



**Universidad**  
Zaragoza

Trabajo Fin de Grado

SIMULACIÓN AVANZADA DE FLUJO  
TURBULENTO EN TURBOMÁQUINAS  
MEDIANTE EL MÉTODO DE ELEMENTOS  
FINITOS

Autor

Emilio José Escobedo Sevilla

Directores

Dr. Guillermo Hauke Bernardos

Dr. Diego Irisarri Jiménez

Área de Mecánica de Fluidos  
Departamento de Ciencia y Tecnología de Materiales y Fluidos  
Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza (EINA)  
Octubre, 2019. Zaragoza

Repositorio de la Universidad de Zaragoza – Zagan <http://zagan.unizar.es>

# SIMULACIÓN AVANZADA DE FLUJO TURBULENTO EN TURBOMÁQUINAS MEDIANTE EL MÉTODO DE ELEMENTOS FINITOS

## Resumen

El presente Trabajo de Fin de Grado, enmarcado en el campo de la mecánica de fluidos computacional, expone el proceso de evolución y aplicación de un código de elementos finitos, basado en las ecuaciones de Navier-Stokes para flujo incompresible bidimensional y programado por investigadores del *Departamento de Ciencia y Tecnología de Materiales y Fluidos de la Universidad de Zaragoza* y del *Laboratorio de Investigación en Fluidodinámica y Tecnologías de la Combustión (LIFTEC)*, cuyo grupo de investigación persigue el desarrollo de un código propio de simulación para la predicción de fenómenos complejos. Las tareas aquí realizadas forman parte de dicho proyecto de investigación.

El objetivo del proceso de evolución del algoritmo es incluir el régimen de flujo turbulento en el abanico de aplicación, inicialmente circunscrito únicamente al régimen laminar, siempre en el contexto de flujos incompresibles bidimensionales. La versión inicial del código ha sido modificada por los investigadores que la desarrollaron, introduciendo modelos de turbulencia considerablemente innovadores, siendo objeto de este Trabajo Fin de Grado realizar la validación de las modificaciones.

El grado de éxito que suponen las modificaciones mencionadas se constata mediante comparaciones con un problema ampliamente estudiado por la fluidodinámica computacional a lo largo de los años y empleado como benchmark habitual en proyectos de investigación. Dicho problema es el LDC (*Lid Driven Cavity*). En el contexto de este problema, también se realiza un estudio de sensibilidad para los diferentes tipos de mallas y su grado resolución.

En el campo de la aplicación práctica, se emplea el código modificado en la simulación de un modelo simplificado (2D) de un ventilador centrífugo cuyo flujo es turbulento en la realidad. Finalmente, con el objetivo de extraer la mayor información posible de los resultados (campo de presión y de velocidad), se desarrolla íntegramente un algoritmo de post-procesado que, mediante el cálculo de parámetros como el rendimiento y el incremento energético, permite generar propuestas de optimización y realizar comparaciones cuantitativas con simulaciones de la misma turbomáquina basadas en el código original.

# AGRADECIMIENTOS

Dedico estas breves líneas para agradecer el apoyo recibido por parte de Diego Irisarri y de Guillermo Hauke, ambos han sido verdaderos mentores. Su paciencia y amabilidad han sido infinitas durante todo el tiempo que ha abarcado la realización de este trabajo

Muchas gracias por vuestra disposición y por vuestros conocimientos, pero sobre todo, por brindarme vuestro tiempo intentando compartirlos.

Agradecer también al CSIC la beca recibida puesto que ha permitido la realización del presente trabajo en las mejores condiciones posibles y a todos los miembros de la EINA, por ser la escuela en cuyo seno ha acontecido a lo largo de estos años una increíble evolución de mi persona, tanto a nivel académico como personal.

Finalmente, manifestar gratitud por el constante respaldo que han sido capaces de ofrecerme, consciente o inconscientemente, mi pareja y todos los miembros de mi familia y de mi círculo de amistades.

En Zaragoza, a 20 de octubre de 2019

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos y alcance . . . . .	2
1.2. Herramientas empleadas . . . . .	4
<b>2. Marco teórico del código</b>	<b>6</b>
2.1. Ecuaciones de Navier-Stokes y Principio de Continuidad . . . . .	6
2.2. Formulación fuerte y formulación débil del problema . . . . .	8
2.3. LES vs DNS. Escalas de Kolmogórov . . . . .	10
<b>3. Validación del modelo turbulento</b>	<b>13</b>
3.1. Fundamentos LDC . . . . .	13
3.2. Resultados y comparativas LDC . . . . .	14
3.3. Análisis de sensibilidad. La influencia de la malla . . . . .	20
<b>4. Simulación de la turbomáquina</b>	<b>26</b>
4.1. Multiple Reference Frame . . . . .	26
4.2. Diseño del modelo 2-D y de su malla. Geometría base . . . . .	28
4.3. Código de post-procesado . . . . .	33
4.4. Resultados principales . . . . .	35
4.4.1. Caso base . . . . .	35
4.4.2. Comparaciones con resultados laminares y experimentales . . . . .	39
4.5. Propuesta de optimización . . . . .	41
<b>5. Conclusiones generales</b>	<b>45</b>
<b>Bibliografía</b>	<b>48</b>
<b>Índice de figuras</b>	<b>50</b>
<b>Índice de tablas</b>	<b>51</b>
<b>Anexos</b>	<b>52</b>
A. Código fuente algoritmo de post-proceso. Turbomáquina . . . . .	52
B. Código fuente algoritmo de cálculo del error. LDC . . . . .	68

---

# 1. Introducción

Es ampliamente conocida la existencia de dos modalidades o regímenes de flujo en la mecánica de fluidos, uno ordenado y suave, denominado laminar y otro, oscilante y caótico, designado turbulento. El interés del estudio de la turbulencia estriba en su ubicuidad, tanto en la naturaleza como en aplicaciones de ingeniería, hasta el punto de que la mayoría de los flujos de interés tecnológico resultan ser turbulentos. Dado que la turbulencia modifica significativamente parámetros tales como la resistencia a la fricción, la transmisión de calor o la capacidad de mezcla, es necesario un estudio de los mismos con el fin de desarrollar modelos matemáticos que puedan reproducir con precisión los fenómenos inherentes a ellos.

