeCompression`: detalla el formato ASCII sin compresión para los archivos de datos, utilizando 8 cifras significativas.

- timeFormat, timePrecision: especifica el formato y precisión con respecto a la numeración de los directorios de tiempo.
- runTimeModifiable: fomenta que este diccionario se vuelva a leer con el comienzo de cada iteración, lo que permite modificar los parámetros durante la simulación.
- functions residuals: por defecto plotea los valores residuales de presión y velocidad. Para añadir nuevos campos al post-procesado, como por ejemplo k y epsilon, se requiere del directorio residuals (an. 11.25).

El diccionario fvSchemes (an. 11.26) establece los esquemas numéricos para los términos (gradientes, divergencias, laplacianas, ...) de las ecuaciones, que se calculan durante la simulación. El conjunto de términos, para los cuales se deben especificar esquemas numéricos, se subdivide en las siguientes categorías:

- timeScheme: con steadyState se establece la primera derivada temporal ( $\partial/\partial t$ ) a cero.
- grandSchemes: subdirectorio que contiene los términos gradiente ( $\nabla$ ).
  1. Mediante la entrada Gauss se especifica la discretización estándar de volúmenes finitos de integración Gaussiana que requiere interpolación de valores desde los centros de las celdillas al centro de las caras. El esquema de interpolación viene dado por la entrada linear, es decir, interpolación lineal o diferenciación central.
  2. El esquema cellLimited limita el gradiente de tal forma que cuando los valores de las celdillas se extrapolan a las caras usando el gradiente calculado, los valores de las caras no caen fuera de los límites de los valores de las celdas circundantes. Se especifica un coeficiente límite después del esquema subyacente para el cual 1 garantiza la delimitación (se anula la discretización de términos de gradiente específicos para mejorar la estabilidad)
- divSchemes: subdirectorio conteniendo los términos divergencia ( $\nabla \cdot$ ).
  1. El hecho de que términos que son fundamentalmente diferentes residan en un subdirectorio significa que el esquema default está establecido a none.
  2. Esquemas basados en la integración de Gauss, utilizando el flujo volumétrico phi y el campo advectivo (U, k, epsilon) interpolado a las caras de la celda mediante un esquema de primer orden acotado.
  3. El término no advectivo, de naturaleza difusa, usa la integración de Gauss con interpolación lineal.

- snGradSchemes: abarca los términos de gradiente normal a la cara de la celda.
  1. El esquema corrected añade una corrección no ortogonal explícita a la componente ortogonal para mantener la precisión de 2º orden.
  2. La solución se estabiliza mediante la aplicación del esquema limited a la corrección. El coeficiente  $\psi=0.33$  ofrece gran estabilidad.
- laplacianSchemes: comprende los términos laplacianos ( $\nabla^2$ ).
  1. Se emplea el esquema de discretización de Gauss con interpolación lineal para la interpolación del término difusión en las ecuaciones de momento.
  2. Basado en el mismo nivel de no ortogonalidad que snGradSchemes.
- InterpolationSchemes: subdirectorio que engloba aquellos términos que son interpolaciones de valores desde los centros de las celdas a los centros de las caras, usados en la interpolación de la velocidad a los centros de las caras para el cálculo del flujo phi. La entrada linear revela que la interpolación lineal es elegida.

Los métodos de procesamiento numérico para resolver cada ecuación matricial discretizada, las tolerancias y los algoritmos se controlan desde el diccionario fvSolution (an. 11.27). Los subdirectorios definidos son explicados a continuación:

- solvers (lineales)
  1. Variable p: el método generalizado Geometric-Algebraic Multi-Grid (GAMG) utiliza el principio de generar una solución rápida para la presión en una malla con un número reducido de celdas, mapear esta solución en una malla más fina, usándolo como una suposición inicial para obtener una solución precisa en la malla fina. Es decir, el solver GAMG comienza con un tamaño de malla aproximado de  $nCellsInCoarsesLevel=500$  elementos y engrosa/refina la malla en etapas. Con  $tolerance=0.05$  y  $relTol=10^{-8}$ , el cálculo es parado si los residuales caen por debajo de la tolerancia o la relación entre los residuales actuales y los iniciales caen por debajo de la tolerancia relativa. Finalmente, un suavizado tipo Gauss-Seidel es introducido.
  2. Variables U, k y epsilon: solver iterativo que utiliza el suavizador GaussSeidel para hacer converger la solución de los campos velocidad, energía cinética y tasa de disipación turbulenta a la tolerancia requerida ( $tolerance=0.1$  o  $relTol=10^{-7}$ ). En orden a mejorar la eficiencia, el residual se evalúa después de cada 2 iteraciones de suavizado ( $nSweeps=2$ ).



- SIMPLE: algoritmo iterativo para acoplar las ecuaciones de conservación de la masa y el momento en problemas de estado estacionario. El bucle de iteración se controla mediante los siguientes parámetros de entrada:
  1. nNonOrthogonalCorrectors: establece que no es necesario repetir soluciones de la ecuación de presión, basta con la primera solución alcanzada, para actualizar la corrección explícita no ortogonal.
  2. pRefValue, pRefCell: en un sistema incompresible cerrado, la presión es relativa, lo que importa es el rango de presión, no los valores absolutos.
- residualControl: se ocupa de concluir el caso cuando el residual inicial de las ecuaciones de campo cae por debajo del umbral especificado. Para el caso de flujo laminar se implanta  $U=10^{-6}$  y  $p=k=epsilon=10^{-5}$ , mientras para el régimen turbulento se decreta unos valores ligeramente más laxos, como son  $U=10^{-4}$  y  $p=k=epsilon=10^{-3}$ .
- RelaxationFactors: técnica utilizada para mejorar la estabilización del cálculo. Funciona limitando la cantidad que cambia una variable de una iteración a la siguiente, ya sea modificando la matriz de solución antes de resolver un campo o modificando el campo directamente.
  1. fields: un factor sutilmente alto ( $\alpha_R=0.7$ ) es utilizado para la presión y de esta forma asegurar la estabilidad.
  2. equations: factor moderadamente bajo ( $\alpha_R=0.2/0.1$ ) con la intención de no ralentizar el proceso iterativo en exceso.

Por último, previamente al lanzamiento de la simulación, se opta por una alternativa de enorme versatilidad que ofrece OpenFOAM y que merma notablemente el tiempo de cálculo, esto es la ejecución en paralelo en procesadores distribuidos. Este método de computación en paralelo se conoce como descomposición del dominio, ya que tanto la geometría como sus campos asociados se dividen en fragmentos y se asignan a procesadores separados para su resolución.

La malla y los campos se descomponen sirviéndose del directorio para tal efecto, decomposeParDict (an. 11.28). Para el cual se detalla el siguiente conjunto de parámetros:

- numberOfSubdomains: número total de subdominios ( $N=8$ )
- method: descomposición geométrica de tipo hierarchical, en la cual el dominio se divide en partes de manera direccional (2 fragmentos en dirección x, 2 en y, 2 en z) y de acuerdo al orden establecido.
- delta: factor límite de oblicuidad admisible para las celdillas.
- order: orden de descomposición xyz.

## 5.7 Resultados caso MRFSimpleFoam

En general, no existe un criterio obvio para la prueba de convergencia. Matemáticamente, uno puede considerar como regla de oro para un análisis MRF, que el valor residual de las ecuaciones debe ser menor que una cierta magnitud, a menudo menor que  $10^{-4}$  para U y  $10^{-3}$  para p, k y epsilon. Desde este punto de vista, el archivo de registro de todas las simulaciones muestra que el requisito de convergencia se acepta y satisface el valor predeterminado.

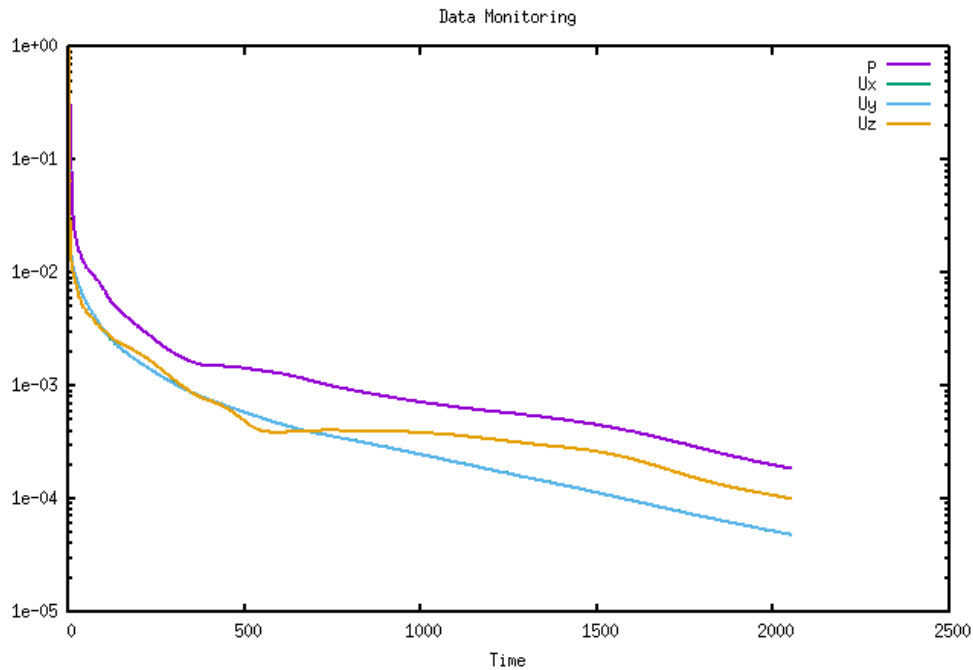


Figura 21: Residuales  $\omega=15$  rad/s,  $Re=1830$  (flujo laminar)

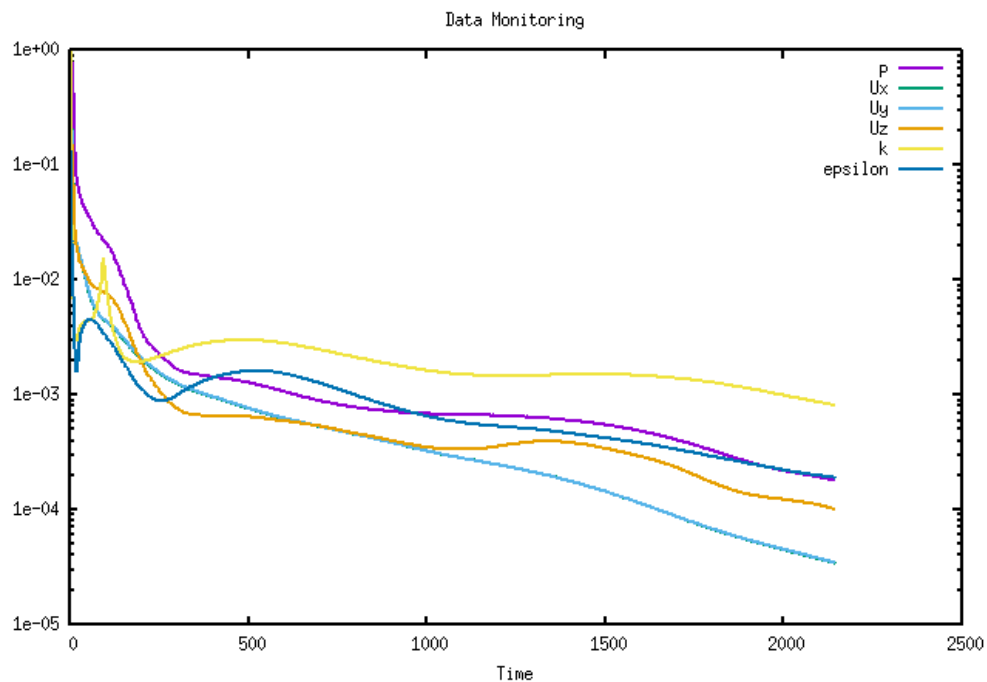


Figura 22: Residuales  $\omega=20$  rad/s,  $Re=2440$  (flujo transitorio bajo)

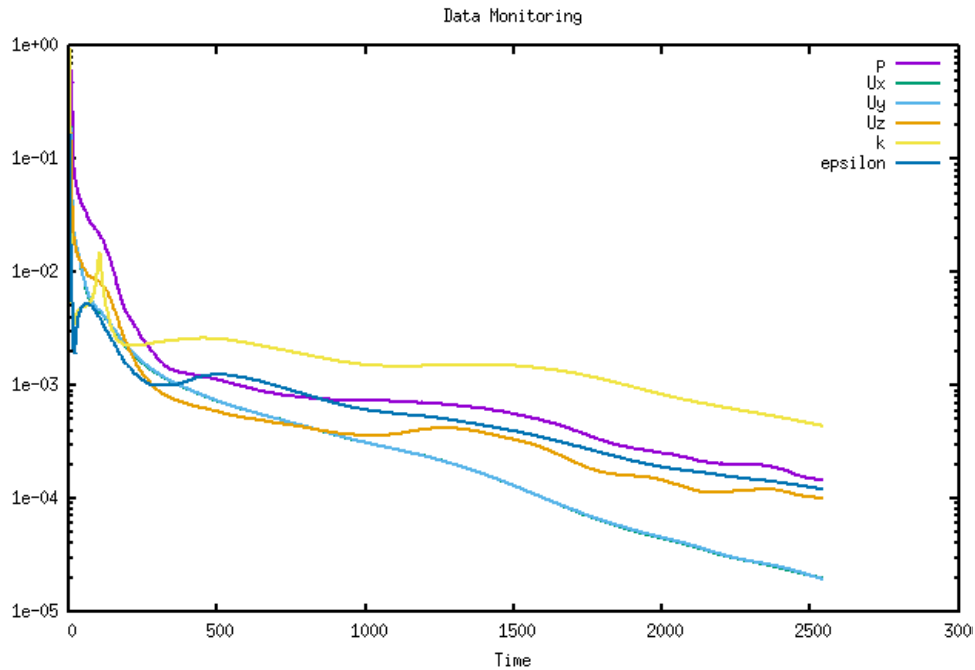


Figura 23: Residuales  $\omega=25$  rad/s,  $Re=3050$  (flujo transitorio medio)

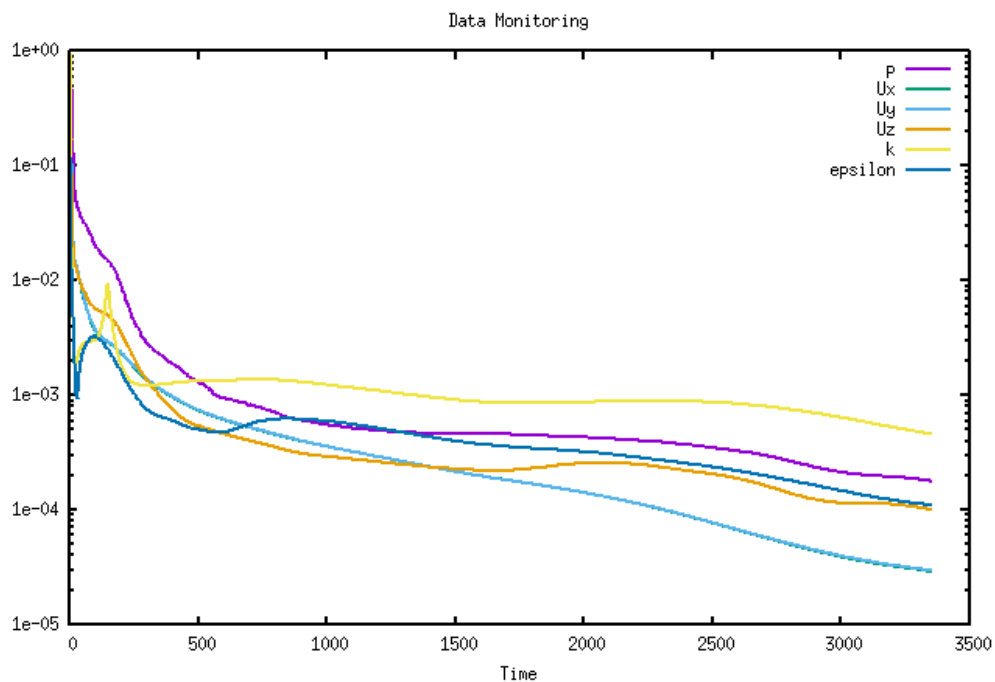


Figura 24: Residuales  $\omega=30$  rad/s,  $Re=3660$  (flujo transitorio alto)

Se puede notar que a medida que aumenta la velocidad angular y el flujo se vuelve más turbulento se requiere de un mayor número de iteraciones para alcanzar la convergencia. Llegando incluso a ser necesario el uso de un factor de relajación de 0.1, de la entrada ecuaciones del directorio fvSolution, para que el caso  $\omega=30$  rad/s se establezca logrando la solución. Recordemos que un valor de relajación bajo conduce a una convergencia más lenta, pero al mismo tiempo, conduce a una mayor estabilidad.

Esto viene motivado porque los problemas de MRF pueden ser difíciles de resolver debido a los grandes gradientes de flujo que resultan de la rotación de una zona del fluido, y la interacción entre cellZones adyacentes podría requerir del uso de factores de relajación inferiores a los predeterminados.

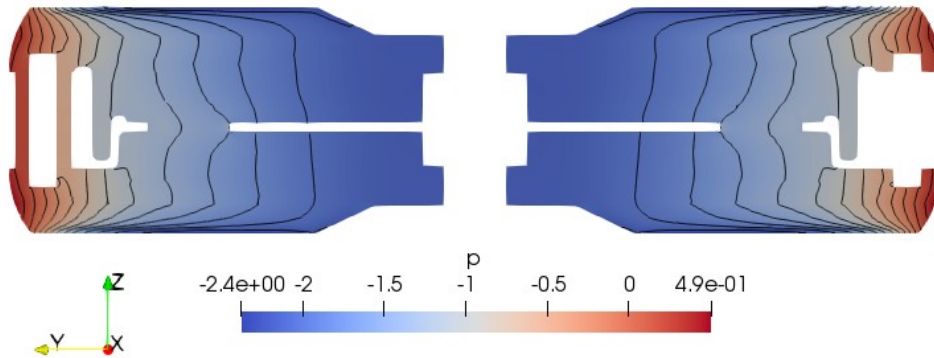


Figura 25: Sección contorno de presiones plano YZ para  $\omega=30 \text{ rad/s}$

Precisamente, son estos intensos gradientes de presión relativa (Figura 25) los que se observan en las proximidades de las aletas, donde se produce una fuerte depresión, tal y como evidencian las isobaras.

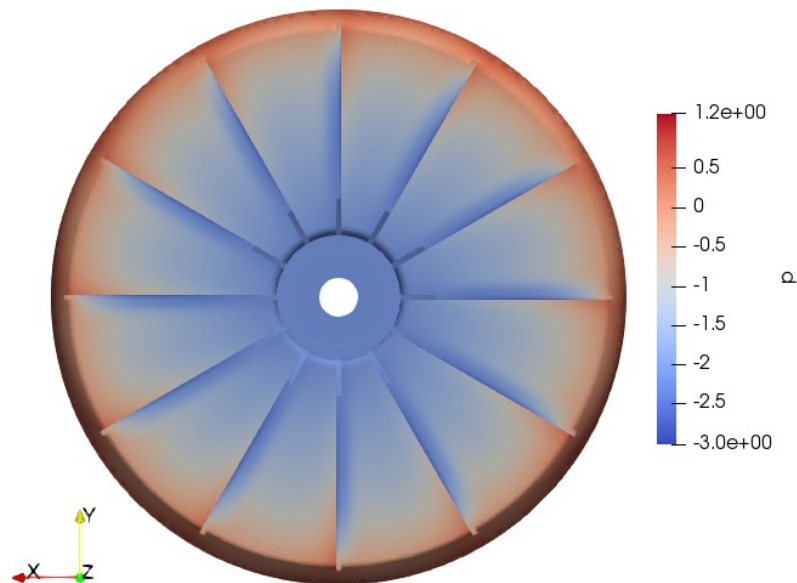


Figura 26: Distribución de presiones para  $\omega=30 \text{ rad/s}$

La figura 26 ilustra el comportamiento del flujo alrededor de las aletas cuando el rotor se encuentra girando en sentido horario. Esto es a menudo explicado usando la teoría del perfil alar y el teorema del momento. De acuerdo a la teoría del perfil aerodinámico, se debe generar una diferencia de presión alrededor de las aletas a medida que estas giran. Paralelamente, de acuerdo con el teorema de la cantidad de movimiento, la disminución de la presión conduce a que la velocidad del fluido aumente. Esto significa que el fluido se acelera a medida que se acerca a las aletas debido a la baja presión en la parte delantera y el aire continúa acelerándose mientras sale de la aleta.