Resulta complejo dar una definición de la turbulencia en pocas palabras, sin embargo, asociados a ella existen ciertos patrones; produce movimientos irregulares muy marcados por ínfimas variaciones en las condiciones iniciales, llegando a aparentar aleatoriedad y pudiéndose enmarcar dentro de la teoría del caos; lleva a cabo una homogeneización de las características del fluido, como temperatura o concentración, debido a la mezcla rápida de las partículas en él; va acompañada de un alto número de Reynolds ( $Re$ ), debido a que el orden de magnitud de las fuerzas inerciales es mayor que el de las fuerzas viscosas; aparecen vórtices de diferentes escalas interactuando entre ellos, alterando los movimientos de otros más pequeños; y disipa energía en forma de calor debido al trabajo de deformación realizado por los esfuerzos cortantes en las escalas menores.

A la gran importancia de estos flujos se añade su complejidad. Es muy difícil realizar un estudio tanto experimental como teórico. Mientras que abordar el cálculo de flujos laminares genera un menor número de dificultades, alcanzando incluso en determinadas ocasiones soluciones analíticas, en el caso de intentar predecir el comportamiento de flujos turbulentos, no existe una teoría completa del fenómeno ni parece que por el momento vaya a ser posible establecer una. La turbulencia y, más aún, la transición de flujo laminar a turbulento, constituye uno de los últimos problemas de la física clásica sin resolver.

Sin embargo, en los últimos años y debido a la rápida evolución de los ordenadores y al desarrollo de algoritmos específicos se ha producido un gran avance en su resolución numérica, lo que se denomina mecánica de fluidos computacional, o *Computational Fluid Dynamics (CFD)* en inglés, siendo precisamente en este contexto general de resolución mediante métodos numéricos del problema de la turbulencia donde se enmarca el desarrollo del presente trabajo.

El uso de métodos numéricos apoyados por la tecnología de hoy día, ofrece diversas aproximaciones a la resolución de un problema, lo que implicará distintas formas de considerar la turbulencia. La más precisa es resolver las ecuaciones de Navier-Stokes numéricamente en mallas computacionales muy refinadas y con pasos temporales muy pequeños. Este enfoque de resolución se denomina *Direct Numerical*

*Simulation (DNS)*. No obstante, este método resulta inviable en aplicaciones de interés ingenieril hoy en día o en un futuro cercano debido al alto coste computacional que conlleva, cuyo orden de magnitud es mostrado más adelante en el apartado 2.3. Otro enfoque consiste en resolver solamente las escalas grandes de un problema y modelizar las escalas más pequeñas (escalas disipativas), denominado *Large Eddy Simulation (LES)*. El código en el que se basa este trabajo está programado en LES puesto que es el enfoque que mejor equilibrio aporta, teniendo en cuenta precisión y tiempo de computación.

El método numérico empleado para resolver las ecuaciones en derivadas parciales de Navier-Stokes es el de los elementos finitos. Éste no es el único método numérico existente para resolver ecuaciones diferenciales ordinarias o en derivadas parciales de los modelos físicos de la naturaleza. Entre sus competidores más directos se pueden mencionar el método de diferencias finitas y el método de volúmenes finitos. En el campo de la mecánica de sólidos, los elementos finitos han desbancado por completo a los métodos restantes por su versatilidad y ante todo, por su flexibilidad para adaptarse a geometrías complejas. Sin embargo, en la simulación de fluidos, coexiste con el método de los volúmenes finitos.

Algunos de los puntos fuertes que abogan por el uso de este método son: la ya mencionada flexibilidad geométrica, siendo posible discretizar un dominio espacial complejo en elementos de geometría sencilla, además de la posibilidad de introducir un mayor número de elementos y de nodos en las regiones en cuyo seno el flujo alberga mayor complejidad y viceversa; la capacidad de fijar externamente el orden de precisión espacial y temporal, variándose simplemente las funciones de forma; la sistematización de su programación, siendo viable el mismo código para elementos de geometrías diferentes o para diferentes órdenes de precisión; la existencia de estimadores de error... Por contra, todas estas ventajas tienen la contrapartida de una mayor complejidad teórica y matemática del método y de su programación, y un mayor coste computacional, materializado tanto en tiempo de cálculo como en demanda de memoria, ambos superiores en comparación con otros métodos para una misma precisión.

## 1.1. **Objetivos y alcance**

Una vez introducidas someramente tanto las características principales del flujo turbulento, como la herramienta de resolución numérica que resulta ser el CFD, y concretados el enfoque (LES) y el método (Elementos Finitos) en los que se basa el algoritmo empleado, resulta pertinente detallar los objetivos del presente trabajo. Antes de ello cabe recordar brevemente que las modificaciones sobre el algoritmo base persiguen incluir el régimen de flujo turbulento en el abanico de aplicación del código, inicialmente circunscrito únicamente al régimen laminar, siempre en el contexto de flujos incompresibles bidimensionales.

Los objetivos principales se pueden compendiar de forma breve en dos puntos muy relacionados, siendo básico el cumplimiento del primero para alcanzar el se-

gundo. En primer lugar, realizar la validación de los novedosos términos turbulentos implementados en el algoritmo y posteriormente, simular y analizar el flujo de una turbomáquina además de generar propuestas para su optimización. La turbomáquina estudiada es un ventilador centrífugo con álabes curvados hacia delante o *Forward curved centrifugal fan* en inglés.

Para concluir el prefacio introductorio, se facilita una sucinta exposición del alcance de las secciones posteriores, que trata de transmitir el modo en que se persigue la consecución de los objetivos previamente referidos, y una breve enumeración de las herramientas que permiten la ejecución de este trabajo.

En la sección 2 se incluyen breves nociones teóricas de los principios físicos y de las ecuaciones que conforman la base principal del algoritmo. A partir de la formulación fuerte o diferencial de las ecuaciones mencionadas, se presenta el paso a la formulación débil, base del método de elementos finitos. También se realiza un inciso referido a las escalas de Kolmogórov, relevantes en posteriores decisiones relacionadas con el mallado de los dominios espaciales.

A continuación, en la sección 3 se desarrollan las comparaciones necesarias para crear referencias adecuadas y consistentes que doten de confianza a los resultados del algoritmo empleado en aplicaciones prácticas. Las comparaciones se articulan con un problema ampliamente estudiado por la fluidodinámica computacional a lo largo de los años y empleado como benchmark habitual en proyectos de investigación como es el LDC (*Lid Driven Cavity*). Se desarrolla un esquema basado en una presentación inicial de los detalles más generales (geometría, condiciones de contorno, resultados más relevantes...) continuada de una comparación minuciosa entre los resultados obtenidos por el algoritmo a validar y los de contrastados artículos de investigación. Finalmente se realiza un análisis de sensibilidad con respecto a diferentes mallas, siendo los principales parámetros a optimizar la densidad de elementos y su distribución espacial.

La sección 4 contiene la exposición de las simulaciones de la turbomáquina, es decir, del ventilador centrífugo con álabes curvados hacia delante. Este apartado se centra en la aplicación tecnológica del algoritmo validado anteriormente, comenzando con unas breves pinceladas del método MRF (*Multiple Reference Frames*) necesario al trabajar en la resolución de flujos en dominios con velocidades relativas de rotación. A continuación se diseña el modelo simplificado 2-D del ventilador y su correspondiente malla, fijando las condiciones de contorno. También se comparan los resultados obtenidos con simulaciones basadas en el código original y con datos experimentales de la turbomáquina real que se pretende modelar. Finalmente se desarrolla de forma original e íntegra un algoritmo de post-procesado que, mediante el cálculo de parámetros como el rendimiento y el incremento energético, permite generar propuestas de optimización de la turbomáquina estudiada.

Por último, en el apartado 5 se recogen las conclusiones, los resultados más relevantes y los aspectos más significativos a los que ha dado lugar el desarrollo del

Trabajo Fin de Grado. Tras él, se presenta la bibliografía, los índices de tablas y figuras y en el final del documento se ha incluido una sección de apéndices con el código fuente de los algoritmos de post-procesado desarrollados de forma original.

La figura 1 pretende describir visualmente el contexto y el alcance del presente Trabajo de Fin de Grado, recientemente detallado en los párrafos anteriores.

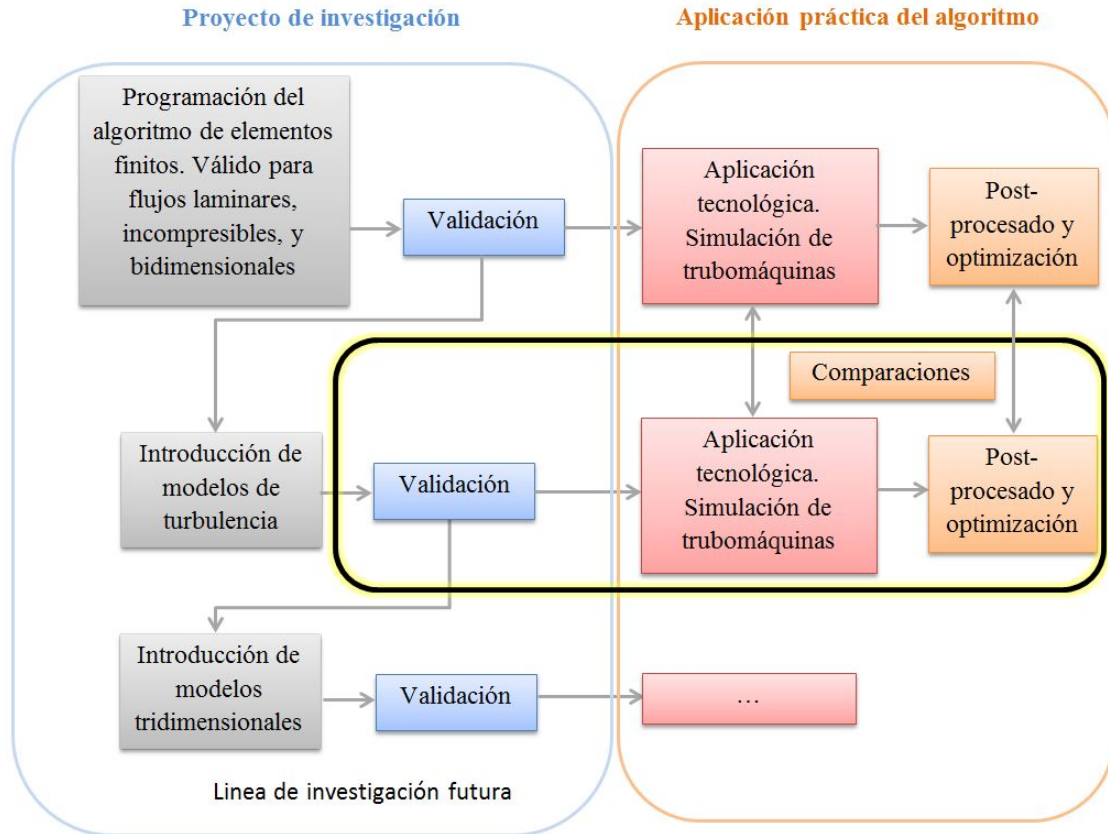


Figura 1: Contexto del trabajo en el proyecto de investigación

## 1.2. Herramientas empleadas

En lo referido a las herramientas empleadas, en primer lugar, las herramientas de hardware son;