A continuación (figura 27), se muestra la superposición del campo de velocidades sobre las líneas de corriente. En ella se aprecia claramente la trayectoria circular rotatoria descrita por las partículas en su movimiento y la velocidad *in crescendo* de estas partículas a medida que nos separamos del eje del motor, obedeciendo a la relación  $v=\omega r$  (con  $v=\omega r_{\max}$  como valor máximo de velocidad lineal).

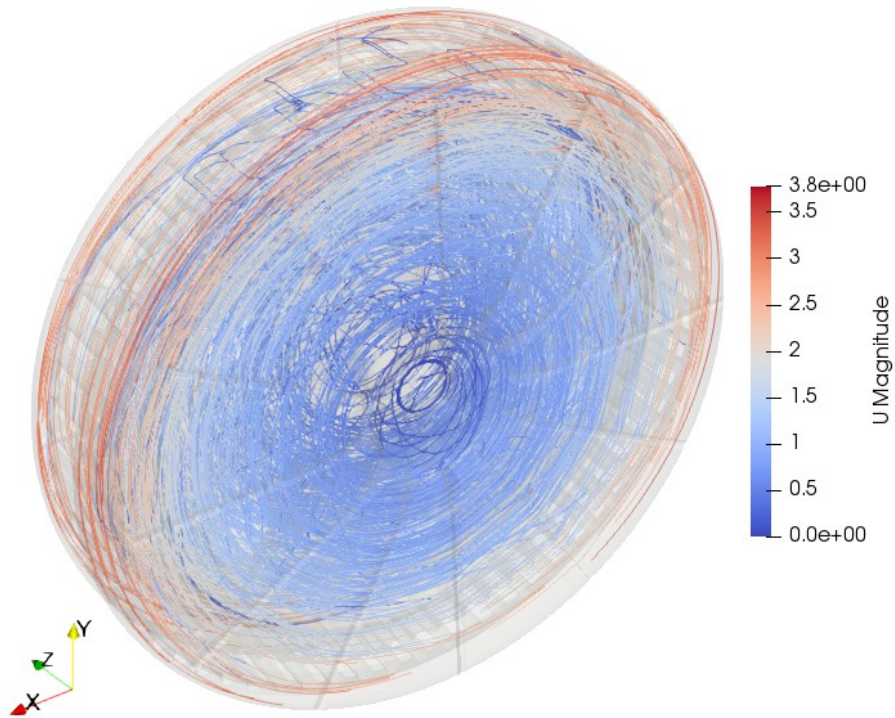


Figura 27: Líneas de corriente de velocidad para  $\omega=30$  rad/s

Lo cual también queda notoriamente patente al analizar la evolución de los contornos de velocidad para diferentes velocidades de giro, donde además, se puede percibir que la velocidad aumenta linealmente a razón de la velocidad angular, pasando de una velocidad máxima de 1.9 m/s para  $\omega=15$  rad/s hasta los 3.1 m/s para  $\omega=30$  rad/s.

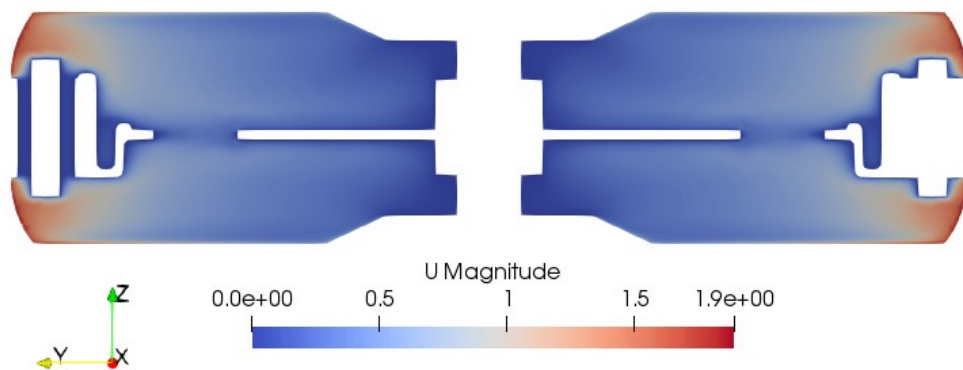


Figura 28: Sección YZ contorno de velocidades para  $\omega=15$  rad/s

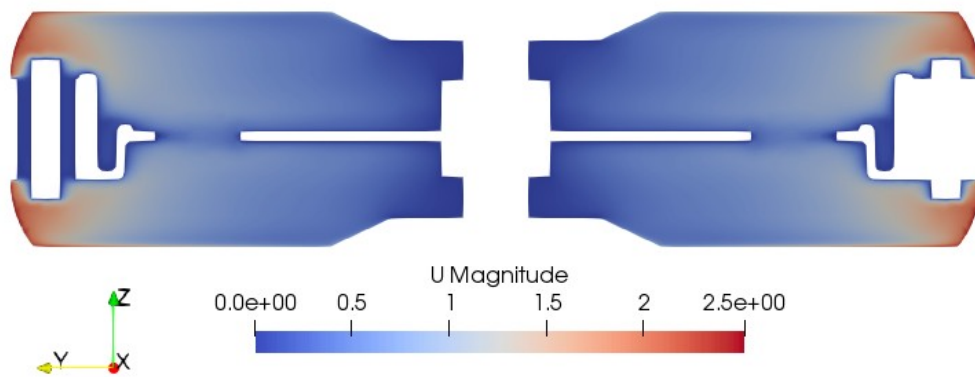


Figura 29: Sección YZ contorno de velocidades para  $\omega=20$  rad/s

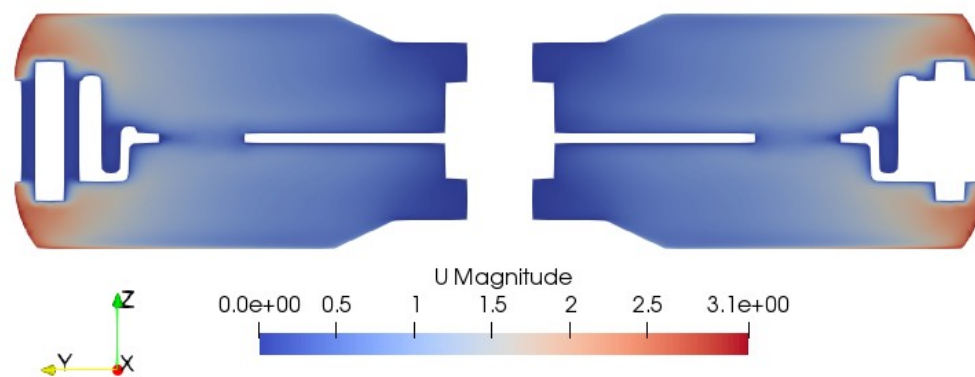


Figura 30: Sección YZ contorno de velocidades para  $\omega=25$  rad/s

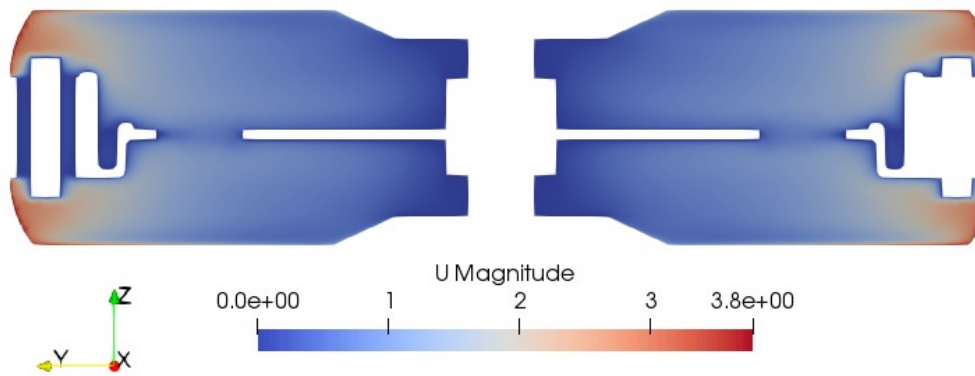


Figura 31: Sección YZ contorno de velocidades para  $\omega=30$  rad/s

Por último, escenificaremos la vorticidad, definida matemáticamente por el rotacional del campo de velocidades:

$$\xi = \nabla \times \mathbf{u}$$

Como se puede observar en la figura 32, la vorticidad se origina fundamentalmente en los contornos sólidos debido a que el aire no es capaz de deslizarse sobre las paredes, y luego se propaga ligeramente hacia el interior

del fluido, debido a que su existencia está íntimamente ligada a las tensiones tangenciales.

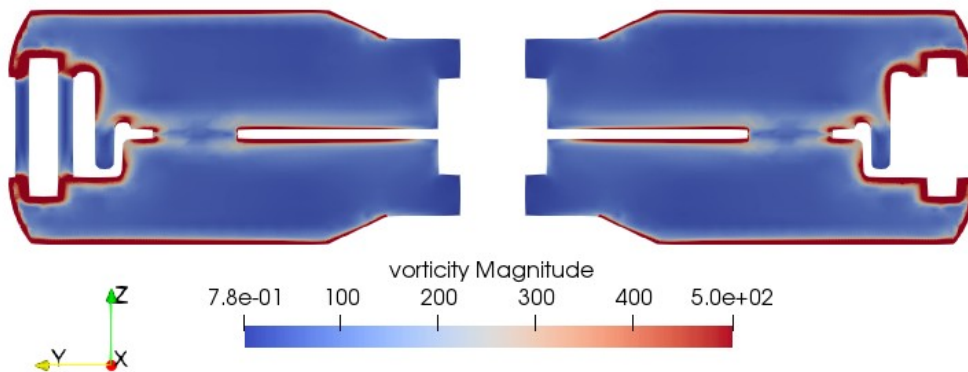


Figura 32: Sección YZ contorno vorticidad para  $\omega=30 \text{ rad/s}$

Singularmente, existen algunas regiones de intensa vorticidad en las inmediaciones al devanado y al núcleo de hierro; situación que se encuentra en sintonía con la hipótesis que ya barajábamos en el apartado 4.2, acerca de que la franja más estrecha entre las aletas y la bobina ( $w_{\min}$ ) sería de las regiones más propensas al desarrollo de un flujo turbulento.

## 6. Caso chtMRFMultiRegionFoam

### 6.1 Introducción

El calor es un fenómeno térmico que tiene lugar en la gran mayoría de motores eléctricos. En algunos motores sin escobillas es fundamental mantener el calor en un nivel permisible. Esto es especialmente cierto en motores de monociclos eléctricos, máquinas rotativas en las que la eficiencia varía indirectamente con la temperatura de funcionamiento.

Los motores BLDC son uno de los métodos más eficientes para crear energía mecánica a partir de otra forma de energía (la eficiencia del motor sin escobillas es muy alta en comparación con cualquier motor de combustión con valores que oscilan entre el 70% y el 90%). Sin embargo, esto no quita que existan pérdidas de calor que afecten a la eficiencia de nuestro motor.

El principal factor que afecta al incremento de la temperatura interna en el motor son las pérdidas en el cobre, seguidas de las pérdidas de energía en el núcleo [35]. El origen se describe a continuación:

- Pérdidas en el cobre: producto de la energía perdida en el devanado. Se relaciona directamente con la resistencia por fase.
- Pérdidas en el hierro: tienen en cuenta la corriente sin carga del motor. La corriente sin carga del motor es la corriente requerida para mantener un motor funcionando en una condición donde el trabajo desarrollado o potencia de salida es nula. Las pérdidas que producen una corriente sin carga provienen de la histéresis, las corrientes parásitas en el núcleo de hierro y las pérdidas mecánicas que se dan en los rodamientos.

El incremento de la temperatura por sí sola no produce una reducción en la eficiencia (siempre y cuando no sobrepasemos el límite de diseño del motor), sino que es el efecto de la temperatura sobre los imanes permanentes de NdFeB y el devanado de cobre, quien provoca una desmagnetización en los primeros y un aumento en la resistencia del segundo, induciendo una merma importante en el par motor.

Los imanes de NdFeB se utilizan en la construcción de motores EUC debido al alto rendimiento y bajo coste. La desventaja de este tipo de imanes es su alta sensibilidad a valores relativamente pequeños de la temperatura, lo que provoca efectos negativos sobre los parámetros del imán y del motor.

En el rango normal de temperatura de funcionamiento, a medida que aumenta la temperatura, la densidad de flujo residual y la intensidad del campo coercitivo intrínseco del imán disminuirán. Cuando se reduce la temperatura, la densidad de flujo y la intensidad del campo coercitivo volverán al valor original. Esta variación en la densidad de flujo residual del imán, junto con la variación en la resistencia de la bobina con la temperatura, influye en la capacidad de par a desarrollar y en la eficiencia del motor.



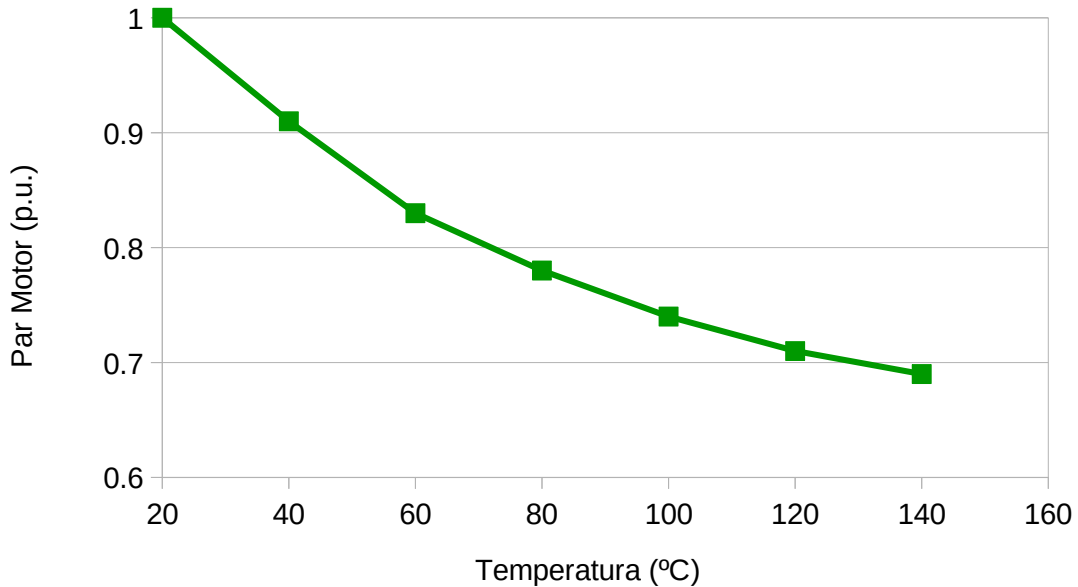


Figura 33: Par frente a temperatura de un motor BLDC genérico [36]

A lo que hay que añadir, que de acuerdo a la ecuación de Arrhenius [37], la tasa de fallo de un dispositivo eléctrico está exponencialmente relacionada con la inversa de su temperatura de funcionamiento, de forma que cuando aumenta ésta disminuye su vida útil.

Por todo esto, cobra bastante sentido llevar a cabo un análisis térmico del motor, pues las pérdidas generadas por el motor en rotación podrían dar lugar al aumento de la temperatura del motor, a través de diferentes formas de transferencia de calor. Si la temperatura de funcionamiento del motor es excesiva, cabría la posibilidad de que aparecieran multitud de problemas como: rotura del devanado del estátor, invalidación de la protección del devanado; desmagnetización de los imanes permanentes; estragos en el aislamiento; deterioro de los rodamientos; etc.

## 6.2 Algoritmo chtMultiRegionFoam

Las ecuaciones de conservación de la masa y momento para el campo fluido han sido ya discutidas en el apartado anterior, ahora nos centraremos en la tercera ecuación de conservación, conocida como la ecuación de la energía total, que en su estado estacionario puede venir dada en sus dos enfoques siguientes [38]:

- En términos de la energía interna específica, e

$$\nabla \cdot (\rho \mathbf{u} e) + \nabla \cdot (\rho \mathbf{u} K) + \nabla \cdot (\mathbf{u} p) = -\nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}) + \rho r + \rho \mathbf{g} \cdot \mathbf{u}$$

donde p es la presión,  $K=|\mathbf{u}|^2/2$  es la energía cinética específica (energía cinética por unidad de masa),  $\mathbf{q}$  es el vector de flujo de calor, y r es cualquier fuente de calor específica.

- En términos de la entalpía, h

$$\nabla \cdot (\rho \mathbf{u} h) + \nabla \cdot (\rho \mathbf{u} K) = -\nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}) + \rho r + \rho \mathbf{g} \cdot \mathbf{u}$$

siendo la entalpía la suma de la energía interna y la presión cinemática ( $h=e+p/\rho$ ).

El código fuente para resolver la ecuación de la energía en OpenFOAM es EEqn.H (an. 11.29), donde se implementa en términos de energía interna e y entalpía h, permitiendo al usuario elegir la variable de solución.

Además asumiendo el aire como un gas perfecto, se cumple la ecuación de estado:

$$\rho = \frac{p}{R_G T}$$

con  $R_G$  la constante de los gases ideales y T la temperatura absoluta.

Para las regiones sólidas, basta con resolver la ecuación de conducción, que viene dada en su forma estacionaria por:

$$\frac{1}{\rho c_p} \nabla \cdot (\kappa \nabla T) + \frac{q}{\rho c_p} = 0$$

donde  $c_p$  es el calor específico a presión constante,  $\kappa$  es la conductividad térmica y q la generación de energía por unidad de volumen.

Al ser la capacidad calorífica  $\rho c_p$  espacialmente uniforme:

$$\nabla \cdot (\alpha \nabla T) + \frac{q}{\rho c_p} = 0$$

siendo  $\alpha = \kappa / \rho c_p$  la difusividad térmica.

El archivo de cabecera encargado de la resolución de esta ecuación es solveSolid.H (an. 11.30). Bucle en el que la ecuación laplaciana es definida:

$$\nabla \cdot (\alpha \nabla T) = 0$$

ecuación que no considera la generación de calor. Por lo que, para ello, se hace uso de la funcionalidad fvOptions.

El solver chtMultiRegionFoam (an. 11.31) fue concebido para resolver problemas de transferencia de calor conjugada que involucran regiones sólidas y fluidas. La transferencia de calor conjugada se refiere a la transferencia de calor que se produce por interacción de sólidos y fluidos. En los sólidos, la conducción de calor es el proceso dominante, mientras que en los fluidos la transferencia de calor está dominada por la convección. La interacción entre las

regiones sólida y fluida se logra mediante el uso de una condición de contorno especial llamada mappedWall.

Aquí el solver sigue una estrategia de solución separada. Esto significa que las ecuaciones para cada variable que caracteriza el sistema se resuelven secuencialmente y la solución de las ecuaciones anteriores se inserta en la ecuación siguiente. Primero se resuelven las ecuaciones para el fluido usando la temperatura del sólido de la iteración anterior para definir las condiciones de contorno para la temperatura en el fluido. Después de eso, la ecuación para el sólido se resuelve usando la temperatura del fluido de la iteración anterior para definir la condición de contorno temperatura del sólido. Este procedimiento de iteración se ejecuta hasta la convergencia.

Para la materia ordinaria, la conducción y la convección son los mecanismos principales, ya que la transferencia de energía térmica por radiación solo representa una parte minúscula de la energía transferida. Pues, aunque la transferencia de energía por radiación aumenta con la cuarta potencia de la temperatura ( $T^4$ ) al ser la constante de proporcionalidad de Stefan-Boltzmann ( $\sigma=5.67 \cdot 10^{-8} \text{ W/m}^2 \cdot \text{K}^4$ ) tan reducida, independientemente del acabado superficial, se requerirán temperaturas superiores a varios miles de kelvin para que la radiación emitida sea significativa. Luego, aunque este algoritmo también es capaz de modelar la transferencia de energía debida al fenómeno de radiación, despreciaremos su contribución en nuestro caso, pues las temperaturas de trabajo son lo bastante bajas como para que la radiación térmica pueda llegar a ser relevante.