- Para el procesamiento y cálculo de las simulaciones, un ordenador que consta de un procesador Intel Core i7-2600 3.40 GHz y 16 GB de memoria RAM, que han sido suficientes a la hora de realizar las mallas y resolver los cálculos numéricos.
- Por otra parte un ordenador personal que incorpora un procesador Intel Core i7-7500 2.70 GHz y 8 GB de memoria RAM, utilizado mayormente durante el post-proceso de los resultados

En segundo lugar, las herramientas de software necesarias que permiten desarrollar el proyecto son;



- **Solver de elementos finitos:** objeto de estudio y principal herramienta. Código en el que se basan todas las simulaciones numéricas realizadas y cuyas pruebas de desarrollo han sido parte del proyecto. Ha sido desarrollado prácticamente desde cero en el Área de Mecánica de Fluidos de la Universidad de Zaragoza por investigadores del mencionado Área y del LIFTEC. Está programado en lenguaje FORTRAN95 y diseñado en base a las ecuaciones de Navier-Stokes para resolver flujos incomprensibles, estacionarios y bidimensionales.
- **GID:** es un programa desarrollado por International Center for Numerical Methods in Engineering (CIMNE) que tiene utilidades para el pre y post-proceso de métodos numéricos como el de los elementos finitos. A la hora de lanzar una simulación es necesario que la geometría del problema este correctamente mallada, es decir, su dominio discretizado en elementos de manera que el dominio continuo pueda ser tratado como una matriz. GID ha sido empleado para este proceso de discretización del dominio y para dotarlo de las condiciones de contorno propicias.
- **Paraview:** es una herramienta de código abierto para el post-proceso de cálculos numéricos. Es utilizada en este proyecto para comprobar de manera visual las soluciones de campos de velocidades y de presiones obtenidas durante las simulaciones.
- **Autodesk Inventor:** es una potente herramienta de diseño gráfico para ingeniería que ha sido usada en la fase del proyecto de diseño de las geometrías de la turbomáquina.

---

## 2. Marco teórico del código

La base física sobre la que se sustenta y de la que nace el solver de elementos finitos está constituida por las ecuaciones en derivadas parciales de Navier-Stokes y el principio de continuidad, aceptando las hipótesis de flujo bidimensional e incompresible y restringido a fluidos newtonianos.

Esta base también es válida como punto de partida en el intento de modelar flujos turbulentos. Pese a que los flujos turbulentos están caracterizados por la transitoriedad y por el amplio rango de escalas espaciales y temporales, incluso las escalas más pequeñas y con fluctuaciones más rápidas, las escalas disipativas, son varios órdenes de magnitud superiores a las escalas moleculares, por lo que se pueden seguir aplicando con validez las ecuaciones constitutivas de la viscosidad de Stokes para flujos newtonianos. Estas leyes constitutivas junto con el Principio Fundamental de la Dinámica, dan lugar a las ecuaciones de Navier-Stokes, ecuaciones que gobiernan los flujos laminares y también los turbulentos.

### 2.1. Ecuaciones de Navier-Stokes y Principio de Continuidad

Las ecuaciones de Navier-Stokes en forma fuerte se pueden deducir de la aplicación de la Segunda Ley de Newton sobre una partícula fluida diferencial. Mediante un enfoque euleriano y desde una referencia absoluta se llega a la siguiente formulación vectorial;

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{u} \quad (1)$$

En (1), el término  $\mathbf{f}$  corresponde al vector de fuerzas másicas,  $\mathbf{u}$  al vector velocidad y  $p$  al campo escalar de presión. Además,  $\rho$  y  $\mu$  son los valores escalares que representan las propiedades del fluido, densidad y viscosidad dinámica respectivamente.

Los términos a la izquierda del igual en (1) nacen de la derivada material o sustancial de la velocidad con respecto al tiempo, es decir, modelan la aceleración de la partícula fluida;

$$\rho \mathbf{a} = \rho \frac{D\mathbf{u}}{Dt} = \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} \quad (2)$$

Debido a la adopción de la descripción euleriana, la derivada ordinaria  $\partial\phi/\partial t$  ya no refleja toda la variación por unidad de tiempo de una determinada propiedad del fluido (o magnitud fluida)  $\phi$ . Existe, además, la variación asociada al movimiento del fluido, representada por el último término de (2), también denominado término convectivo.

Los términos a la derecha del igual en (1) modelan las fuerzas a las que está sometida la partícula fluida por unidad de volumen. Respectivamente se corresponden con; las fuerzas másicas o volumétricas ( $\rho \mathbf{f}$ ), entre las que se encuentra la gravitatoria; las fuerzas de presión, originadas por el gradiente del campo homónimo ( $-\nabla p$ ); y las fuerzas viscosas, cuyo origen reside en los esfuerzos viscosos tangenciales modelados para fluidos newtonianos ( $\mu \nabla^2 \mathbf{u}$ ), este último también es denominado término difusivo.

Bajo las hipótesis de flujo incompresible y viscosidad contante, es decir,  $\rho$  y  $\mu$  uniformes y conocidas en todo el dominio espacial y temporal, aceptable en condiciones de flujo subsónico y sin efectos térmicos, las ecuaciones bidimensionales de Navier-Stokes relacionan tres variables incógnita  $\mathbf{u} = (u, v)$  y  $p$  mediante únicamente dos ecuaciones diferenciales.

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial p}{\partial x} = 0 \quad (3)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} - \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{\partial p}{\partial y} = 0 \quad (4)$$

Las ecuaciones diferenciales (3) y (4) corresponden a las componentes extendidas de la ecuación diferencial vectorial (1). Con la consideración adicional, impuesta desde este momento,  $\mathbf{f} = \mathbf{0}$ , despreciando el efecto de las fuerzas másicas como la gravedad, tal y como se hace en la programación del solver. Nótese que  $\nu$  simboliza la viscosidad cinemática;  $\mu/\rho$ .

Surge de esta indeterminación (dos ecuaciones diferenciales y tres variables incógnita) la necesidad de aplicar el principio continuidad, formulado a través de la ecuación de conservación de la masa como sigue;

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (5)$$

Puesto que se acepta la hipótesis de flujo incompresible, la ecuación de continuidad se puede formular de forma equivalente como la divergencia de la velocidad igualada a cero en todo el dominio espacial;

$$\nabla \cdot \mathbf{u} = 0 \quad (6)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (7)$$

De este modo se dispone de tres ecuaciones diferenciales (3), (4), (7) y tres variables incógnita ( $u, v, p$ ). En caso de no aceptar la hipótesis de flujo incompresible, la densidad constituiría un cuarto campo escalar incógnita, por lo que sería necesario contemplar y formular el principio de conservación de la energía.

## 2.2. Formulación fuerte y formulación débil del problema

El sistema de ecuaciones diferenciales está compuesto por las expresiones mostradas en el apartado 2.1, ver (3), (4) y (7). Junto con las condiciones de contorno conforman la denominada formulación fuerte del problema (8). Siendo  $\Omega$  el dominio del problema definido en  $\mathbb{R}^2$  con frontera  $\Gamma$ , repartida en  $\Gamma_g$  y  $\Gamma_h$ , donde se imponen condiciones de contorno de tipo Dirichlet y de tipo Neumann respectivamente, tal que  $\Gamma_g \cup \Gamma_h = \Gamma$  y  $\Gamma_g \cap \Gamma_h = \emptyset$ .

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p / \rho - \nu \nabla^2 \mathbf{u} = \mathbf{0} \quad \text{en } \Omega \\ \nabla \cdot \mathbf{u} = 0 \quad \text{en } \Omega \\ \mathbf{u} = \mathbf{g} \quad \text{en } \Gamma_g \\ \nu \nabla \mathbf{u} \cdot \mathbf{n} = \mathbf{h} \quad \text{en } \Gamma_h \end{array} \right. \quad (8)$$

La formulación fuerte alberga ecuaciones en derivadas parciales con términos no lineales, dotando a la resolución analítica de una complejidad inabordable. Tiene mucho más sentido enfocar la resolución mediante un método numérico como el de elementos finitos. El objetivo básico de este tipo de métodos se puede resumir mediante las siguientes palabras; convertir el sistema diferencial en un sistema de ecuaciones algebraico lineal, fácilmente computable y por tanto, asequible para que lo resuelva un ordenador. Para ello, el primer paso es transformar la formulación fuerte del problema (8) ya presentada, en la formulación débil (12).

Además de ser el primer paso para lograr la aplicación del MEF, la formulación débil también es básica para el Método Variacional Multiescala (VMS). El VMS constituye el modelo turbulento implementado en el algoritmo por parte de los investigadores y sus bases se pueden consultar en [11].

La formulación débil de un determinado problema definido mediante ecuaciones diferenciales, también denominada formulación variacional, es una forma alternativa de expresar dichas ecuaciones diferenciales en forma integral, dando lugar a expresiones abordables mediante los métodos del álgebra lineal. A cambio, se relajan las imposiciones realizadas originalmente por la formulación fuerte sobre las funciones solución, como rebajar el orden de derivabilidad de las mismas. Por tanto, podrán obtenerse soluciones que satisfagan la formulación débil pero no la fuerte.

Para obtener la formulación débil del problema, (12), se parte de la formulación fuerte (8) y se realiza el producto escalar por una función vectorial ( $\mathbf{w}$ ) de la misma dimensión que  $\mathbf{u}$  para la primera ecuación y por una función escalar ( $q$ ) para la segunda, posteriormente se integran ambas en todo el dominio del problema ( $\Omega$ ), obteniendo (9). Puesto que ambas expresiones son iguales a cero, ninguna de estas operaciones ha alterado la validez de las igualdades y se siguen cumpliendo  $\forall (\mathbf{w}, q)$ .

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{w} \, d\Omega + \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{w} \, d\Omega + \int_{\Omega} \nabla p / \rho \cdot \mathbf{w} \, d\Omega - \int_{\Omega} \nu \nabla^2 \mathbf{u} \cdot \mathbf{w} \, d\Omega = \mathbf{0} \\ \int_{\Omega} (\nabla \cdot \mathbf{u}) \, q \, d\Omega = 0 \end{array} \right. \quad (9)$$

El objetivo detrás de estas operaciones es predisponer los términos de la formulación fuerte para una integración por partes, reduciendo de este modo el grado de la ecuación diferencial. Tras ello todos los términos involucrados serán de primer orden, algo imprescindible para la aplicación del MEF.