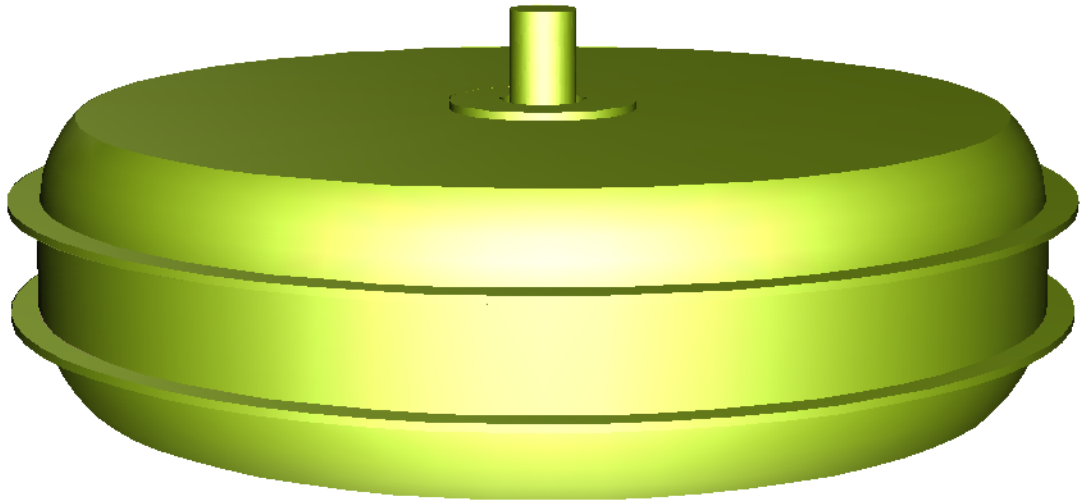
### 6.3 Mallado modelo chtMultiRegionFoam

Primeramente, se almacenan cada uno de los archivos de superficie que especifican los límites de la malla en el directorio triSurface, el cual contendrá un archivo de malla que establece los límites exteriores del dominio conjunto (CAD.stl), así como una malla de superficie por cada interfase común sólido-sólido o sólido-fluido (cap\_motor-bearing.stl, stator, stator\_coil-coil, etc).

Tal y como se comentó en el apartado anterior, la coherencia de las normales de los elementos 2D resulta primordial para la generación exitosa de la malla tetraédrica, de forma que las normales a los triángulos deben apuntar hacia el exterior de los sólidos, o lo que es lo mismo, hacia el interior del fluido, con la única excepción de las superficies de interfaz sólido-sólido que pueden manejar indistintamente cualquier sentido.

Creada la malla hexagonal estructurada base, partiendo de los mismos parámetros de entrada con la utilidad blockMesh, se hace uso de nuevo de los diccionarios surfaceFeaturesDict (an. 11.32) y snappyHexMeshDict (an. 11.33), pero ahora incluyendo cada una de las superficies que conforman entidades comunes o individuales de cada sólido.

En ocasiones, regiones particulares en la malla, como son los sólidos, requieren propiedades determinadas. Para ello, subconjuntos de celdillas de malla (cellZones) tienen que ser definidos. Con tal propósito se utilizan las



entradas `cellZone`, `cellZoneInside` (área seleccionada) e `insidePoint` (vector de un punto interno) que generan regiones para cada uno de los sólidos a través de `snappyHexMesh`.

Tal y como se puede deducir del bloque `snapControls`, de manera complementaria a la característica explícita, la entrada `multiregionFeatureSnap` es usada para ajustar automáticamente los bordes limítrofes de las superficies STL detectadas.

Una vez generada la malla, tendremos un único dominio determinado por la superficie exterior `CAD.stl` y un subgrupo de `cellZones` que representan cada uno de los sólidos. Sin embargo, para dividir la malla en los subdominios correspondientes a cada uno de los sólidos y al fluido interno, se necesitará de la utilidad de manipulación de malla `splitMeshRegions`. Como resultado, cada región (incluido el fluido) se define como un dominio. Muy importante, si queremos mantener, durante el mapeado de parches con `splitMeshRegions`, la estructura de interfaces especificada en `snappyHexMeshDict`, debemos de usar la opción `-useFacesZones`, la cual se sirve de las `faceZones` para parchear la interfaz entre regiones, en lugar de establecer un parche único.

Para la distribución de celdillas del dominio fluido a aquel marco inercial y relativo al cual asignar la velocidad angular, se ejecuta de nuevo el directorio `topoSetDict` con la peculiaridad ahora de que este archivo se debe ubicar en `sytem/fluid` y debemos indicar mediante terminal el dominio al cual queremos aplicar esta acción.

### 6.3 Pre-procesado caso `chtMultiRegionFoam`

Aunque se mantiene la estructura del caso (`time`, `constant` y `system`) expuesta en el apartado anterior, haremos hincapié en los nuevos directorios y describiremos las diferencias más llamativas respecto al `MRFSimpleFoam`.

#### 6.3.1 Directorio “0”

Carpeta compuesta por 9 dominios sólidos y 1 dominio fluido. Que a su vez se subdividen en 2 archivos de campo (`p` y `T`) para el ámbito sólido y 4 (`p`, `T`, `p_rgh` y `U`) en el caso de la región fluida.

Respecto a los sólidos, aunque carezca de sentido físico el campo de presión (`p`) aplicado a un cuerpo rígido, indistintamente, deber ser inicializado (an. 11.34). La condición de contorno básica `calculated` ha sido destinada a tal efecto. Por otro lado, para la temperatura (`T`), se distinguen 2 condiciones:

- `externalWallHeatFluxTemperature` (an. 11.35): adjudica una condición de flujo de calor para la temperatura en una pared externa. Las entradas requeridas son: a) el coeficiente de transferencia de calor, donde se ha considerado convección libre con el aire ( $h_a=10 \text{ W/m}^2\text{-K}$ ) debido a que la carcasa impide la convección forzada con el cubremotor; y b) la temperatura ambiente ( $T_a=300 \text{ K}$ ).

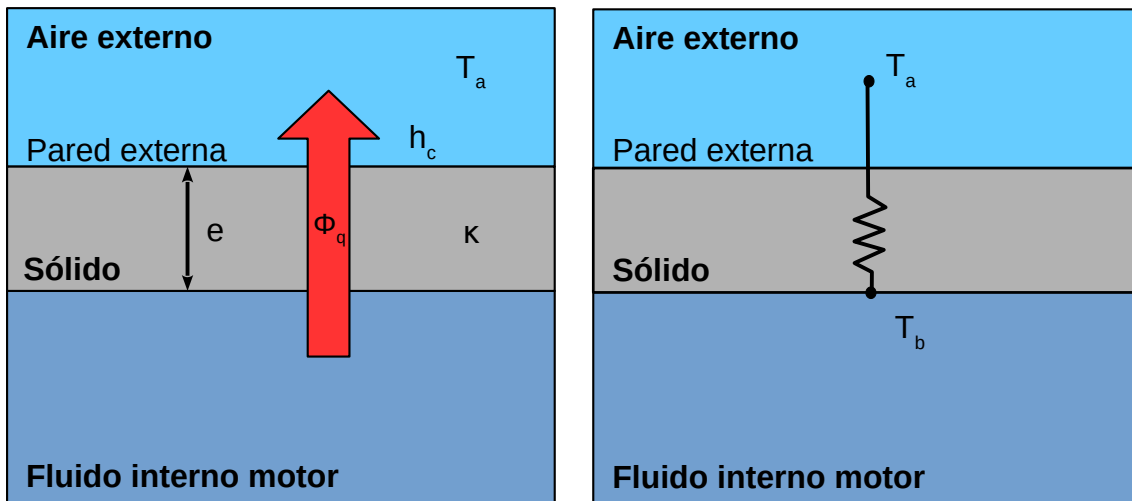


Figura 35: Condición flujo de calor pared externa

Especificado el coeficiente de película y la temperatura ambiente:

$$\phi_q = \frac{T_a - T_b}{R_{th}}$$

siendo  $\Phi_b$  el flujo de calor,  $T_b$  la temperatura en la cara interna del sólido,  $e_s$  el espesor sólido y  $R_{th}$  la resistencia térmica de transferencia de calor por convección y conducción, que viene dada por:

$$R_{th} = \frac{1}{h_a} + \frac{e_s}{\kappa}$$

- compressible::turbulentTemperatureCoupledBaffleMixed (an. 11.36): tipo mixto de condición de contorno para la temperatura, diseñada para acoplar regiones sólidas (interfaz sólido-sólido) y fluidas (sólido-fluido) térmicamente. Se ha supuesto contacto ideal, es decir, no se introduce resistencia de contacto entre regiones.

Un apunte relevante, llegados a este punto, es la simplificación n.º 3 que llevamos a cabo en el apartado 3. Pues para que la simplificación geométrica de eliminar el irrisorio gap de aire existente entre el rotor y el estátor tenga sentido térmico se debe imponer que esta pared de interface que separa los imanes permanentes del núcleo del bobinado sea adiabática, de modo que no pueda darse transferencia de calor a través de la misma. Para ello, se puede emplear la condición de contorno zeroGradient para el campo T (an. 11.37).

Retomando el dominio fluido, la definición del campo de presión (an. 11.38) coincide con el detallado para la región sólida. Mientras, para el campo temperatura (an. 11.39), la condición de contorno de acoplamiento es aplicado a todos los parches. El archivo de velocidades U difiere poco del exhibido en el capítulo 5.6.1, con la única discrepancia de que ahora hay más parches a los que aplicarle la condición de pared tipo noslip (an. 11.40). Por último, queda la presión hidrostática ( $p_{rgh}$ ), que como se puede apreciar emplea una condición denominada FixedFluxPressure para todos los límites de pared (an. 11.41).

### 6.3.2 Directorio “constant”

Complementariamente a los archivos ya conocidos, se nos presenta el directorio `thermophysicalProperties` que es leído por cualquier solver que utilice la biblioteca de modelos termofísicos. Un modelo termofísico se construye en OpenFOAM como un sistema de presión temperatura ( $p$ - $T$ ) a partir del cual se calculan otras propiedades.

Previamente a entrar más en detalle con la definición de esta clase, es primordial conocer los materiales de los componentes a modelar, así como, su caracterización:

- Aluminio: cubremotor y rodamientos (an. 11.42)
- NdFeB: imanes permanentes (an. 11.43)
- Cobre: bobina (an. 11.44)
- Núcleo: hierro (an. 11.45)
- Acero: eje y estátor (an. 11.46)

La entrada inicial del diccionario, llamada `thermoType`, es obligatoria y especifica el paquete de modelado termofísico a emplear en la simulación. En orden a ensamblar el modelo correspondiente a las regiones sólidas (an. 11.47) las subentradas que le acompañan son las siguientes:

- `type`: `heSolidThermo`, cálculo del modelo termofísico general basado en entalpía o energía interna del sólido.
- `mixture`: `pureMixture`, opción que representa una mezcla de composición fija, es decir, sin reacciones.
- `transport`: `constIso`, establece la plantilla necesaria para la definición de una conductividad térmica  $\kappa$  constante e isotrópica.
- `thermo`: `eConst`, asume un calor específico (a volumen constante)  $c_v$  y un calor de fusión  $H_f$  constante.
- `equationOfState`: `rhoConst`, asigna una densidad constante.
- `specie`: `specie`, contiene el número de moles (`nMoles`) de la especie y la masa molecular (`molWeight`).
- `energy`: `sensibleInternalEnergy`, especifica que la ecuación de la energía interna será usada en la solución, y que el campo temperatura será calculado a partir de esta variable.

En referencia al dominio fluido (an. 11.48), el modelo termofísico relativo al aire se construye a partir de:

- type: heRhoThermo, modelo termofísico para la energía de una mezcla de composición fija basado en la densidad.
- mixture: pureMixture, no se dan reacciones.
- transport: sutherland, calcula la viscosidad dinámica  $\mu$  como una función de la temperatura, de acuerdo, con la ecuación de Sutherland.

$$\mu = \frac{A_s \sqrt{T}}{1 + T_s/T}$$

donde  $A_s$  y  $T_s$  simbolizan el coeficiente y la temperatura de Sutherland, respectivamente.

- thermo: janaf, calcula el calor específico  $c_p$  en función de la temperatura. La función es válida entre un límite inferior ( $T_l$ ) y superior de temperatura y ( $T_h$ ). Dos conjuntos de coeficientes son establecidos, el primero para temperaturas comprendidas por encima de una temperatura común  $T_c$  y por debajo de  $T_h$ , el segundo para temperaturas por debajo de  $T_c$  y por encima de  $T_l$ . La función en cuestión se muestra a continuación.

$$c_p = R_G (((((a_4 T + a_3) T + a_2) T + a_1) T + a_0))$$

siendo  $a_1 \dots a_4$  coeficientes para alta y baja temperatura. Además, hay constantes de integración,  $a_5$  y  $a_6$ , para evaluar la entalpía  $h$  y entropía  $s$ .

- equationOfState: perfectGas, designa la ecuación de estado para gases ideales.

$$\rho = \frac{1}{R_G T} p$$

- specie: specie, define número de moles y peso molecular.
- energy: sensibleInternalEnergy, especifica que la ecuación de la energía interna será usada en la solución, y que el campo temperatura será calculado a partir de esta variable.

Asimismo, a la carpeta raíz constant se ha incorporado el directorio regionProperties (an. 11.49), clase simple que contiene información sobre las regiones acopladas, esto es, señala que regiones son sólidas y cuales fluidas. E initialConditions que ha sido actualizado incorporando las variables Tinitial y pinitial para la inicialización del caso.



### 6.3.3 Directorio “system”

El único aspecto importante a destacar respecto al directorio controlDict (an. 11.50) es que la aplicación chtMRFMultiRegionFoam ha sido seleccionada en lugar de simpleFoam, y que nuevas funciones de campos como  $e$  y  $p\_rgh$  han sido agregadas para el post-procesado de residuales iniciales.

Volviendo al dominio sólido, toca resaltar para el directorio fvSchemes (an. 11.51):

- No se necesitan esquemas de divergencia, pues no existe este término en el caso sólido.
- Para el término gradiente normal de superficie, el esquema uncorrected es aplicado. Este es equivalente al corrected, pero sin la corrección no ortogonal.
- Al subdirectorio que contiene los términos laplacianos le es superpuesto el esquema de Gauss con interpolación lineal sin corrección.

La región fluida (an. 11.52) mantiene la forma vista en el apartado 5.6.3, con la única adición al subdirectorio de términos divergencia de la advección de los campos: energía interna, entalpía, energía cinética y flujo de presión.

Respecto al algoritmo preferido para la discretización de la ecuación de la energía para la fase sólida, tal y como viene dado en el primer subdirectorio de fvSolution (an. 11.53), se halla el solver PCG (Preconditioned Conjugate Gradient) con preconditionador DIC (Diagonal Incomplete-Cholesky) orientado a matrices simétricas. Por otro lado, para el segundo subdirectorio, un factor de relajación de 0.7 ha sido impuesto a la ecuación de ambos campos  $e$  y  $h$ .

Prosiguiendo con los controles de cálculo lineales, para el medio fluido (an. 11.54) se ha optado esta vez por un suavizado DIC para el solver GAMG de presión. Mientras para las demás variables ( $U$ ,  $h$  y  $e$ ), el motor PBiCGStab (Stabilised Preconditioned Bi-Conjugate Gradient) con preconditionador DILU (Diagonal Incomplete-LU) destinado a matrices asimétricas ha sido escogido. A diferencia del dominio sólido, los factores de relajación de la entalpía y energía interna han sido atenuados hasta un valor de 0.3.

Muchas aplicaciones de OpenFOAM contienen sistemas de ecuaciones que se pueden manipular mediante opciones de volúmenes finitos establecidas por el usuario a través del directorio fvOptions, facilitando, por ejemplo, añadir el término fuente o hacer cumplir ciertas restricciones a las ecuaciones que rigen el sistema, sin la necesidad de reescribir el código fuente original.

La restricción fixedTemperatureConstraint en el archivo fvOptions (an. 11.55) fijará la temperatura en la ecuación de la energía para la región de malla correspondiente al devanado, independientemente de la temperatura del flujo entrante o del comportamiento del flujo local. Aguas abajo de la restricción, el campo de temperatura continuará desarrollándose normalmente.

## 6.4 Resultados caso chtMultiRegionFoam

Fijado a 5000 el número de iteraciones, se consigue llevar los residuales iniciales por debajo de una tolerancia de  $10^{-4}$  para todos los campos fluidos, así como para la energía interna de los sólidos internos del motor. Por otro lado, los sólidos externos (aquellos a los que la condición de pared externa fue asignada) satisfacen la convergencia para un orden menor de magnitud ( $10^{-3}$ ).

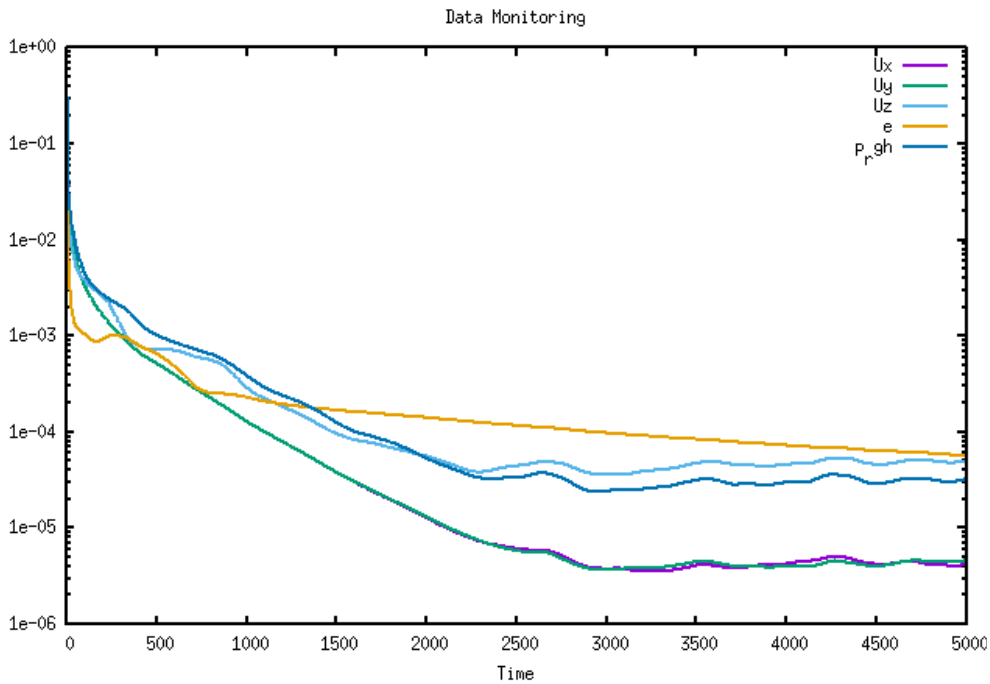


Figura 36: Evolución de los residuales del dominio fluido

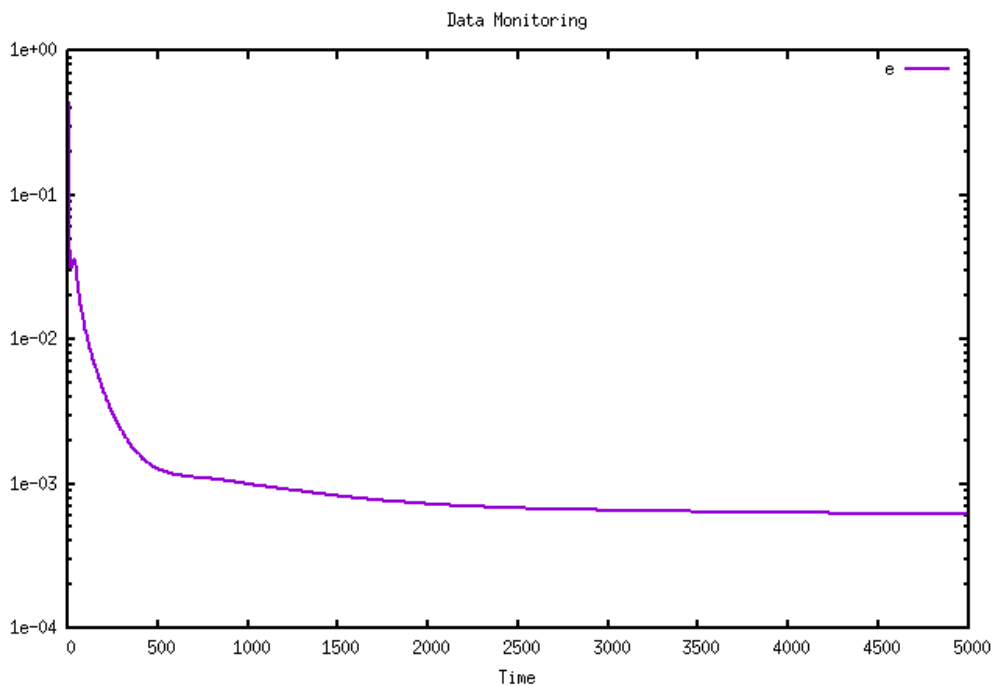


Figura 37: Evolución de los residuales del cubremotor

En este nuevo contexto multifísico, el perfil de velocidades del aire sigue mostrando un comportamiento bastante similar al evaluado durante el apartado 5.7 para el contorno de velocidades correspondiente a  $\omega=15$  rad, motivado por su proximidad en velocidad angular con respecto a este.