El único término diferencial de grado superior a uno en (9) es el difusivo, por lo que sólo es necesario aplicar la integración por partes sobre él;

$$\begin{aligned} \int_{\Omega} \nu \nabla^2 \mathbf{u} \cdot \mathbf{w} \, d\Omega &= \int_{\Gamma_h} \nu \nabla \mathbf{u} \cdot \mathbf{n} \cdot \mathbf{w} \, d\Gamma_h - \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{w} \, d\Omega \\ &= \int_{\Gamma_h} \mathbf{h} \cdot \mathbf{w} \, d\Gamma_h - \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{w} \, d\Omega \end{aligned} \quad (10)$$

Obteniendo de este modo la formulación débil del problema (12), expresada abreviando las integrales del producto escalar de dos funciones a lo largo de un determinado dominio  $\Psi$  tal y como se recoge en (11);

$$\int_{\Psi} \mathbf{a} \cdot \mathbf{b} \, d\Psi = (\mathbf{a}, \mathbf{b})_{\Psi} \quad (11)$$

$$\left\{ \begin{array}{l} \left( \frac{\partial \mathbf{u}}{\partial t}, \mathbf{w} \right)_{\Omega} + ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{w})_{\Omega} + \left( \frac{\nabla p}{\rho}, \mathbf{w} \right)_{\Omega} + \nu (\nabla \mathbf{u}, \nabla \mathbf{w})_{\Omega} = (\mathbf{h}, \mathbf{w})_{\Gamma_h} \quad \text{en } \Omega \\ (\nabla \cdot \mathbf{u}, q)_{\Omega} = 0 \quad \text{en } \Omega \\ \mathbf{u} = \mathbf{g} \quad \text{en } \Gamma_g \\ \nu \nabla \mathbf{u} \cdot \mathbf{n} = \mathbf{h} \quad \text{en } \Gamma_h \end{array} \right. \quad (12)$$

El método de elementos finitos se nutre de esta formulación (12) para generar el sistema de ecuaciones algebraico lineal. La base de este método, explicada de

---

una forma muy breve y generalista, consiste en dividir el dominio del problema  $\Omega$  en elementos geométricos simples, como cuadriláteros o triángulos, cuyos vértices se denominan nodos (este proceso de discretización también se denomina mallado). Además, las funciones solución se consideran sumatorios de funciones aproximación ( $N_i$ ) cuya principal característica consiste en adoptar el valor unidad para las coordenadas del nodo  $i$  y el valor nulo para el resto de nodos, ver (13). Concretamente el algoritmo empleado en este trabajo utiliza funciones aproximación bilineales para los elementos cuadriláteros.

$$(u, v, p) \simeq \sum_{i=1}^n a_i N_i \quad (13)$$

Donde  $a_i$  corresponde a los coeficientes escalares incógnita del incipiente sistema algebraico y  $n$  simboliza el número de nodos totales.

A partir de este contexto, empleando la circunstancia de que la formulación débil se cumple  $\forall (\mathbf{w}, q)$ , se generan  $n$  variaciones de la misma, con una dupla de funciones de ponderación  $(\mathbf{w}, q)$  diferente para cada una de ellas. Con mucha frecuencia, y también en el caso de este algoritmo, se emplean como funciones ponderación las mismas que para las funciones aproximación. De este modo, e integrando numéricamente todos los términos se alcanza un sistema de ecuaciones algebraico lineal cuyo vector incógnita es el de coeficientes de las funciones de aproximación ( $a_i$ ).

Para un mayor detalle de las bases de programación del MEF en el solver se invita a consultar [7].

### 2.3. LES vs DNS. Escalas de Kolmogórov

Tal y como se ha mencionado en el apartado 1, el algoritmo que conforma la base del presente trabajo está programado en LES (*Large Eddy Simulation*), es decir, resuelve únicamente las escalas mayores de un problema y modeliza las escalas más pequeñas, las escalas disipativas.

Evidentemente el LES es un procedimiento más inexacto que el enfoque DNS (*Direct Numerical Simulation*). La resolución numérica directa de las ecuaciones de Navier-Stokes es la manera más evidente y precisa de predecir un flujo turbulento. Se resuelven todas las escalas espaciales y temporales del flujo turbulento sin promediados o aproximaciones, los únicos errores proceden de las discretizaciones numéricas. La idea es tan sencilla como, por lo desorbitando de su coste computacional, difícil de llevar a la práctica. Las escalas de refinamiento que se deben alcanzar para poder simular las escalas disipativas de estos flujos generan un coste computacional inabarcable.

Es precisamente el dimensionamiento de estas escalas disipativas  $\eta$  (escalas de Kolmogórov), como argumento para justificar el enfoque LES en detrimento del más exacto DNS, el tema que se extiende a continuación.

En LES, la idea es la siguiente. Los grandes remolinos (*eddys*), de tamaño  $l$ , presentes en un determinado flujo, sujetos a fuerzas inerciales, se fragmentan rápidamente traspasando su energía a los vórtices menores. De hecho, la esperanza de vida de un remolino típico es más bien corta, del orden de su tiempo de rotación,  $l/u$ . Por supuesto, los remolinos más pequeños también son transitorios y pronto transmiten su energía hacia estructuras incluso más pequeñas. Así, en cada instante, hay una cascada continua de energía desde la escala más grande hacia la más pequeña.

Crucialmente, la viscosidad no forma parte en esta cascada puesto que si el valor de  $Re = ul/\nu$  es elevado, como ocurre en los remolinos de mayor escala, las tensiones viscosas actuando en dichos remolinos son despreciables. Esto es también cierto para su descendencia y todo el proceso es esencialmente dirigido por fuerzas inerciales. La cascada se detiene, sin embargo, cuando el tamaño del remolino llega a ser tan pequeño que el  $Re$  asociado al tamaño del vórtice, decrece en los más pequeños hasta el orden de la unidad. En este punto, las fuerzas viscosas llegan a ser significativas y la disipación empieza a ser importante. Por tanto, la energía es disipada sólo en las etapas finales de este proceso, cuando las estructuras son tan finas que el  $Re$  es del orden de la unidad.

Para determinar el orden de magnitud de la escala más pequeña en un flujo turbulento y por tanto, estimar el coste computacional del DNS y su viabilidad, se puede seguir el sencillo procedimiento mostrado a continuación; identificando a  $u$  y a  $v$  con las velocidades medias asociadas a los remolinos más grandes y más pequeños respectivamente del flujo en cuestión y a  $l$  y  $\eta$  con las escalas de longitud de las estructuras más grandes y pequeñas respectivamente, la energía por unidad de masa transmitida por parte de los remolinos de mayor tamaño al comienzo de la cascada es del orden de  $u^2$  y puesto que el tiempo de vida medio de los mismos ronda su tiempo de rotación  $l/u$ , la potencia por unidad de masa transmitida a las escalas inferiores se puede estimar como;

$$\Pi \approx u^2/(l/u) = u^3/l \quad (14)$$

A su vez, la potencia disipada en las escalas menores se puede dimensionar mediante la expresión siguiente (obtención detallada en el apartado 2.1.4 de [2], donde  $\nu$  representa la viscosidad cinemática del fluido involucrado y  $v$  la velocidad característica de los vórtices pequeños);

$$\varepsilon \approx \nu(v^2/\eta^2) \quad (15)$$

Cuando las condiciones del flujo son estables y las medias de las variables mantienen un valor constante en el tiempo, la potencia transmitida (14) debe coincidir exactamente con la potencia disipada en las escalas más pequeñas (15) puesto que no puede existir acumulación de energía en escalas intermedias sin la alteración del régimen estacionario. De este modo, igualando (14) y (15) se obtiene una primera

ecuación que permite relacionar los parámetros a estimar ( $v$  y sobre todo  $\eta$ ) con los parámetros conocidos ( $u$  y  $l$ ).

También es conocido que las escalas más pequeñas, y la consecuente interrupción de la cascada de energía, se alcanzan cuando aparecen los efectos disipativos generados por los esfuerzos viscosos. En ese instante, el  $Re$  basado en  $v$  y  $\eta$  es del orden de la unidad.

$$Re_\eta = \frac{v\eta}{\nu} \approx 1 \quad (16)$$

Por tanto, del sistema formado por (16) y la igualdad entre (14) y (15) es sencillo extraer las siguientes relaciones entre la escala disipativa ( $v$  y  $\eta$ ) y la del problema global ( $u$  y  $l$ ), donde  $Re$  está basado en los remolinos de mayor escala;

$$\eta \approx l Re_{l,u}^{-3/4} \quad (17)$$

$$v \approx u Re_{l,u}^{-1/4} \quad (18)$$

Las escalas  $\eta$  y  $v$  son las llamadas *microescalas de Kolmogorov* de turbulencias, mientras que  $l$  y  $u$  corresponde a la *escala integral*. Es preciso realizar, como primera observación, que las escalas disipativas en un flujo siempre se reducen a medida que aumenta el  $Re$  del flujo global, es decir, a medida que “se torna más turbulento el flujo”.

En base a las microescalas de Kolmogórov, es sencillo obtener la información necesaria para demostrar que el enfoque DNS empleado en la resolución de flujos turbulentos es inviable en aplicaciones prácticas puesto que supone un coste computacional desmedido e inasumible para la tecnología actual. Mediante (17) y (18) se puede conocer el orden de magnitud del tamaño de la malla del dominio espacial y de los incrementos temporales que debe simular el solver para reproducir fielmente el flujo que acontece en la realidad de un problema concreto.

La gran mayoría de flujos con interés ingenieril albergan un  $Re$  en el abanico ( $10^5, 10^8$ ), lo que irremediablemente demanda mallas con una densidad de nodos desorbitada y un ínfimo paso temporal entre iteraciones para el DNS. Esta es una de las principales razones por las que el solver empleado en este trabajo está programado en LES, ya que es el enfoque que mejor equilibrio aporta, teniendo en cuenta precisión y tiempo de computación.



---

### 3. Validación del modelo turbulento

Una vez expuestos los conceptos teóricos básicos que fundamentan el algoritmo y antes de comenzar a emplear dicho código en aplicaciones tecnológicas, un proceso imprescindible es la fase de verificación de las implementaciones realizadas. En esta fase, el objetivo principal es encontrar las posibles discrepancias con la realidad, las debilidades o los aspectos a mejorar para ayudar a los investigadores a continuar evolucionando el código. También es importante encontrar las limitaciones de aplicación para no abordar problemas concretos que estén fuera del alcance actual del algoritmo.

Para la búsqueda de puntos de referencia y la realización de verificaciones, se escoge el problema denominado LDC (*Lid Driven Cavity*) por los amplios estudios realizados sobre él a lo largo de los años de evolución de la fluidodinámica computacional, y en consecuencia, por su abundante recopilación de resultados computacionales contrastados experimentalmente.

#### 3.1. Fundamentos LDC

La geometría del problema es simple y bidimensional, y las condiciones de contorno también son sencillas. El caso estándar consiste en un fluido contenido en un dominio cuadrado (cuya longitud de lado suele ser la unidad) con condiciones de contorno de Dirichlet en todas las aristas; tres de ellas estáticas (condiciones de contorno homogéneas) y una móvil con velocidad tangente al contorno y generalmente de módulo unidad. Además, el punto de la esquina inferior izquierda se suele adoptar como punto de referencia para el campo de presiones asignándole un valor nulo. Ver figura 2. Los aspectos térmicos resultan ser despreciables y por ello tanto el campo de temperatura como las condiciones de contorno térmicas son irrelevantes.