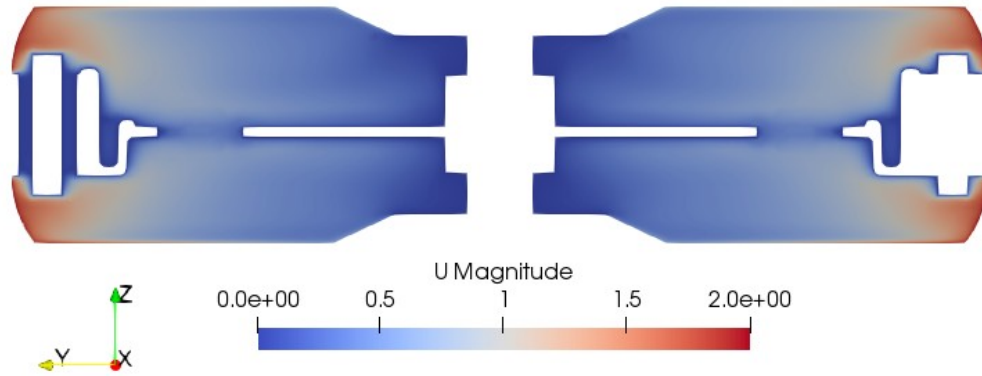


Figura 38: Sección YZ del campo velocidad del fluido,  $\omega=16$  rad/s

Tal y como se deduce de la figura 39, la interfaz de contacto del fluido interno con el devanado se encuentra a la temperatura impuesta de 335 K, mientras que, la superficie de los sólidos en contacto con el aire exterior se encuentra a temperatura ambiente. Además, se percibe que las líneas de corriente más alejadas al eje se encuentran a una menor temperatura que las más próximas, a pesar de encontrarse más cerca del componente generador del calor.

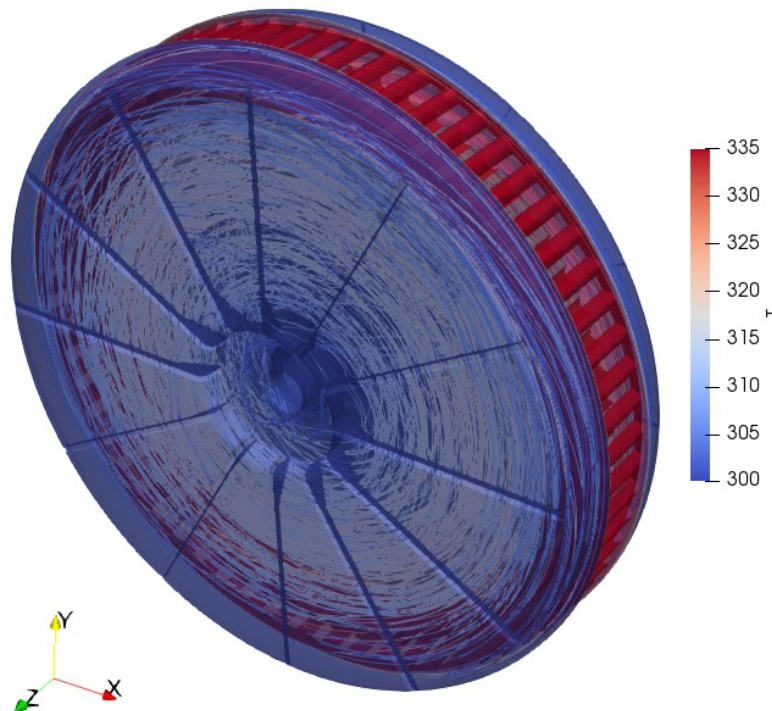


Figura 39: Distribución de la temperatura del fluido con líneas de corriente

De la distribución de temperatura de la figura de abajo, podemos denotar que el núcleo de hierro alcanza una temperatura uniforme de 335 K conforme al fenómeno de conducción con la bobina, y solamente la región más cercana al devanado del estátor se homogeneiza con esa temperatura. Por otro lado, se puede advertir que el aire, en cierto modo, se comporta como un aislante no transfiriendo adecuadamente la energía calorífica desde el estátor al rotor, y provocando con ello su propio calentamiento. El incremento insignificante de la temperatura del cubremotor y demás componentes externos en comparación al fluido es prueba de ello.

Otro de los motivos que incentiva una ineficiente expulsión del calor es el comportamiento laminar del aire, pues como se observa la temperatura del flujo es inferior en las zonas más alejadas al eje (donde la velocidad es mayor), aun estando más cerca al foco de calor, y superior en la zona central donde la turbulencia del aire es inapreciable.

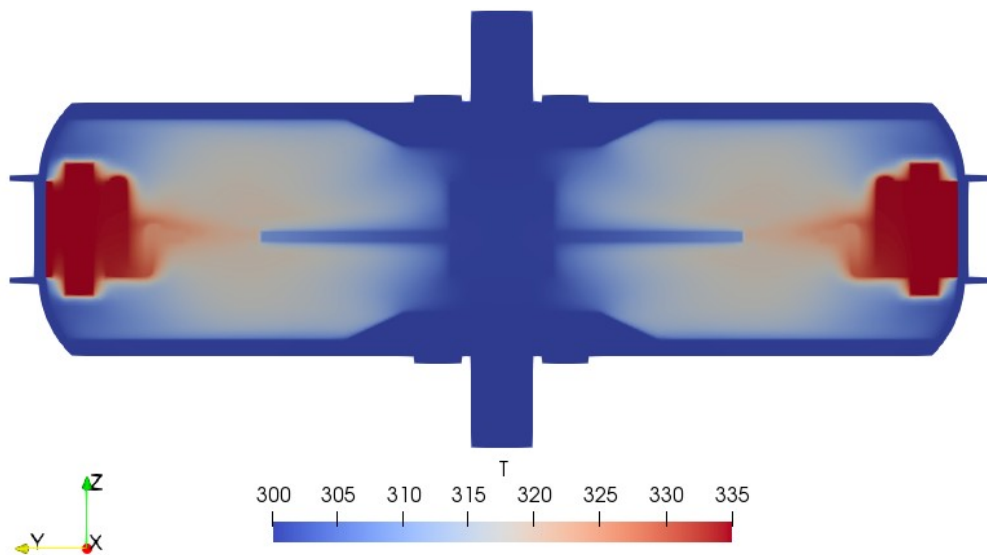


Figura 40: Sección YZ de la distribución de temperatura del motor

Adicionalmente, se aprecia como la condición adiabática dada entre los imanes y el núcleo de la bobina trabaja fielmente. Condición razonablemente realista, ya que, al pequeño gap de aire existente entre el estátor y el rotor que impide la conducción directa, se suma que la conductividad térmica del NdFeB es irrisoria ( $9 \text{ W/m}\cdot\text{K}$ ).

Siguiendo el patrón del campo de temperaturas, tenemos la densidad y presión del fluido de acuerdo a la ecuación de los gases ideales. Cumpliéndose en el caso de la densidad, por efecto de Coriolis, que las masas de mayor densidad (menor temperatura), al estar dotadas de una mayor fuerza radial, se acumulan en las regiones más próximas y más lejanas al eje. La presión, por su parte, varía muy tenuemente, y el cambio principalmente está ligado a la presión dinámica, y en menor medida al fenómeno térmico.

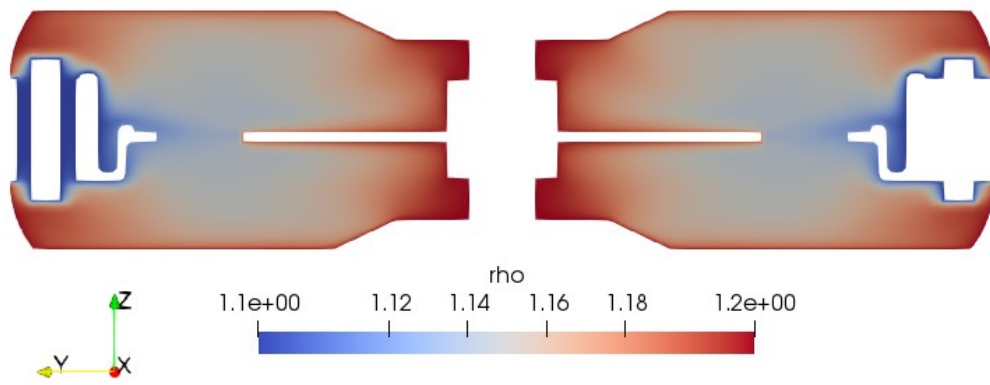


Figura 41: Sección YZ del contorno densidad

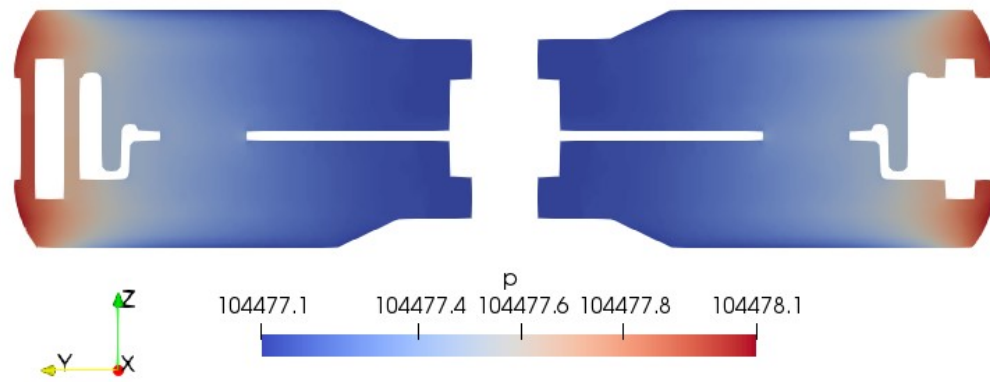


Figura 42: Sección YZ del contorno presión

## 7. Refrigeración líquida directa del motor

### 7.1 Introducción

El fenómeno de transferencia de calor por convección sólido-fluido es el principal camino de refrigeración en máquinas eléctricas. ¿Pero cómo podemos mejorar esto? Analicemos la ley de enfriamiento de Newton que determina la transferencia de calor por convección entre un sólido y un fluido:

$$\Phi = h_c A (T_s - T_f)$$

donde  $\Phi$  son las pérdidas en forma de calor por unidad de tiempo,  $h_c$  el coeficiente de convección,  $A$  la superficie de intercambio y  $T_s$  la temperatura en la superficie del sólido y  $T_f$  la temperatura del fluido lejos del cuerpo.

Parece lógico que si debemos mantener la geometría (superficie de intercambio) y la eficiencia termoeléctrica (temperatura de la bobina) la única variable de diseño factible de modificación sería el coeficiente de transferencia de calor  $h_c$ . Para lo cual, existen dos vías que harían posible un incremento del coeficiente de película:

1. Refrigeración del motor usando un fluido de diferente naturaleza al aire, que disponga de un mayor coeficiente de convección.
2. Incrementar la convección forzada, ya sea, mediante una velocidad de rotación mayor o provocando a través de las aletas un comportamiento más turbulento del flujo.

Respecto a la segunda alternativa, el aumento de la velocidad angular es inviable puesto que viene determinada por el usuario, la potencia del motor y la morfología del terreno. Por otro lado, incentivar a través del diseño de las aletas el desarrollo de un flujo más turbulento, implicaría cambios geométricos en el cubremotor.

De modo que nos centraremos en la primera opción, la cual no requiere de ningún tipo de trabajo de rediseño, permitiendo ser más fácilmente adoptada por el constructor o fabricante que se encarga del desarrollo del motor eléctrico.

Más concretamente, nos estamos refiriendo al uso de un líquido dotado de una buena conductividad térmica, pero que a su vez impida la conductividad eléctrica (aislante), en definitiva, nos estamos refiriendo a un fluido dieléctrico.

Además, una de las ventajas sobresalientes de este tipo de refrigeración es que goza de un carácter pasivo, al beneficiarse del mecanismo de rotación de la rueda para estar en constante movimiento y llegar a todos los rincones del interior del motor, transfiriendo, de esta forma, el calor generado en el bobinado hacia el exterior del rotor.

## 7.2 Líquidos dieléctricos

En la actualidad existen principalmente cuatro grandes grupos de fluidos refrigerantes de uso industrial: a) aceites minerales, b) aceites de silicona, c) ésteres sintéticos y d) ésteres naturales; merecedores de nuestra atención por las propiedades físicas y térmicas que nos brindan.

Estos líquidos son ampliamente utilizados como fluidos aislantes y de refrigeración en motores, transformadores, disyuntores, celdas, etc, y sus características más destacadas son las siguientes:

- Enfriamiento de componentes: transfieren al exterior el calor generado por la resistencia de los conductores metálicos y devanados.
- Propiedades aislantes: los dieléctricos también tienen la función de aislar los componentes metálicos.
- Durabilidad: cuentan con excelentes propiedades antioxidantes, pues deben mantener sus propiedades durante largos períodos de tiempo.
- Alta resistencia dieléctrica: capacidad del fluido para resistir el voltaje entre los devanados, evitando la formación de arcos y descargas eléctricas.
- Viscosidad: la baja viscosidad permite la circulación del líquido y, en consecuencia, mejora la transferencia de calor.
- Tensión interfacial: una medición indirecta de la cantidad de compuestos polares presentes en el fluido. Es un indicador de la presencia de ácidos polares resultantes de la degradación del líquido. La presencia de contaminantes polares cambia la dirección del campo eléctrico, provocando pérdidas de energía.

El aceite mineral ha sido utilizado como fluido aislante en equipos eléctricos durante más de un siglo debido a sus propiedades dieléctricas, físicas, térmicas, y sobre todo por su bajo coste. Sin embargo, existen nuevos requerimientos ambientales que posicionan al aceite mineral y de silicona en clara desventaja debido a su resistencia a la biodegradación y por su presencia más nociva hacia el medioambiente. Los ésteres, por otra parte, además de contar con un aislamiento adecuado, con propiedades químicas y térmicas para su uso en componentes eléctricos, también muestran notables ventajas como son su alto punto de inflamabilidad, bajo impacto medioambiental, ausencia de productos tóxicos y facilidad de reciclado.

Aunque todos estos líquidos presentan propiedades físico-químicas que podrían ajustarse a la perfección a nuestra aplicación, nos decantaremos por el éster natural, ya que además de cumplir con nuestras especificaciones, resulta ser el más respetuoso con el medio ambiente. Estos dieléctricos vegetales se obtienen naturalmente a partir de cultivos de granos y plantas (soja, girasol, colza, etc) pudiendo llegar a ser totalmente recuperables y reciclables.

### 7.3 Propiedades termofísicas

Tal y como ya apuntamos anteriormente en el apartado 6.3.2, existen diversos modelos para definir las propiedades termofísicas de un fluido. Lo ideal, es emplear el que mejor se adapte a la caracterización del medio en cuestión. Las propiedades a conocer son: la densidad, la capacidad calorífica, la conductividad térmica y la viscosidad dinámica; parámetros que de acuerdo a un éster natural típico [39] varían con la temperatura de forma polinómica.

Los datos se utilizan en combinación con la hoja de cálculo Calc para realizar un ajuste polinómico. Para la densidad y la conductividad térmica se usa un polinomio de primer orden:  $f(T)=AT+B$ . El calor específico se ajusta con una polinomio cuadrático:  $f(T)=AT^2+BT+C$ . Y para el caso de la viscosidad un polinomio de tercer grado es fijado:  $f(T)=AT^3+BT^2+CT+D$ .

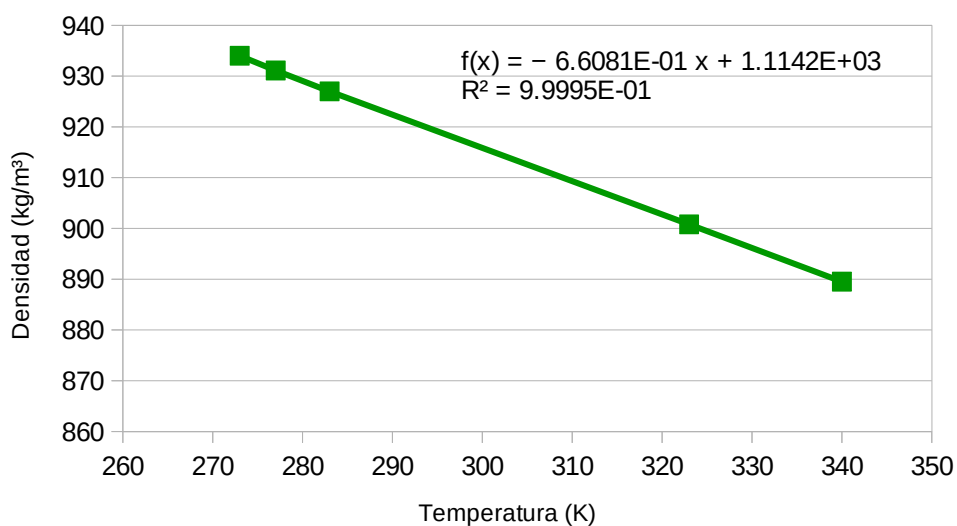


Figura 43: Densidad éster natural y ajuste lineal

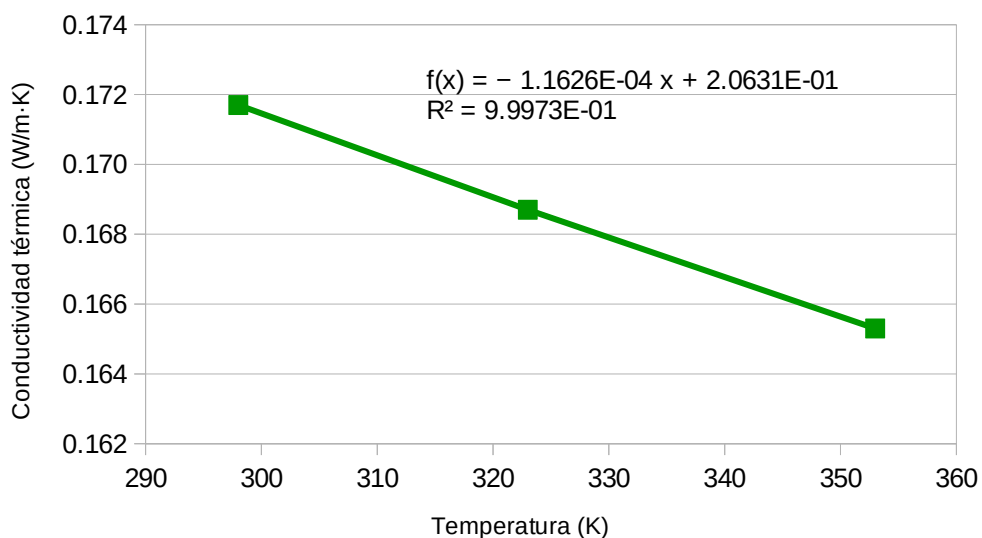


Figura 44: Conductividad térmica éster natural y ajuste lineal



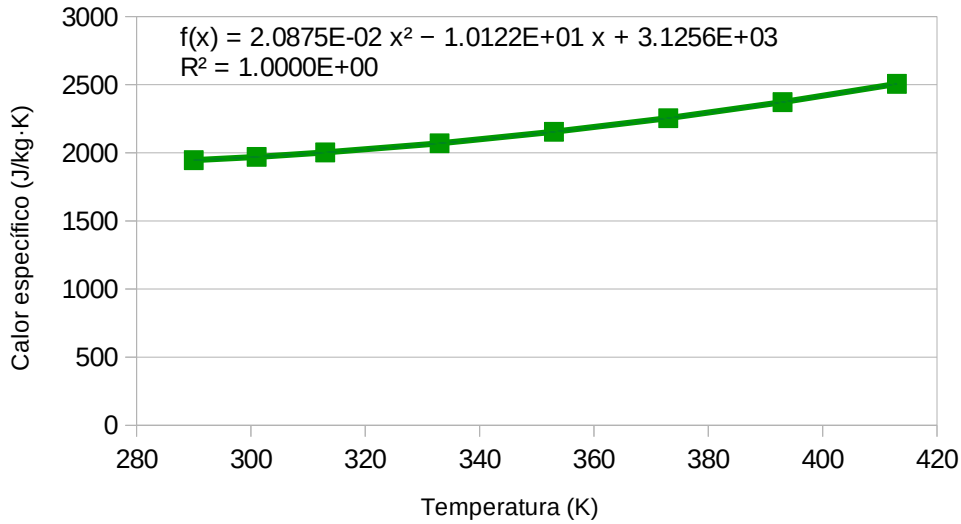


Figura 45: Calor específico éster natural y ajuste cuadrático

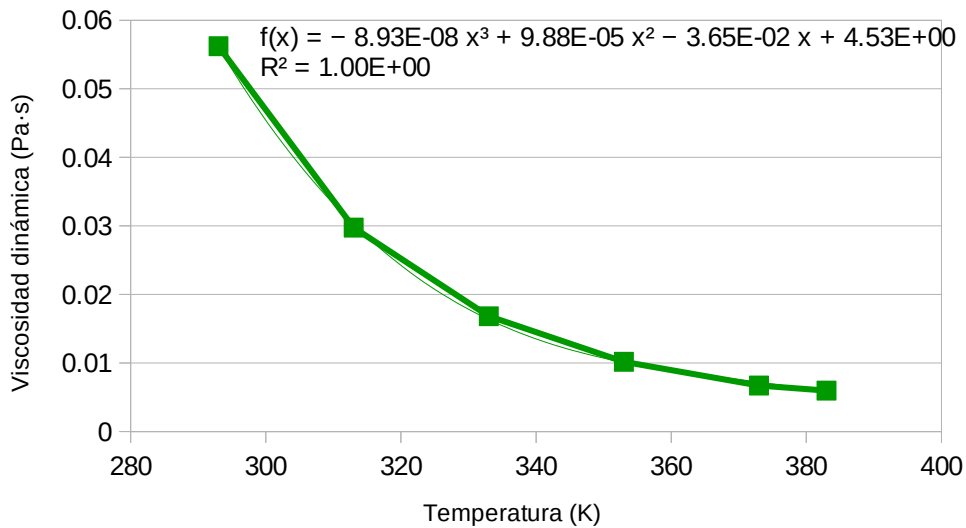


Figura 46: Viscosidad cinemática éster natural y ajuste cúbico

OpenFOAM, ofrece la posibilidad de un enfoque polinomial para cada una de estas variables, conforme al directorio thermophysicalProperties (an. 11.55):

- transport: calcula la viscosidad dinámica  $\mu$  y la conductividad térmica  $\kappa$  como una función de la temperatura  $T$  de un polinomio de orden  $N$ .

$$\mu = \sum_{i=0}^{N-1} a_i T^i$$

- hPolynomial: determina  $c_p$  en función de la temperatura mediante un polinomio de cualquier orden  $N$ .

$$c_p = \sum_{i=0}^{N-1} a_i T^i$$

- icoPolynomial: ecuación de estado polinomial e incompresible.

$$\rho = \sum_{i=0}^{N-1} a_i T^i$$

donde  $a_i$  son coeficientes polinomiales de cualquier orden N.

## 7.4 Resultados

Con una mayor viscosidad y por consiguiente un menor Re y turbulencia que el aire, este fluido dieléctrico alcanza más rápidamente una tolerancia de convergencia de  $10^{-4}$  para el dominio fluido. Sin embargo, como el fenómeno de conducción es más intenso en este estudio, se requiere de un mayor número de iteraciones para lograr emplazar el residual inicial de las regiones sólidas por debajo de una tolerancia de  $10^{-3}$ .

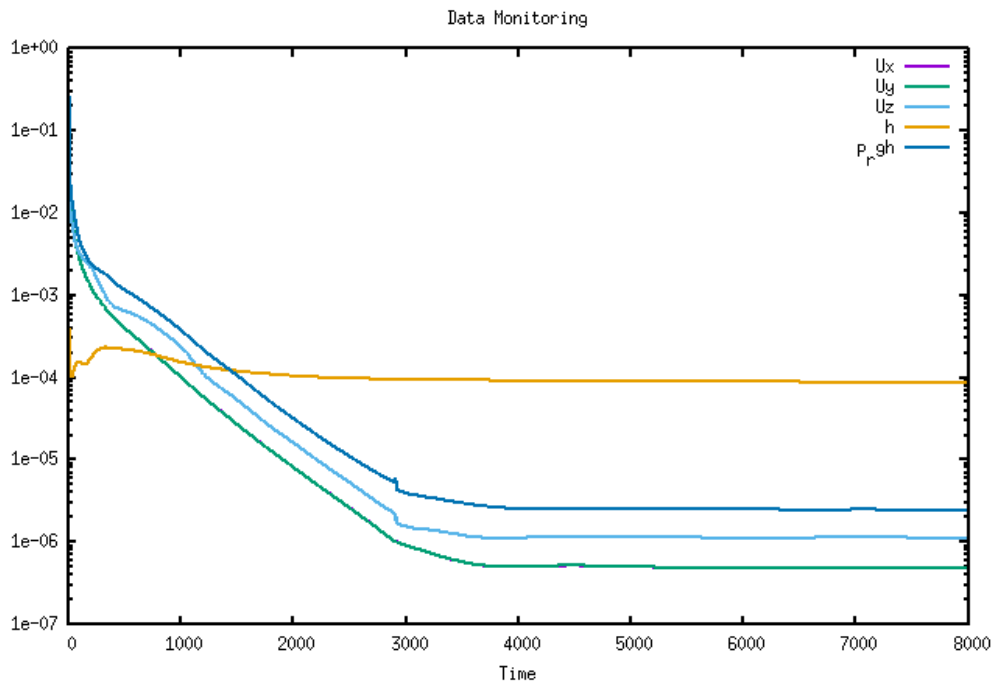


Figura 47: Evolución de los residuales del dominio fluido

El campo de velocidades continúa con la tendencia marcada por el aire, sin darse diferencias significativas más que una cierta propensión a formar un mayor espesor de capa límite en el caso del líquido dieléctrico, motivada por su superior viscosidad. Pues las fuerzas de inercia y las debidas a los esfuerzos viscosos por cortante en la pared comienzan a ser del mismo orden.

Como era de esperar, en términos absolutos, la variación de la densidad (figura 50) para el dominio fluido es bastante reducida debido a que el líquido no experimenta una distribución térmica suficientemente heterogénea, a excepción de algunas regiones afines al devanado.

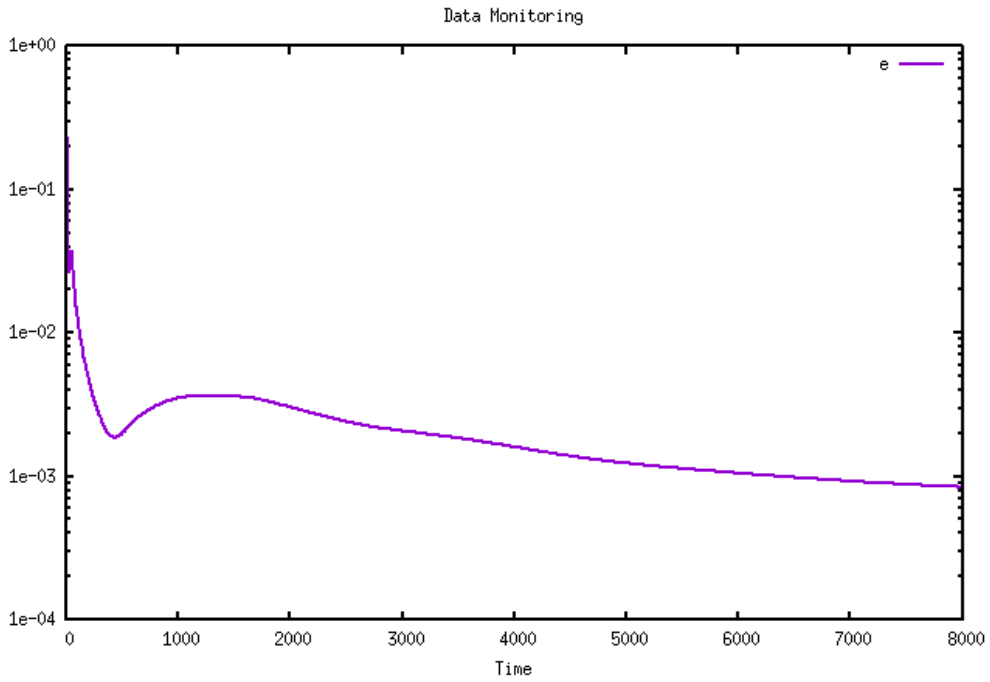


Figura 48: Evolución de los residuales para el cubremotor

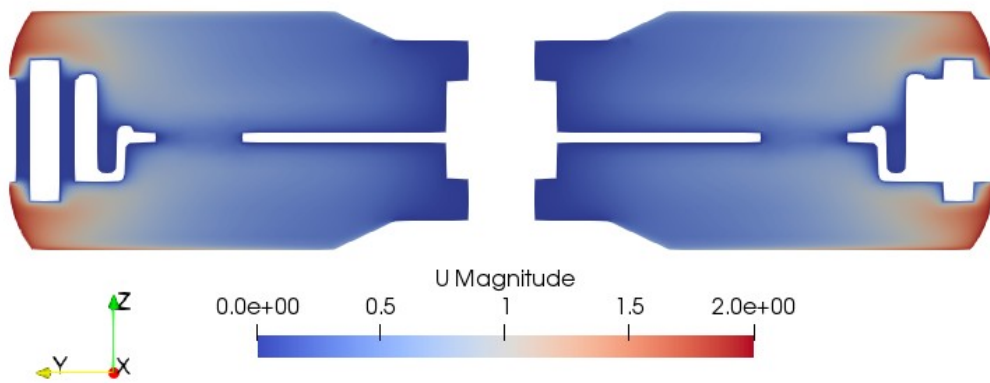


Figura 49: Sección YZ del campo de velocidad para el éster natural

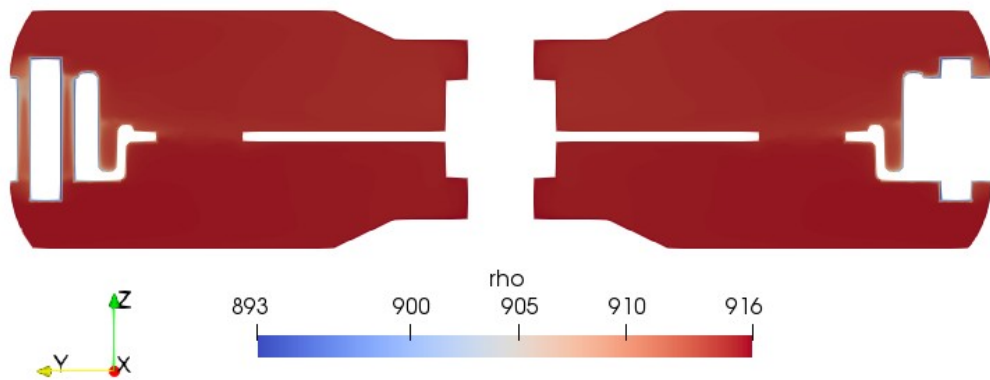


Figura 50: Sección YZ para el contorno densidad del éster natural

Muy diferente a la distribución térmica del apartado 6.5, es el contorno de temperatura que experimenta el éster natural, donde se contempla que el fluido se mantiene a una temperatura más baja y uniforme, permitiendo a su vez, un menor calentamiento de los elementos aledaños al devanado. Lo cual viene motivado por una eficiente conducción del calor del líquido refrigerante hacia el exterior del motor.

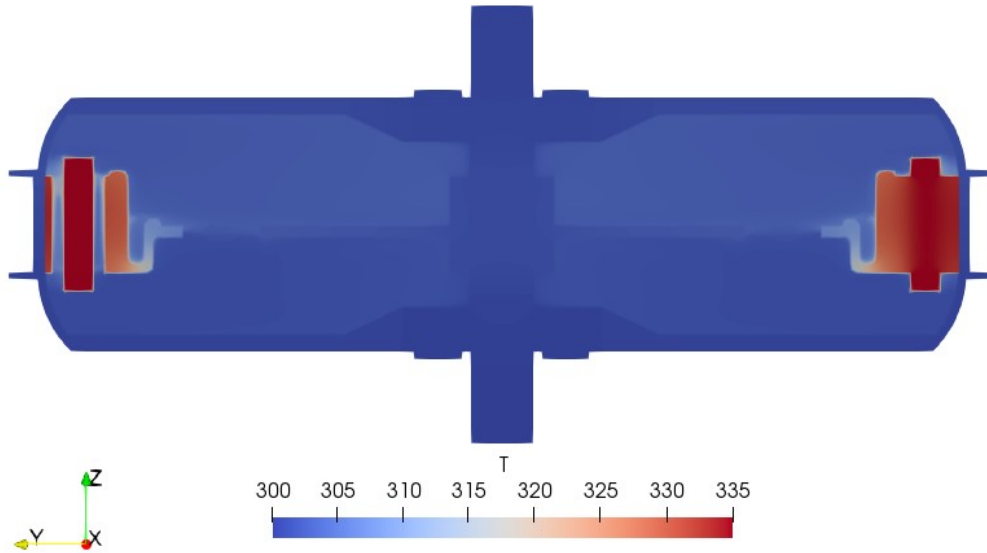


Figura 51: Sección YZ de la distribución de temperatura del motor

Con el objetivo de llevar a cabo una comparativa entre la diferencia que supondría el uso de un gas como el aire o un líquido dieléctrico como es el éster natural para la refrigeración del motor eléctrico, calcularemos el flujo de calor que se vierte al exterior a través del cubremotor, en cada uno de los dos casos:

Refrigerante	Flujo de calor (W/m <sup>2</sup> )
Aire	0.31
Éster natural	1.95

Tabla 4: Flujo de calor expulsado al exterior a través del cubremotor

Luego, de la tabla se puede deducir que el uso de un líquido dieléctrico supondría una eficiencia 6 veces mayor que el aire. Además, no podemos olvidar que una mejor refrigeración significa menor pérdida de rendimiento y vida útil prolongada.

## 8. Conclusiones

Recientemente, con la creciente preocupación por los graves problemas ambientales, la UE ha comenzado a apostar fuertemente por las estrategias y políticas en el ámbito de la movilidad urbana (en particular en lo que respecta a la eficiencia energética y al cambio climático). Una vía importante para abordar los retos relacionados con las áreas urbanas, es el de los Planes de Movilidad Urbana Sostenible (PMUS), conjunto de actuaciones que tienen como objetivo implantar formas de desplazamiento más sostenibles en el espacio urbano reduciendo el consumo energético y las emisiones contaminantes, logrando al mismo tiempo garantizar la calidad de vida de la ciudadanía.

Por otro lado, los motores de corriente continua sin escobillas (BLDC) se han vuelto muy comunes y se utilizan ampliamente en diversas áreas: electrodomésticos, vehículos eléctricos, dispositivos aeroespaciales y médicos, equipos de automatización industrial, etc. Los motores BLDC sin escobillas y conmutados electrónicamente, muestran una vida útil mejorada y una alta eficiencia porque al eliminar las escobillas y los conmutadores se reduce la fricción mecánica. A lo que hay que añadir que, los motores BLDC no producen ruido eléctrico o mecánico y se pueden fabricar en un tamaño compacto.

Los avances tecnológicos experimentados durante estos últimos años junto con la necesidad de impulsar una movilidad más sostenible, ecológica y limpia ha propiciado la eclosión de vehículos de movilidad personal (VMP), en muchos casos eléctricos, que pueden ser parte de una solución sostenible al problema que plantea la movilidad de primera y última milla en las ciudades y que facilitan el uso multimodal combinado con el transporte público en sustitución total o parcial del uso del coche.

La actual producción de motores eléctricos de alto rendimiento y densidad de potencia, en el ámbito de la micromovilidad, supone también un aumento de la generación de calor. Lo cual resulta de vital importancia vigilar, pues, una disipación deficiente de calor puede causar un aumento excesivo de la temperatura en el interior del motor, causando una desmagnetización de los imanes permanentes y un aumento de la resistencia en la bobina, y por lo tanto una disminución en el rendimiento del motor. Además, la tasa de fallos de un dispositivo eléctrico está relacionada con la temperatura de funcionamiento, de forma que cuando aumenta esta disminuye su vida útil.

Por lo tanto, la refrigeración efectiva de los devanados del estátor y de los imanes del rotor contribuye directamente a la vida, la producción de potencia y la fiabilidad de cualquier motor eléctrico. Esto ha provocado avances significativos en la refrigeración por aire y líquidos para poder gestionar mayores flujos de calor. Actualmente este es un aspecto primordial en el mercado de desarrollo de nuevos motores junto a los mejores rendimientos, el menor consumo de energía y la reducción de tamaño del propio motor.

Con este pretexto se ha analizado el comportamiento termofísico del aire en el interior del motor eléctrico de una rueda monociclo de alta potencia. Para comprobar que su régimen laminar a alta temperatura debería ser alterado (e.

g. a través de modificaciones geométricas de las aletas del cubremotor) con objeto de conseguir un movimiento transitorio del fluido de mayor eficiencia térmica, y/o el medio refrigerante sustituido por un líquido dieléctrico (de mayor coeficiente de convección). Finalmente, el carácter más generalista y de fácil aplicación constructiva de esta segunda opción hizo que nos decantáramos por la misma.

La imposibilidad de utilizar una malla de tipo móvil y llevar a cabo un cálculo transitorio definiendo cada uno de los datos de velocidad angular y temperatura obtenidos durante la fase de prueba real del monociclo, debido al coste computacional que esto supone, nos obliga a analizar velocidades discretas de rotación y temperatura. Respecto a la velocidad angular, 4 casos (15, 20, 25 y 30 rad/s), demostrados de mayor frecuencia durante el ensayo, fueron estudiados. Mientras, en el caso de la temperatura, se tomó como valor representativo la máxima temperatura alcanzada por el sistema y su velocidad de giro asociada.

Analizado el comportamiento del fluido correspondiente a cada una de estas velocidades mediante un modelado MRF del dominio fluido aire y su posterior estudio multifísico del motor completo con  $\omega_m=16$  rad/s y  $T_{max}=62$  °C, se concluye que sería conveniente fomentar el desarrollo de un flujo más turbulento mediante cambios geométricos, y sustituir el gas refrigerante por un líquido dieléctrico de mayor coeficiente de película.

El uso de un éster natural, como dieléctrico de mayor biodegradabilidad, permite incrementar en más de un 600% (en comparación con el aire) la transferencia de calor desde el estátor al ambiente, consiguiendo una menor temperatura interna general en el motor, y por consiguiente, una menor pérdida de torque por desmagnetización de los imanes y/o aumento de la resistencia de la bobina.

Comprobado el régimen mayoritariamente laminar a la velocidad angular de  $\omega_m=16$  rad/s, adicionalmente al uso de un líquido dieléctrico, habría sido de interés haber puesto en práctica otras recomendaciones que contribuirían positivamente en la transferencia de calor hacia el exterior, como son:

1. Un incremento en la profundidad de las aletas, pues, como hemos analizado en el capítulo 5, los principales causantes del desarrollo de flujo turbulento son las aletas. Y, por otro lado, cuanto mayor sea la turbulencia de un flujo, mayor será su eficiencia en la transferencia calor.
2. Trocear el núcleo de hierro en varias láminas estrechas para reducir las corrientes parásitas y por ende las pérdidas por corrientes de Foucault. Las pilas de láminas delgadas están recubiertas con una capa aislante, que se encuentran lo más paralelas posible a las líneas de flujo. Las capas de aislamiento sirven como una barrera para las corrientes parásitas, por lo que la corriente solo puede fluir en bucles estrechos dentro del espesor de cada laminación individual. Dado que la corriente en un bucle de corrientes parásitas es proporcional al área del bucle,

esto evita que fluya la mayor parte de la corriente, lo que reduce las pérdidas a un nivel muy pequeño.

3. La apertura de ciertas ranuras estéticas en la carcasa del monociclo con el objetivo de aumentar el coeficiente de transferencia de calor del motor con el aire exterior, pasando de esta forma de la convección libre actual a una convección forzada.

Además, de los puntos anteriormente expuestos, podría ser de interés para desarrollos futuros, el considerar un refrigerante compuesto por una mezcla multifase líquido/gas. De esta forma, la eficiencia conseguida podría ser aún mayor, al no sacrificar parte del par motor en mover un líquido de mayor densidad.

## 9. Glosario

### 9.1 Acrónimos

AIE: Agencia Internacional de Energía

APP: APPlication

ASCII: American Standard Code for Information Interchange

BLDC: BrushLess DC electric

CAD: Computer-Aided Design

CATIA: Computer-Aided Three Dimensional Interactive Application

CEF: Connecting Europe Facility

CFD: Computational Fluid Dynamics

CO<sub>2</sub>: Dióxido de Carbono

COVID: COronaVirus Disease

CPU: Central Processing Unit

DIC: Diagonal Incomplete-Cholesky

DILU: Diagonal Incomplete-LU

EUC: Electric UniCycle

FEDER: Fondo Europeo de DEsarrollo Regional

GAMG: Geometric-Algebraic Multi-Grid

HPC: High Performance Computing

H2020: Horizonte 2020

INB: Ingreso Nacional Bruto

KS: KingSong

LIFE: L'Instrument Financier pour l'Environnement

MRF: Multiple Reference Frame

NdFeB: Neodymium magnet

NPD Group: National Purchase Diary Panel

ONU: Organización de las Naciones Unidas

OpenFOAM: Open Field Operation and Manipulation

PMUS: Planes de Movilidad Urbana Sostenible

PBiCGStab: Stabilised Preconditioned Bi-Conjugate Gradient

PCG: Preconditioned Conjugate Gradient

PMV: Personal Mobility Vehicle

RAS: Reynolds-Average Simulation

RDT: Rapid Distortion Theory



RTE-T: Red TransEuropea de Transporte  
SARS: Severe Acute Respiratory Syndrome  
SIMPLE: Semi-Implicit Method for Pressure Linked Equations  
STL: STereoLithography  
UE: Unión Europea  
U-MOB: Red Europea de Movilidad Sostenible en la Universidad  
VMP: Vehículos de Movilidad Personal  
2D: Bi-Dimensional  
3D: Tri-Dimensional

## 9.2 Nomenclatura

A: superficie de intercambio  
 $A_s$ : coeficiente de Sutherland  
 $a_i$ : coeficientes polinomio de grado N  
 $a_1, \dots, a_4$ : coeficientes modelo termodinámico janaf  
 $a_5$  y  $a_6$ : constantes de integración modelo termodinámico janaf  
 $C_1, C_2, C_3$ : constantes del modelo de turbulencia k- $\epsilon$   
 $C_\mu$ : coeficiente para la viscosidad turbulenta  
 $c_p$ : calor específico a presión constante  
 $c_v$ : calor específico a volumen constante  
 $D_k$ : difusividad efectiva para k  
 $D_\epsilon$ : difusividad efectiva para  $\epsilon$   
e: energía interna específica  
 $e_s$ : espesor del sólido  
**f**: término fuente  
**g**: vector gravedad  
 $H_f$ : calor de fusión  
h: entalpía  
 $h_a$ : coeficiente de convección libre con el aire ambiente  
 $h_c$ : coeficiente de convección  
K: energía cinética específica  
k: energía cinética turbulenta  
**n**: vector unitario  
P: punto arbitrario perteneciente al dominio fluido  
p: presión  
 $P_k$ : tasa de producción de energía cinética turbulenta

$p_k$ : presión cinemática  
 $q$ : generación de energía por unidad de volumen  
 $\mathbf{q}$ : vector flujo de calor  
 $R$ : radio de la rueda  
 $Re$ : Número de Reynolds  
 $R_G$ : constante de los gases ideales  
 $R_{th}$ : resistencia térmica de transferencia de calor  
 $r$ : fuente de calor específica  
 $\mathbf{r}$ : vector de posición sistema relativo no inercial  
 $r_m$ : radio medio dominio fluido  
 $r_{max}$ : radio exterior de las aletas  
 $\mathbf{r}_0$ : vector de posición sistema absoluto inercial  
 $s$ : entropía  
 $T$ : temperatura  
 $T_a$ : temperatura ambiente  
 $T_b$ : temperatura de la cara interior del sólido  
 $T_c$ : temperatura común modelo janaf  
 $T_f$ : temperatura del fluido  
 $T_h$ : límite superior modelo janaf  
 $T^i$ : grado variable temperatura de un polinomio de orden  $N$   
 $T_l$ : límite inferior modelo janaf  
 $T_{max}$ : temperatura máxima del sistema  
 $T_S$ : temperatura de Sutherland  
 $T_s$ : temperatura en la superficie del sólido  
 $\mathbf{u}$ : vector velocidad  
 $\mathbf{u}_I$ : vector velocidad en el marco de referencia absoluto inercial  
 $\mathbf{u}_r$ : velocidad relativa (velocidad vista desde el marco en movimiento)  
 $\mathbf{u}_T$ : vector velocidad del marco en traslación  
 $v$ : velocidad lineal  
 $w_{max}$ : máxima anchura de las aletas  
 $w_{min}$ : longitud característica  
 $\alpha$ : difusividad térmica  
 $\alpha_R$ : factor de relajación  
 $\varepsilon$ : tasa de disipación de energía cinética turbulenta  
 $\kappa$ : conductividad térmica  
 $\mu$ : viscosidad dinámica

$\rho$ : densidad

$\zeta$ : vorticidad

$\sigma$ : constante de Stefan-Boltzmann

$\nu$ : viscosidad cinemática

$\nu_t$ : viscosidad turbulenta

$\Phi$ : pérdidas en forma de calor por unidad de tiempo

$\Phi_b$ : flujo de calor hacia el ambiente

$\psi$ : coeficiente para la estabilización del término gradiente normal

$\Omega$ : eje de rotación sistema no inercial

$\omega$ : velocidad angular

$\omega_m$ : velocidad angular media de datos con temperatura igual a la  $T_{max}$

## 10. Bibliografía

- [1]-<https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/micromobilitys-15000-mile-checkup> [16/01/2020]
- [2]-<https://www.seat.es/sobre-seat/noticias/corporativas/micromobilidad.html> [16/01/2020]
- [3]-[https://www.electricunicycles.eu/product-ci170-kingsong\\_ks\\_18xl#tab\\_specification](https://www.electricunicycles.eu/product-ci170-kingsong_ks_18xl#tab_specification) [18/01/2020]
- [4]-<https://micromobility.io/blog/2019/2/23/the-micromobility-definition> [19/01/2020]
- [5]-[https://www.itdp.org/wp-content/uploads/2019/12/ITDP\\_The-Electric-Assist\\_-\\_Leveraging-E-bikes-and-E-scooters-for-More-Livable-Cities.pdf](https://www.itdp.org/wp-content/uploads/2019/12/ITDP_The-Electric-Assist_-_Leveraging-E-bikes-and-E-scooters-for-More-Livable-Cities.pdf) [19/01/2020]
- [6]-<https://medium.com/datadriveninvestor/micromobility-is-the-future-of-vehicles-220c2c0c9b0> [26/01/2020]
- [7]-<https://www.wri.org/blog/2018/10/8-things-you-need-know-about-ipcc-15-c-report> [01/02/2020]
- [8]-<https://www.nasa.gov/press-release/nasa-releases-detailed-global-climate-change-projections> [01/02/2020]
- [9]-<https://www.fastcompany.com/3061930/transportation-is-now-the-fastest-growing-cause-of-greenhouse-gas-emissions> [01/02/2020]
- [10]-<https://u-mob.eu/es/proyecto/> [15/02/2020]
- [11]-<http://eshorizonte2020.cdti.es/index.asp?MP=87&MS=719&MN=2> [15/02/2020]
- [12]-[http://www.albea-transenergy.com/docs/cef\\_transport\\_es.pdf](http://www.albea-transenergy.com/docs/cef_transport_es.pdf) [15/02/2020]
- [13]-[https://ec.europa.eu/regional\\_policy/es/policy/themes/transport-energy/](https://ec.europa.eu/regional_policy/es/policy/themes/transport-energy/) [15/02/2020]
- [14]-<https://www.iea.org/reports/global-ev-outlook-2019> [16/02/2020]
- [15]-<https://micromobility.io/blog/2019/3/20/the-five-categories-of-micromobility> [07/03/2020]
- [16]-<https://pbs.twimg.com/media/D12fRvIUwAAyWWh?format=png&name=900x900> [08/03/2020]
- [17]-<https://micromobility.io/blog/2019/3/12/the-automobility-micromobility-dichotomy> [08/03/2020]

- [18]-[World Economic and Social Survey 2013: Sustainable Development Challenges](#)
- [19]-[World Urbanization Prospects: The 2018 Revision](#)
- [20]-<https://www.nytimes.com/2020/05/18/nyregion/bike-shortage-coronavirus.html> [23/05/2020]
- [21]-<https://www.weforum.org/agenda/2020/01/smart-and-the-city-working-title/> [23/05/2020]
- [22]-<https://www.itdp.org/2020/04/08/the-air-quality-crisis-has-made-the-coronavirus-more-deadly/> [23/05/2020]
- [23]-<https://www.europapress.es/economia/transportes-00343/noticia-plataformas-micromovilidad-defienden-ser-alternativa-uso-masivo-coche-privado-desescalada-20200505174305.html> [24/05/2020]
- [24]-[https://www.metropoliabierta.com/informacion-municipal/movilidad/micromovilidad-clave-ciudades-futuro\\_30786\\_102.html](https://www.metropoliabierta.com/informacion-municipal/movilidad/micromovilidad-clave-ciudades-futuro_30786_102.html) [24/05/2020]
- [25]-<https://github.com/slastowski/WheelLogAndroid> [27/10/2019]
- [26]-<http://wheellogviewer.net/> [28/10/2019]
- [27]-<https://plot.ly/create/> [30/10/2019]
- [28]-[https://en.wikipedia.org/wiki/Reynolds\\_number](https://en.wikipedia.org/wiki/Reynolds_number) [15/02/2020]
- [29]- Caretto L.S., Gosman A.D., Patankar S.V., Spalding D.B. Two calculation procedures for steady, three-dimensional flows with recirculation. In: Cabannes H., Temam R. Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics. Lecture Notes in Physics, 19, 60-8. Springer, Berlin, Heidelberg (1973).
- [30]-[https://openfoamwiki.net/index.php/OpenFOAM\\_guide/The\\_SIMPLE\\_algorithm\\_in\\_OpenFOAM](https://openfoamwiki.net/index.php/OpenFOAM_guide/The_SIMPLE_algorithm_in_OpenFOAM) [11/04/2020]
- [31]-[https://www.cfd-online.com/Wiki/SIMPLE\\_algorithm](https://www.cfd-online.com/Wiki/SIMPLE_algorithm) [12/04/2020]
- [32]-Launder B.E., Spalding D.B. The numerical computation of turbulent flows. Computer method in applied mechanics and engineering, 3, 269-89 (1974).
- [33]-El Tahry S.  $k-\epsilon$  equation for compressible reciprocating engine flows. Journal of Energy, 7, 345-53 (1983).
- [34]-<https://cfd.direct/openfoam/user-guide/> [25/04/2020]

[35]-Wang B., Li J., Ai D. Temperature Research of Permanent Magnet Brushless DC Motor for Electric Balanced Vehicle. World Journal of Engineering and Technology, 7, 1-9 (2019).

[36]-Lungoci C., Stoia D. Temperature effects on torque production and efficiency of motors with NdFeB. Revue Roumaine des Sciences Techniques - Serie Électrotechnique et Énergétique, 53, 445-54 (2008).

[37]-[https://es.wikipedia.org/wiki/Ecuación\\_de\\_Arrhenius](https://es.wikipedia.org/wiki/Ecuación_de_Arrhenius) [18/07/2020]

[38]-<https://cfd.direct/openfoam/energy-equation/> [25/07/2020]

[39]-<https://repositorio.unican.es/xmlui/bitstream/handle/10902/7712/379668.pdf?sequence=1&isAllowed=y> [08/08/2020]

# 11. Anexos

## 11.1 simpleFoam.C (\$FOAM\_SOLVERS/incompressible/simpleFoam)

```
/*-----*\
=====
\| / Field      | OpenFOAM: The Open Source CFD Toolbox
\| / Operation  | Website: https://openfoam.org
\| / And       | Copyright (C) 2011-2020 OpenFOAM Foundation
\| Manipulation |
-----\

License
  This file is part of OpenFOAM.

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

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

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

Application
  simpleFoam

Description
  Steady-state solver for incompressible, turbulent flow, using the SIMPLE
  algorithm.

\*-----*/

#include "fvCFD.H"
#include "singlePhaseTransportModel.H"
#include "kinematicMomentumTransportModel.H"
#include "simpleControl.H"
#include "fvOptions.H"

// *****

int main(int argc, char *argv[])
{
    #include "postProcess.H"

```

```

#include "setRootCaseLists.H"
#include "createTime.H"
#include "createMesh.H"
#include "createControl.H"
#include "createFields.H"
#include "initContinuityErrs.H"

turbulence->validate();

// ***** //

Info<< "\nStarting time loop\n" << endl;

while (simple.loop(runTime))
{
    Info<< "Time = " << runTime.timeName() << nl << endl;

    // --- Pressure-velocity SIMPLE corrector
    {
        #include "UEqn.H"
        #include "pEqn.H"
    }

    laminarTransport.correct();
    turbulence->correct();

    runTime.write();

    Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
        << " ClockTime = " << runTime.elapsedClockTime() << " s"
        << nl << endl;
}

Info<< "End\n" << endl;

return 0;
}

// ***** //

```

## 11.2 pEqn.H (\$FOAM\_SOLVERS/incompressible/simpleFoam)

```

{
    volScalarField rAU(1.0/UEqn.A());
    volVectorField HbyA(constrainHbyA(rAU*UEqn.H(), U, p));
    surfaceScalarField phiHbyA("phiHbyA", fvc::flux(HbyA));
    MRF.makeRelative(phiHbyA);
    adjustPhi(phiHbyA, U, p);

    tmp<volScalarField> rAtU(rAU);

```



```

if (simple.consistent())
{
    rAtU = 1.0/(1.0/rAU - UEqn.H1());
    phiHbyA +=
        fvc::interpolate(rAtU() - rAU)*fvc::snGrad(p)*mesh.magSf();
    HbyA -= (rAU - rAtU())*fvc::grad(p);
}

tUEqn.clear();

// Update the pressure BCs to ensure flux consistency
constrainPressure(p, U, phiHbyA, rAtU(), MRF);

// Non-orthogonal pressure corrector loop
while (simple.correctNonOrthogonal())
{
    fvScalarMatrix pEqn
    (
        fvm::laplacian(rAtU(), p) == fvc::div(phiHbyA)
    );

    pEqn.setReference(pRefCell, pRefValue);

    pEqn.solve();

    if (simple.finalNonOrthogonalIter())
    {
        phi = phiHbyA - pEqn.flux();
    }
}

#include "continuityErrs.H"

// Explicitly relax pressure for momentum corrector
p.relax();

// Momentum corrector
U = HbyA - rAtU()*fvc::grad(p);
U.correctBoundaryConditions();
fvOptions.correct(U);
}

```

### 11.3 UEqn.H (\$FOAM\_SOLVERS/incompressible/simpleFoam)

```

// Momentum predictor

MRF.correctBoundaryVelocity(U);

tmp<fvVectorMatrix> tUEqn

```

```

(
    fvm::div(phi, U)
    + MRF.DDt(U)
    + turbulence->divDevSigma(U)
    ==
    fvOptions(U)
);
fvVectorMatrix& UEqn = tUEqn.ref();

UEqn.relax();

fvOptions.constrain(UEqn);

if (simple.momentumPredictor())
{
    solve(UEqn == -fvc::grad(p));

    fvOptions.correct(U);
}

```

#### 11.4 createFields.H (\$FOAM\_SOLVERS/incompressible/simpleFoam)

```

Info<< "Reading field p\n" << endl;
volScalarField p
(
    IOobject
    (
        "p",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);

Info<< "Reading field U\n" << endl;
volVectorField U
(
    IOobject
    (
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);

```

```
#include "createPhi.H"
```

```
label pRefCell = 0;  
scalar pRefValue = 0.0;  
setRefCell(p, simple.dict(), pRefCell, pRefValue);  
mesh.setFluxRequired(p.name());
```

```
singlePhaseTransportModel laminarTransport(U, phi);
```

```
autoPtr<incompressible::momentumTransportModel> turbulence  
(  
    incompressible::momentumTransportModel::New(U, phi, laminarTransport)  
);
```

```
#include "createMRF.H"  
#include "createFvOptions.H"
```

### 11.5 createFields.H (\$FOAM\_SRC/finiteVolume/cfdTools/general/include)

```
IOMRFZoneList MRF(mesh);
```

### 11.6 MRFZone.C (\$FOAM\_SRC/finiteVolume/cfdTools/general/include)

```
void Foam::MRFZone::addCoriolis  
(  
    const volVectorField& U,  
    volVectorField& ddtU  
) const  
{  
    if (cellZoneID_ == -1)  
    {  
        return;  
    }  
  
    const labelList& cells = mesh_.cellZones()[cellZoneID_];  
    vectorField& ddtUc = ddtU.primitiveFieldRef();  
    const vectorField& Uc = U;  
  
    const vector Omega = this->Omega();  
  
    forAll(cells, i)  
    {  
        label celli = cells[i];  
        ddtUc[celli] += (Omega ^ Uc[celli]);  
    }  
}
```

## 11.7 blockMeshDict (\$FOAM\_RUN/MRFSimpleFoamCase/system)

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

backgroundMesh
{
    xMin -0.15;
    xMax  0.15;
    yMin -0.15;
    yMax  0.15;
    zMin -0.04;
    zMax  0.04;
    xCells 30;
    yCells 30;
    zCells 8;
}

convertToMeters 1;

vertices
(
    ($:backgroundMesh.xMin $:backgroundMesh.yMin $:backgroundMesh.zMin)
    ($:backgroundMesh.xMax $:backgroundMesh.yMin $:backgroundMesh.zMin)
    ($:backgroundMesh.xMax $:backgroundMesh.yMax $:backgroundMesh.zMin)
    ($:backgroundMesh.xMin $:backgroundMesh.yMax $:backgroundMesh.zMin)
    ($:backgroundMesh.xMin $:backgroundMesh.yMin $:backgroundMesh.zMax)
    ($:backgroundMesh.xMax $:backgroundMesh.yMin $:backgroundMesh.zMax)
    ($:backgroundMesh.xMax $:backgroundMesh.yMax $:backgroundMesh.zMax)
    ($:backgroundMesh.xMin $:backgroundMesh.yMax $:backgroundMesh.zMax)
);

blocks
(
    hex (0 1 2 3 4 5 6 7)
    (
        $:backgroundMesh.xCells
        $:backgroundMesh.yCells
        $:backgroundMesh.zCells
    )
    simpleGrading (1 1 1)
);

edges
(
);
```

```
boundary
(
);
```

```
mergePatchPairs
(
);
```

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

## 11.8 surfaceFeaturesDict (\$FOAM\_RUN/MRFSimpleFoamCase/system)

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

```
surfaces
(
  "fixedWall.stl"
  "rotatingWall.stl"
  "rotatingZoneWall.stl"
);
```

```
includedAngle 150;
```

```
subsetFeatures
{
  nonManifoldEdges yes;
  openEdges      yes;
}
```

```
trimFeatures
{
  minElem 0;
  minLen 0;
}
```

```
writeObj yes;
```

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

## 11.9 snappyHexMeshDict (\$FOAM\_RUN/MRFSimpleFoamCase/system)

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

```
#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"
```

```
castellatedMesh on;
snap          on;
addLayers     off;
```

```

geometry
{
    fixedWall
    {
        type triSurfaceMesh;
        file "fixedWall.stl";
    }
    rotatingWall
    {
        type triSurfaceMesh;
        file "rotatingWall.stl";
    }
    rotatingZoneWall
    {
        type triSurfaceMesh;
        file "rotatingZoneWall.stl";
    }
};

castellatedMeshControls
{
    features
    (
        { file "fixedWall.eMesh"; level 3; }
        { file "rotatingWall.eMesh"; level 3; }
        { file "rotatingZoneWall.eMesh"; level 3; }
    );

    refinementSurfaces
    {
        fixedWall
        {
            level (3 5);
            patchInfo { type wall; }
        }
        rotatingWall
        {
            level (3 5);
            patchInfo { type wall; }
        }
        rotatingZoneWall
        {
            level (3 5);
            patchInfo { type wall; }
        }
    }

    refinementRegions
    {
        /*

```

```

    rotatingZone
    {
        mode inside;
        levels ((1E15 2));
    }
    */
}

locationInMesh (-0.056 0.009 0.014); // Offset from (0 0 0) to avoid
// coinciding with face or edge
nCellsBetweenLevels 3;
}

snapControls
{
    explicitFeatureSnap true;
    implicitFeatureSnap false;
}

addLayersControls
{
    layers
    {
        "(rotating|fixed)"
        {
            nSurfaceLayers 2;
        }
    }
}

relativeSizes true; // false, usually with firstLayerThickness
expansionRatio 1.2;
finalLayerThickness 0.5;
minThickness 1e-3;
// firstLayerThickness 0.01;

// maxThicknessToMedialRatio 0.6;
}

writeFlags
(
// scalarLevels
layerSets
layerFields
);

mergeTolerance 1e-6;

// ***** //

```

## 11.10 snappyHexMeshDict.cfg (\$FOAM\_ETC/caseDicts/mesh/generation)

```
castellatedMesh true;
snap true;
addLayers true;

castellatedMeshControls
{
    maxLocalCells 100000;
    maxGlobalCells 100000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 3;
    resolveFeatureAngle 30;
    allowFreeStandingZoneFaces true;
    features
    (
    );
}

snapControls
{
    nSmoothPatch 3;
    tolerance 2.0;
    nSolveIter 100;
    nRelaxIter 5;

    nFeatureSnapIter 10;

    explicitFeatureSnap false;
    multiRegionFeatureSnap false;
    implicitFeatureSnap true;
}

addLayersControls
{
    featureAngle 100;
    slipFeatureAngle 30;

    nLayerIter 50;
    nRelaxedIter 20;
    nRelaxIter 5;

    nGrow 0;

    nSmoothSurfaceNormals 1;
    nSmoothNormals 3;
    nSmoothThickness 10;
    maxFaceThicknessRatio 0.5;
    maxThicknessToMedialRatio 0.3;
```



```

minMedialAxisAngle    90;
nMedialAxisIter       10;

nBufferCellsNoExtrude  0;
additionalReporting    false;
// nSmoothDisplacement  0;
// detectExtrusionIsland false;
}

meshQualityControls
{
  #include "${FOAM_CASE}/system/meshQualityDict"
}

debug 0;
mergeTolerance 1e-6;

// ***** //

```

### 11.11 meshQualityDict (\$FOAM\_ETC/caseDicts/mesh/generation)

```

// ***** //

//- Maximum non-orthogonality allowed. Set to 180 to disable.
maxNonOrtho 65;

//- Max skewness allowed. Set to <0 to disable.
maxBoundarySkewness 20;
maxInternalSkewness 4;

//- Max concaveness allowed. Is angle (in degrees) below which concavity
// is allowed. 0 is straight face, <0 would be convex face.
// Set to 180 to disable.
maxConcave 80;

//- Minimum pyramid volume. Is absolute volume of cell pyramid.
// Set to a sensible fraction of the smallest cell volume expected.
// Set to very negative number (e.g. -1E30) to disable.
minVol 1e-13;

//- Minimum quality of the tet formed by the face-centre
// and variable base point minimum decomposition triangles and
// the cell centre. Set to very negative number (e.g. -1E30) to
// disable.
// <0 = inside out tet,
// 0 = flat tet
// 1 = regular tet
minTetQuality 1e-15;

```

```

//- Minimum face area. Set to <0 to disable.
minArea -1;

//- Minimum face twist. Set to <-1 to disable. dot product of face normal
// and face centre triangles normal
minTwist 0.02;

//- Minimum normalised cell determinant. This is the determinant of all
// the areas of internal faces. It is a measure of how much of the
// outside area of the cell is to other cells. The idea is that if all
// outside faces of the cell are 'floating' (zeroGradient) the
// 'fixedness' of the cell is determined by the area of the internal faces.
// 1 = hex, <= 0 = folded or flattened illegal cell
minDeterminant 0.001;

//- Relative position of face in relation to cell centres (0.5 for orthogonal
// mesh) (0 -> 0.5)
minFaceWeight 0.05;

//- Volume ratio of neighbouring cells (0 -> 1)
minVolRatio 0.01;

//- Per triangle normal compared to average normal. Like face twist
// but now per (face-centre decomposition) triangle. Must be >0 for Fluent
// compatibility
minTriangleTwist -1;

nSmoothScale 4;

errorReduction 0.75;

//- If >0 : preserve cells with all points on the surface if the
// resulting volume after snapping (by approximation) is larger than
// minVolCollapseRatio times old volume (i.e. not collapsed to flat cell).
// If <0 : delete always.
//minVolCollapseRatio 0.1;

```

### 11.12 topoSetDict (\$FOAM\_ETC/caseDicts/mesh/generation)

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

```

actions
(
{
name rotatingSet-1;
type cellSet;
action new;
source boxToCell;
sourceInfo
{

```

```

        box (-0.5 -0.5 -0.5) (0.5 0.5 -0.023);
    }
}
{
    name rotatingSet-2;
    type cellSet;
    action new;
    source boxToCell;
    sourceInfo
    {
        box (-0.5 -0.5 0.023) (0.5 0.5 0.5);
    }
}
{
    name rotatingZone-1;
    type cellZoneSet;
    action new;
    source setToCellZone;
    sourceInfo { set rotatingSet-1; }
}
{
    name rotatingZone-2;
    type cellZoneSet;
    action new;
    source setToCellZone;
    sourceInfo { set rotatingSet-2; }
}
);

// ***** //

```

### 11.13 boundary (\$FOAM\_RUN/MRFSimpleFoamCase/constant/polyMesh)

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

```

3
(
    fixedWall
    {
        type wall;
        inGroups List<word> 1(wall);
        nFaces 1197886;
        startFace 19090501;
    }
    rotatingWall
    {
        type wall;
        inGroups List<word> 1(wall);
        nFaces 42713;
        startFace 20288387;
    }
)

```

```

    }
    rotatingZoneWall
    {
        type        wall;
        inGroups    List<word> 1(wall);
        nFaces      367070;
        startFace   20331100;
    }
)
// ***** //

```

#### 11.14 U (\$FOAM\_RUN/MRFSimpleFoamCase/0)

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

```
#include "$FOAM_CASE/constant/initialConditions"
```

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

```
internalField uniform (0 0 0);
```

```
boundaryField
```

```

{
    fixedWall
    {
        type        noSlip;
    }
    rotatingWall
    {
        type        rotatingWallVelocity;
        origin      (0 0 0);
        axis        (0 0 1);
        omega       $wcte;
    }
    rotatingZoneWall
    {
        type        fixedValue;
        value       uniform (0 0 0);
    }
}

```

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

#### 11.15 p (\$FOAM\_RUN/MRFSimpleFoamCase/0)

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

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

```

internalField uniform 0;

boundaryField
{
    fixedWall
    {
        type zeroGradient;
    }
    rotatingWall
    {
        type zeroGradient;
    }
    rotatingZoneWall
    {
        type zeroGradient;
    }
}

// ***** //

```

### 11.16 k (\$FOAM\_RUN/MRFSimpleFoamCase/0)

```

// ***** //

dimensions [0 2 -2 0 0 0];

internalField uniform 0.375;

boundaryField
{
    ".*"
    {
        type kqRWallFunction;
        value $internalField;
    }
}

// ***** //

```

### 11.17 epsilon (\$FOAM\_RUN/MRFSimpleFoamCase/0)

```

// ***** //

dimensions [0 2 -3 0 0 0];

internalField uniform 0.54;

boundaryField
{
    ".*"

```

```

    {
      type      epsilonWallFunction;
      value     $internalField;
    }
  }
}

// ***** //

```

### 11.18 nut (\$FOAM\_RUN/MRFSimpleFoamCase/0)

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

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

```
internalField uniform 0;
```

```
boundaryField
```

```

{
  ".*"
  {
    type      nutkWallFunction;
    value     uniform 0;
  }
}

```

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

### 11.19 transportProperties (\$FOAM\_RUN/MRFSimpleFoamCase/constant)

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

```
transportModel Newtonian;
```

```
nu [0 2 -1 0 0 0] 1.5e-05;
```

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

### 11.20 momentumTransport (\$FOAM\_RUN/MRFSimpleFoamCase/constant)

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

```
simulationType laminar;
```

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

### 11.21 momentumTransport (\$FOAM\_RUN/MRFSimpleFoamCase/constant)

```

// ***** //

simulationType RAS;

RAS
{
    model      kEpsilon;

    turbulence  on;

    printCoeffs  on;
}

// ***** //

11.22 MRFProperties ($FOAM_RUN/MRFSimpleFoamCase/constant)

// ***** //

#include "$FOAM_CASE/constant/initialConditions"

MRF1
{
    cellZone  rotatingZone-1;
    active    yes;

    // Fixed patches (by default they 'move' with the MRF zone)
    nonRotatingPatches ();

    origin    (0 0 0);
    axis      (0 0 1);
    omega     $wcte;
}

MRF2
{
    cellZone  rotatingZone-2;
    active    yes;

    // Fixed patches (by default they 'move' with the MRF zone)
    nonRotatingPatches ();

    origin    (0 0 0);
    axis      (0 0 1);
    omega     $wcte;
}

// ***** //

```

### 11.23 initialConditions (\$FOAM\_RUN/MRFSimpleFoamCase/constant)

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

```
wcte 5;
```

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

### 11.24 controlDict (\$FOAM\_RUN/MRFSimpleFoamCase/system)

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

```
application simpleFoam;
```

```
startFrom latestTime;
```

```
startTime 0;
```

```
stopAt endTime;
```

```
endTime 5000;
```

```
deltaT 1;
```

```
writeControl timeStep;
```

```
writeInterval 200;
```

```
purgeWrite 5;
```

```
writeFormat ascii;
```

```
writePrecision 8;
```

```
writeCompression off;
```

```
timeFormat general;
```

```
timePrecision 6;
```

```
runTimeModifiable true;
```

```
functions
```

```
{  
#includeFunc residuals  
}
```

### 11.25 residuals (\$FOAM\_RUN/MRFSimpleFoamCase/system)

```
\*-----*/
```



```
#includeEtc "caseDicts/postProcessing/numerical/residuals.cfg"
```

```
fields (p U k epsilon);
```

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

## 11.26 fvSchemes (\$FOAM\_RUN/MRFSimpleFoamCase/system)

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

```
ddtSchemes
```

```
{  
  default    steadyState;  
}
```

```
gradSchemes
```

```
{  
  default    Gauss linear;  
  grad(U)    cellLimited Gauss linear 1;  
}
```

```
divSchemes
```

```
{  
  default    none;  
  div(phi,U) bounded Gauss upwind;  
  div(phi,k) bounded Gauss upwind;  
  div(phi,epsilon) bounded Gauss upwind;  
  div((nuEff*dev2(T(grad(U)))) Gauss linear;  
}
```

```
laplacianSchemes
```

```
{  
  default    Gauss linear limited corrected 0.33;  
}
```

```
interpolationSchemes
```

```
{  
  default    linear;  
}
```

```
snGradSchemes
```

```
{  
  default    limited corrected 0.33;  
}
```

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

## 11.27 fvSolution (\$FOAM\_RUN/MRFSimpleFoamCase/system)

```

// ***** //

solvers
{
  p
  {
    solver      GAMG;
    tolerance   1e-08;
    relTol      0.05;
    smoother    GaussSeidel;
    nCellsInCoarsestLevel 500;
  }

  "(U|k|epsilon)"
  {
    solver      smoothSolver;
    smoother    GaussSeidel;
    nSweeps     2;
    tolerance   1e-07;
    relTol      0.1;
  }
}

SIMPLE
{
  nNonOrthogonalCorrectors 0;
  pRefCell 0;
  pRefValue 0;

  residualControl
  {
    U      1e-6;
    p      1e-5;
    "(k|epsilon)" 1e-5;
  }
}

relaxationFactors
{
  fields
  {
    p 0.7;
  }
}

equations
{
  U 0.2;
  k 0.2;
  epsilon 0.2;
}

```

```

}

// ***** //
11.28 decomposeParDict ($FOAM_RUN/MRFSimpleFoamCase/system)
// ***** //

numberOfSubdomains 8;

method      hierarchical;

hierarchicalCoeffs
{
    n      (2 2 2);
    delta  0.001;
    order  xyz;
}

// ***** //

11.29 EEqn.H ($FOAM_SOLVERS/heatTransfer/chtMultiRegionFoam/fluid)

{
    volScalarField& he = thermo.he();

    fvScalarMatrix EEqn
    (
        fvm::ddt(rho, he) + fvm::div(phi, he)
        + fvc::ddt(rho, K) + fvc::div(phi, K)
        + (
            he.name() == "e"
            ? fvc::div
            (
                fvc::absolute(phi/fvc::interpolate(rho), U),
                p,
                "div(phiv,p)"
            )
            : -dpdt
        )
        + thermophysicalTransport.divq(he)
        ==
        rho*(U&g)
        + rad.Sh(thermo, he)
        + reaction.Qdot()
        + fvOptions(rho, he)
    );

    EEqn.relax();

    fvOptions.constrain(EEqn);

```

```

EEqn.solve();

fvOptions.correct(he);

thermo.correct();
rad.correct();

Info<< "Min/max T:" << min(thermo.T()).value() << ' '
    << max(thermo.T()).value() << endl;
}

```

### 11.30 solveSolid.H (\$FOAM\_SOLVERS/heatTransfer/chtMultiRegionFoam/solid)

```

{
  while (pimple.correctNonOrthogonal())
  {
    fvScalarMatrix eEqn
    (
      fvm::ddt(betav*rho, e)
      - (
        thermo.isotropic()
        ? fvm::laplacian(betav*thermo.alpha(), e, "laplacian(alpha,e)")
        : fvm::laplacian(betav*taniAlpha(), e, "laplacian(alpha,e)")
      )
    )
    ==
    fvOptions(rho, e)
  );

  eEqn.relax();

  fvOptions.constrain(eEqn);

  eEqn.solve();

  fvOptions.correct(e);
}

thermo.correct();

Info<< "Min/max T:" << min(thermo.T()).value() << ' '
    << max(thermo.T()).value() << endl;

```

### 11.31 chtMRFMultiRegionFoam (\$FOAM\_SOLVERS/heatTransfer)

```
\*-----*/
```

```
#include "fvCFD.H"
```

```

#include "fluidThermoMomentumTransportModel.H"
#include "rhoReactionThermophysicalTransportModel.H"
#include "rhoReactionThermo.H"
#include "CombustionModel.H"
#include "fixedGradientFvPatchFields.H"
#include "regionProperties.H"
#include "compressibleCourantNo.H"
#include "solidRegionDiffNo.H"
#include "solidThermo.H"
#include "radiationModel.H"
#include "fvOptions.H"
#include "coordinateSystem.H"
#include "pimpleMultiRegionControl.H"
#include "pressureControl.H"

// ***** //

int main(int argc, char *argv[])
{
    #define NO_CONTROL
    #define CREATE_MESH createMeshesPostProcess.H
    #include "postProcess.H"

    #include "setRootCaseLists.H"
    #include "createTime.H"
    #include "createMeshes.H"
    #include "createFields.H"
    #include "initContinuityErrs.H"
    pimpleMultiRegionControl pimples(fluidRegions, solidRegions);
    #include "createFluidPressureControls.H"
    #include "createTimeControls.H"
    #include "readSolidTimeControls.H"
    #include "compressibleMultiRegionCourantNo.H"
    #include "solidRegionDiffusionNo.H"
    #include "setInitialMultiRegionDeltaT.H"

    while (pimples.run(runTime))
    {
        #include "readTimeControls.H"
        #include "readSolidTimeControls.H"

        #include "compressibleMultiRegionCourantNo.H"
        #include "solidRegionDiffusionNo.H"
        #include "setMultiRegionDeltaT.H"

        runTime++;

        Info<< "Time = " << runTime.timeName() << nl << endl;

        // --- PIMPLE loop

```

```

while (pimples.loop())
{
    forAll(fluidRegions, i)
    {
        Info<< "\nSolving for fluid region "
            << fluidRegions[i].name() << endl;
        #include "setRegionFluidFields.H"
        #include "solveFluid.H"
    }

    forAll(solidRegions, i)
    {
        Info<< "\nSolving for solid region "
            << solidRegions[i].name() << endl;
        #include "setRegionSolidFields.H"
        #include "solveSolid.H"
    }
}

runTime.write();

Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
    << " ClockTime = " << runTime.elapsedClockTime() << " s"
    << nl << endl;
}

Info<< "End\n" << endl;

return 0;
}

// ***** //

```

### 11.32 surfaceFeatures (\$FOAM\_RUN/chtMultiRegionFoamCase/system)

```

// ***** //

surfaces
(
    "CAD.stl"

    "cap_motor-magnets.stl"
    "sym_cap_motor-magnets.stl"
    "stator_coil-magnets.stl"

    "cap_motor.stl"
    "cap_motor_rotating.stl"
    "cap_motor-bearing.stl"

    "sym_cap_motor.stl"

```

```

"sym_cap_motor_rotating.stl"
"sym_cap_motor-sym_bearing.stl"

"bearing.stl"
"bearing-shaft.stl"

"sym_bearing.stl"
"sym_bearing-shaft.stl"

"stator.stl"
"stator-stator_coil.stl"
"stator-shaft.stl"

"stator_coil.stl"
"stator_coil-coil.stl"

"coil.stl"

"shaft.stl"
);

includedAngle 150;

subsetFeatures
{
    nonManifoldEdges yes;
    openEdges      yes;
}

trimFeatures
{
    minElem      0;
    minLen       0;
}

writeObj      yes;

// ***** //

11.33 snappyHexMesh ($FOAM_RUN/chtMultiRegionFoamCase/system)

// ***** //

#includeEtc "caseDicts/mesh/generation/snappyHexMeshDict.cfg"

castellatedMesh on;
snap          on;
addLayers     off;

geometry

```

```

{
  CAD
  {
    type triSurfaceMesh;
    file "CAD.stl";

    regions
    {
      cap_motor-ext { name cap_motor-ext; }
      bearing-ext { name bearing-ext; }
      sym_cap_motor-ext { name sym_cap_motor-ext; }
      sym_bearing-ext { name sym_bearing-ext; }
      magnets-ext { name magnets-ext; }
      shaft-ext1 { name shaft-ext1; }
      shaft-ext2 { name shaft-ext2; }
    }
  }

  cap_motor-magnets
  {
    type triSurfaceMesh;
    file "cap_motor-magnets.stl";
  }
  sym_cap_motor-magnets
  {
    type triSurfaceMesh;
    file "sym_cap_motor-magnets.stl";
  }
  stator_coil-magnets
  {
    type triSurfaceMesh;
    file "stator_coil-magnets.stl";
  }

  cap_motor
  {
    type triSurfaceMesh;
    file "cap_motor.stl";
  }
  cap_motor_rotating
  {
    type triSurfaceMesh;
    file "cap_motor_rotating.stl";
  }
  cap_motor-bearing
  {
    type triSurfaceMesh;
    file "cap_motor-bearing.stl";
  }
}

```



```

sym_cap_motor
{
    type triSurfaceMesh;
    file "sym_cap_motor.stl";
}
sym_cap_motor_rotating
{
    type triSurfaceMesh;
    file "sym_cap_motor_rotating.stl";
}
sym_cap_motor-sym_bearing
{
    type triSurfaceMesh;
    file "sym_cap_motor-sym_bearing.stl";
}

bearing
{
    type triSurfaceMesh;
    file "bearing.stl";
}
bearing-shaft
{
    type triSurfaceMesh;
    file "bearing-shaft.stl";
}

sym_bearing
{
    type triSurfaceMesh;
    file "sym_bearing.stl";
}
sym_bearing-shaft
{
    type triSurfaceMesh;
    file "sym_bearing-shaft.stl";
}

stator
{
    type triSurfaceMesh;
    file "stator.stl";
}
stator-stator_coil
{
    type triSurfaceMesh;
    file "stator-stator_coil.stl";
}
stator-shaft
{

```

```

    type triSurfaceMesh;
    file "stator-shaft.stl";
}

stator_coil
{
    type triSurfaceMesh;
    file "stator_coil.stl";
}
stator_coil-coil
{
    type triSurfaceMesh;
    file "stator_coil-coil.stl";
}

coil
{
    type triSurfaceMesh;
    file "coil.stl";
}

shaft
{
    type triSurfaceMesh;
    file "shaft.stl";
}
};

castellatedMeshControls
{
    features
    (
        { file "CAD.eMesh"; level 3; }

        { file "cap_motor-magnets.eMesh"; level 3; }
        { file "sym_cap_motor-magnets.eMesh"; level 3; }
        { file "stator_coil-magnets.eMesh"; level 3; }

        { file "cap_motor.eMesh"; level 3; }
        { file "cap_motor_rotating.eMesh"; level 3; }
        { file "cap_motor-bearing.eMesh"; level 3; }

        { file "sym_cap_motor.eMesh"; level 3; }
        { file "sym_cap_motor_rotating.eMesh"; level 3; }
        { file "sym_cap_motor-sym_bearing.eMesh"; level 3; }

        { file "bearing.eMesh"; level 3; }
        { file "bearing-shaft.eMesh"; level 3; }

        { file "sym_bearing.eMesh"; level 3; }
    )
}

```

```

    { file "sym_bearing-shaft.eMesh"; level 3; }

    { file "stator.eMesh"; level 3; }
    { file "stator-stator_coil.eMesh"; level 3; }
    { file "stator-shaft.eMesh"; level 3; }

    { file "stator_coil.eMesh"; level 3; }
    { file "stator_coil-coil.eMesh"; level 3; }

    { file "coil.eMesh"; level 3; }

    { file "shaft.eMesh"; level 3; }
);

```

refinementSurfaces

```

{
    CAD
    {
        level (4 4);
        patchInfo
        {
            type wall;
            inGroups (externalWall);
        }
    }

    cap_motor-magnets
    {
        level (4 4);
        faceZone cap_motorTomagnets;
    }

    sym_cap_motor-magnets
    {
        level (4 4);
        faceZone sym_cap_motorTomagnets;
    }

    stator_coil-magnets
    {
        level (4 4);
        faceZone magnetsTofluid;
        cellZone magnets;
        cellZoneInside insidePoint;
        insidePoint (0.042 0.120 0.002);
    }

    cap_motor-bearing
    {
        level (4 4);
        faceZone cap_motorTobearing;
    }
}

```

```

}
cap_motor_rotating
{
  level (4 4);
  faceZone cap_motor_rotatingTofluid;
}
cap_motor
{
  level (4 4);
  faceZone cap_motorTofluid;
  cellZone cap_motor;
  cellZoneInside insidePoint;
  insidePoint (-0.045 0.0 0.033);
}

sym_cap_motor-sym_bearing
{
  level (4 4);
  faceZone sym_cap_motorTosym_bearing;
}
sym_cap_motor_rotating
{
  level (4 4);
  faceZone sym_cap_motor_rotatingTofluid;
}
sym_cap_motor
{
  level (4 4);
  faceZone sym_cap_motorTofluid;
  cellZone sym_cap_motor;
  cellZoneInside insidePoint;
  insidePoint (-0.016 -0.017 -0.028);
}

bearing-shaft
{
  level (4 4);
  faceZone bearingToshaft;
}
bearing
{
  level (4 4);
  faceZone bearingTofluid;
  cellZone bearing;
  cellZoneInside insidePoint;
  insidePoint (0.011 0.009 0.029);
}

sym_bearing-shaft
{

```

```

    level (4 4);
    faceZone sym_bearingToshaft;
}
sym_bearing
{
    level (4 4);
    faceZone sym_bearingTofluid;
    cellZone sym_bearing;
    cellZoneInside insidePoint;
    insidePoint (-0.003 0.014 -0.029);
}

stator
{
    level (4 4);
    faceZone statorTofluid;
}
stator-stator_coil
{
    level (4 4);
    faceZone stator-stator_coil;
}
stator-shaft
{
    level (4 4);
    faceZone statorToshaft;
    cellZone stator;
    cellZoneInside insidePoint;
    insidePoint (-0.01 0.006 0.0);
}

stator_coil
{
    level (4 4);
    faceZone stator_coilTofluid;
}
stator_coil-coil
{
    level (4 4);
    faceZone stator_coilTocoil;
    cellZone stator_coil;
    cellZoneInside insidePoint;
    insidePoint (0.112 0.003 0.0);
}

coil
{
    level (4 4);
    faceZone coilTofluid;
    cellZone coil;
}

```

```

        cellZoneInside insidePoint;
        insidePoint (0.116 -0.001 0.0);
    }

    shaft
    {
        level (4 4);
        faceZone shaftTofluid;
        cellZone shaft;
        cellZoneInside insidePoint;
        insidePoint (-0.001 -0.001 -0.018);
    }
}

refinementRegions
{
    CAD
    {
        mode inside;
        levels ((1E15 2));
    }
}

locationInMesh (-0.023 0.026 -0.03); // Offset from (0 0 0) to avoid
// coinciding with face or edge
nCellsBetweenLevels 3;
}

snapControls
{
    //tolerance 3.0;
    explicitFeatureSnap true;
    implicitFeatureSnap false;
    multiRegionFeatureSnap true;
}

addLayersControls
{
    layers
    {
        "CAD.*"
        {
            nSurfaceLayers 2;
        }
    }
}

relativeSizes true; // false, usually with firstLayerThickness
expansionRatio 1.2;
finalLayerThickness 0.5;
minThickness 1e-3;

```

```

// firstLayerThickness 0.01;

// maxThicknessToMedialRatio 0.6;
}

meshQualityControls
{
    maxNonOrtho      83.0;
    minFaceWeight    0.02;
}

writeFlags
(
// scalarLevels
  layerSets
  layerFields
);

mergeTolerance 1e-6;

// ***** //

```

### 11.34 p (\$FOAM\_RUN/chtMultiRegionFoamCase/0/solid)

```

// ***** //

#include "$FOAM_CASE/constant/initialConditions"

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

internalField   uniform $pInitial;

boundaryField
{
    ".*"
    {
        type      calculated;
        value     uniform $pInitial;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

// ***** //

```

### 11.35 T (externalWall) (\$FOAM\_RUN/chtMultiRegionFoamCase/0/solid)

```

{
    type      externalWallHeatFluxTemperature;
}

```

```

mode      coefficient;
h         uniform 10;
Ta        $internalField;
value     $internalField;
}

```

### 11.36 T (coupledWall) (\$FOAM\_RUN/chtMultiRegionFoamCase/0/solid)

```

{
type      compressible::turbulentTemperatureCoupledBaffleMixed;
value     $internalField;
Tnbr      T;
}

```

### 11.37 T (adiabaticWall) (\$FOAM\_RUN/chtMultiRegionFoamCase/0/solid)

```

magnet_to_fluid/fluid_to_magnet
{
type      zeroGradient;
}

```

### 11.38 p (\$FOAM\_RUN/chtMultiRegionFoamCase/0/fluid)

```

// ***** //

#include "$FOAM_CASE/constant/initialConditions"

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

internalField uniform $pInitial;

boundaryField
{
    . **
    {
type      calculated;
value     uniform $pInitial;
}

#includeEtc "caseDicts/setConstraintTypes"
}

// ***** //

```

### 11.39 T (\$FOAM\_RUN/chtMultiRegionFoamCase/0/fluid)

```

// ***** //

#include "$FOAM_CASE/constant/initialConditions"

```



```

dimensions    [0 0 0 1 0 0 0];

internalField uniform $Tinitial;

boundaryField
{
    // **
    {
        type          compressible::turbulentTemperatureCoupledBaffleMixed;
        value         $internalField;
        Tnbr          T;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

// ***** //

```

#### 11.40 U (\$FOAM\_RUN/chtMultiRegionFoamCase/0/fluid)

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

```
#include "$FOAM_CASE/constant/initialConditions"
```

```

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

internalField uniform (0 0 0);

boundaryField
{
    sym_bearingTofluid_fluid_to_sym_bearing
    {
        type          noSlip;
    }
    bearingTofluid_fluid_to_bearing
    {
        type          noSlip;
    }
    coilTofluid_fluid_to_coil
    {
        type          noSlip;
    }
    statorTofluid_fluid_to_stator
    {
        type          noSlip;
    }
    sym_cap_motorTofluid_fluid_to_sym_cap_motor
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
}

```

```

}
sym_cap_motor_rotatingTofluid_fluid_to_sym_cap_motor
{
    type        rotatingWallVelocity;
    origin      (0 0 0);
    axis        (0 0 1);
    omega       $wcte;
}
cap_motorTofluid_fluid_to_cap_motor
{
    type        fixedValue;
    value       uniform (0 0 0);
}
cap_motor_rotatingTofluid_fluid_to_cap_motor
{
    type        rotatingWallVelocity;
    origin      (0 0 0);
    axis        (0 0 1);
    omega       $wcte;
}
stator-stator_coil_fluid_to_stator_coil
{
    type        noSlip;
}
stator_coilTofluid_fluid_to_stator_coil
{
    type        noSlip;
}
stator_coilTocoil_fluid_to_stator_coil
{
    type        noSlip;
}
coilTofluid_fluid_to_stator_coil
{
    type        noSlip;
}
shaftTofluid_fluid_to_shaft
{
    type        noSlip;
}

#includeEtc "caseDicts/setConstraintTypes"
}

// ***** //

11.41 p_rgh ($FOAM_RUN/chtMultiRegionFoamCase/0/fluid)

// ***** //

```

```

#include "$FOAM_CASE/constant/initialConditions"

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

internalField uniform 0;

boundaryField
{
    .
    {
        type        fixedFluxPressure;
        value        $internalField;
    }

    #includeEtc "caseDicts/setConstraintTypes"
}

// ***** //

```

#### 11.42 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/aluminium)

```

// ***** //

mixture
{
    specie
    {
        nMoles    1;
        molWeight 27; // [g/mol]
    }

    transport
    {
        kappa 240; // [W/m/K]
    }

    thermodynamics
    {
        Hf    0;
        Cv    896; // [J/kg/K]
    }

    equationOfState
    {
        rho 2712; // [kg/m^3]
    }
}

// ***** //

```

### 11.43 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/NdFeB)

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

```
mixture
{
  specie
  {
    nMoles 1;
    molWeight 209; // [g/mol]
  }

  transport
  {
    kappa 9; // [W/m/K]
  }

  thermodynamics
  {
    Hf 0;
    Cv 450; // [J/kg/K]
  }

  equationOfState
  {
    rho 7500; // [kg/m^3]
  }
}
```

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

### 11.44 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/copper)

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

```
mixture
{
  specie
  {
    nMoles 1;
    molWeight 63.5; // [g/mol]
  }

  transport
  {
    kappa 380; // [W/m/K]
  }
}
```

```

thermodynamics
{
  Hf    0;
  Cv    385;    // [J/kg/K]
}

equationOfState
{
  rho   8940;    // [kg/m^3]
}
}

```

// \*\*\*\*\* //

**11.45 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/iron)**

// \*\*\*\*\* //

```

mixture
{
  specie
  {
    nMoles    1;
    molWeight  56; // [g/mol]
  }

  transport
  {
    kappa  80; // [W/m/K]
  }

  thermodynamics
  {
    Hf    0;
    Cv    450; // [J/kg/K]
  }

  equationOfState
  {
    rho   7874; // [kg/m^3]
  }
}

```

// \*\*\*\*\* //

**11.46 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/steel)**

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

mixture
{
  specie
  {
    nMoles 1;
    molWeight 56; // [g/mol]
  }

  transport
  {
    kappa 54; // [W/m/K]
  }

  thermodynamics
  {
    Hf 0;
    Cv 500; // [J/kg/K]
  }

  equationOfState
  {
    rho 7850; // [kg/m^3]
  }
}

```

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

#### 11.47 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/solid)

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

```
thermoType
{
  type heSolidThermo;
  mixture pureMixture;
  transport constIso;
  thermo eConst;
  equationOfState rhoConst;
  specie specie;
  energy sensibleInternalEnergy;
}

```

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

#### 11.48 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/air)

```

// ***** //

thermoType
{
    type        heRhoThermo;
    mixture     pureMixture;
    transport   sutherland;
    thermo      janaf;
    equationOfState perfectGas;
    specie      specie;
    energy      sensibleInternalEnergy;
}

mixture
{
    specie
    {
        nMoles      1;
        molWeight    28.9596;
    }
    thermodynamics
    {
        Tlow        200;
        Thigh       3500;
        Tcommon     1000;
        highCpCoeffs ( 3.57304 -7.24383e-04 1.67022e-06 -1.26501e-10 -
4.20580e-13 -1047.41 3.12431 );
        lowCpCoeffs ( 3.09589 1.22835e-03 -4.14267e-07 6.56910e-11 -
3.87021e-15 -983.191 5.34161 );
    }
    transport
    {
        As          1.67212e-06;
        Ts          170.672;
    }
}

// ***** //

```

#### 11.49 regionProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant)

```

// ***** //

regions
(
    fluid    (fluid)
    solid    (bearing cap_motor coil magnets shaft stator stator_coil
sym_bearing sym_cap_motor)
);

```

```
// ***** //  
11.50 controlDict ($FOAM_RUN/chtMultiRegionFoamCase/system)
```

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

```
application    chtMultiRegionFoam;  
  
startFrom      startTime;  
  
startTime      0;  
  
stopAt         endTime;  
  
endTime        5000;  
  
deltaT         1;  
  
writeControl   timeStep;  
  
writeInterval  200;  
  
purgeWrite     5;  
  
writeFormat    ascii;  
  
writePrecision 8;  
  
writeCompression off;  
  
timeFormat     general;  
  
timePrecision  6;  
  
runTimeModifiable true;  
  
functions  
{  
    residualsBearing  
    {  
        type    residuals;  
        libs    ("libutilityFunctionObjects.so");  
        region  bearing;  
        fields  (e);  
    }  
  
    residualsCap_motor  
    {  
        type    residuals;  
        libs    ("libutilityFunctionObjects.so");  
    }  
}
```



```

region cap_motor;
fields (e);
}

residualsCoil
{
type residuals;
libs ("libutilityFunctionObjects.so");
region coil;
fields (e);
}

residualsFluid
{
type residuals;
libs ("libutilityFunctionObjects.so");
region fluid;
fields (U e p_rgh);
}

residualsMagnets
{
type residuals;
libs ("libutilityFunctionObjects.so");
region magnets;
fields (e);
}

residualsShaft
{
type residuals;
libs ("libutilityFunctionObjects.so");
region shaft;
fields (e);
}

residualsStator
{
type residuals;
libs ("libutilityFunctionObjects.so");
region stator;
fields (e);
}

residualsStator_coil
{
type residuals;
libs ("libutilityFunctionObjects.so");
region stator_coil;
fields (e);
}

```

```

    }

    residualsSym_bearing
    {
        type    residuals;
        libs    ("libutilityFunctionObjects.so");
        region sym_bearing;
        fields (e);
    }

    residualsSym_cap_motor
    {
        type    residuals;
        libs    ("libutilityFunctionObjects.so");
        region sym_cap_motor;
        fields (e);
    }
}

// ***** //

```

### 11.51 fvSchemes (\$FOAM\_RUN/chtMultiRegionFoamCase/system/solid)

```

// ***** //

ddtSchemes
{
    default    steadyState;
}

gradSchemes
{
    default    Gauss linear;
}

divSchemes
{
    default    none;
}

laplacianSchemes
{
    default    Gauss linear uncorrected;
}

interpolationSchemes
{
    default    linear;
}

```

```

snGradSchemes
{
    default    uncorrected;
}

```

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

## 11.52 fvSchemes (\$FOAM\_RUN/chtMultiRegionFoamCase/system/fluid)

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

```

ddtSchemes
{
    default    steadyState;
}

```

```

gradSchemes
{
    default    Gauss linear;
    grad(U)    cellLimited Gauss linear 1;
}

```

```

divSchemes
{
    default    none;
    div(phi,U)    bounded Gauss upwind;
    div(phi,h)    bounded Gauss upwind;
    div(phi,e)    bounded Gauss upwind;
    div(phi,K)    bounded Gauss upwind;
    div(phiv,p)    Gauss upwind;
    div((((rho*nuEff)*dev2(T(grad(U)))))) Gauss linear;
}

```

```

laplacianSchemes
{
    default    Gauss linear limited corrected 0.33;
}

```

```

interpolationSchemes
{
    default    linear;
}

```

```

snGradSchemes
{
    default    limited corrected 0.33;
}

```

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

### 11.53 fvSolution (\$FOAM\_RUN/chtMultiRegionFoamCase/system/solid)

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

```
solvers
```

```
{  
  "(h|e)"  
  {  
    solver      PCG;  
    preconditioner DIC;  
    tolerance   1e-06;  
    relTol     0.1;  
  }  
}
```

```
"(h|e)Final"  
{  
  $h;  
}  
}
```

```
PIMPLE
```

```
{  
  nNonOrthogonalCorrectors 0;  
}
```

```
relaxationFactors
```

```
{  
  equations  
  {  
    "(h|e).*" 0.7;  
  }  
}
```

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

### 11.54 fvSolution (\$FOAM\_RUN/chtMultiRegionFoamCase/system/fluid)

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

```
solvers
```

```
{  
  "p_rgh.*"  
  {  
    solver      GAMG;  
    tolerance   1e-7;  
    relTol     0.01;  
    smoother    DIC;  
    maxIter    100;  
  }  
}
```

```

"(U|h|e).*"
{
  solver      PBiCGStab;
  preconditioner DILU;
  tolerance   1e-6;
  relTol      0.1;
}
}

PIMPLE
{
  nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
  fields
  {
    rho      1;
    p_rgh    0.7;
  }

  equations
  {
    U        0.3;
    "(h|e)"  0.3;
  }
}

// ***** //

```

### 11.55 fvOptions (\$FOAM\_RUN/chtMultiRegionFoamCase/system/coil)

```

// ***** //

fixedTemperature
{
  type      fixedTemperatureConstraint;
  active    yes;
  selectionMode all;

  mode      uniform;

  temperature constant 335; // Set temperature (K)
}

//***** //

```

## 11.56 thermophysicalProperties (\$FOAM\_RUN/chtMultiRegionFoamCase/constant/natural\_ester

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

thermoType
{
    type        heRhoThermo;
    mixture     pureMixture;
    transport   polynomial;
    thermo      hPolynomial;
    equationOfState icoPolynomial;
    specie      specie;
    energy      sensibleEnthalpy;
}

mixture
{
    specie
    {
        nMoles    1;
        molWeight 175; // [g/mol]
    }
    thermodynamics
    {
        Hf    0;
        Sf    0;
        CpCoeffs<8> (3125.632763 -10.122025 0.020875 0 0 0 0 0); // [J/kg/K]
    }
    equationOfState
    {
        rhoCoeffs<8> (1114.1985 -0.6608 0 0 0 0 0 0); // [kg/m^3]
    }
    transport
    {
        muCoeffs<8> (4.5300037869 -0.0365455972 9.8785281086e-05 -
8.9318312852e-08 0 0 0 0); // [kg/m/s]
        kappaCoeffs<8> (0.20631363 -0.00011626 0 0 0 0 0 0); // [W/m·K]
    }
}

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