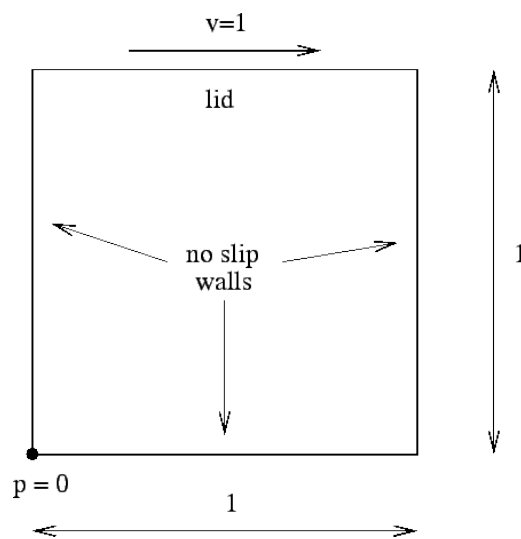


Figura 2: Condiciones de contorno y geometría del problema LDC

Para caracterizar el fluido interior es suficiente con definir los valores numéricos de la densidad ( $\rho$ ) y de la viscosidad dinámica ( $\mu$ ) del mismo. Puesto que los aspectos térmicos son irrelevantes, no es necesario definir el valor del calor específico. El número adimensional más relevante e influyente en este problema es el de Reynolds ( $Re$ ) definido de la forma expuesta en (19), donde  $L$  se corresponde a la longitud de las aristas y  $u$  a la velocidad horizontal del lado móvil.

$$Re = \frac{\rho Lu}{\mu} = \frac{Lu}{\nu} = \frac{1}{\nu} \quad (19)$$

El valor que adquiera  $Re$  determinará las características más importantes del flujo; el régimen (laminar o turbulento), el campo vectorial de velocidades, el campo escalar de presiones, el número y la localización de los torbellinos, las zonas de recirculación, etc. . .

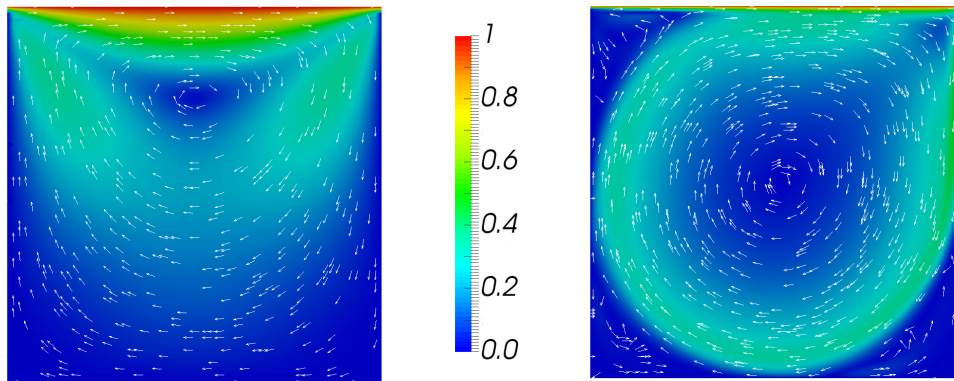
De todas las características del flujo determinadas por  $Re$ , la más relevante para lograr el objetivo aquí perseguido, es el tipo de régimen. Por ello es de vital importancia conocer el intervalo de valores a partir de los cuales el flujo se torna turbulento. De acuerdo con Kosseff y Street [6], dicho intervalo de  $Re$  corresponde a (6.000 , 8.000).

Expuestos los fundamentos más básicos del LDC, se muestran en el apartado 3.2 los resultados y las comparativas que pretenden validar el algoritmo modificado . En la página 25 se detalla con qué malla se ha realizado cada simulación y a lo largo del apartado 3.3 se argumentan dichas elecciones con un breve estudio de sensibilidad referido al número de elementos y su distribución.

La motivación del mencionado estudio de sensibilidad radica en el hecho de que, a pesar de su simple geometría, el flujo de la cavidad conserva una rica física que se manifiesta mediante múltiples vórtices giratorios y regiones de recirculación en las esquinas de la cavidad, íntimamente relacionados con el número de Reynolds y susceptibles de alterar los resultados de la simulación si no se resuelvan con una malla adecuada.

## 3.2. Resultados y comparativas LDC

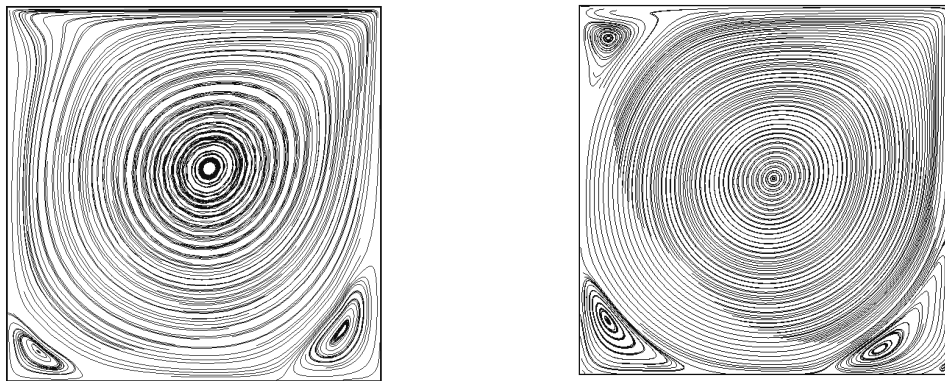
Con el objetivo de generar una idea intuitiva del flujo que se desarrolla en este problema se añaden las ilustraciones 3, 4 y 5 que representan visualmente resultados obtenidos con el solver para dos valores de  $Re$ . A la izquierda de las tres figuras se muestran los resultados correspondientes a un flujo laminar ( $Re = 1,000$ ) y a la derecha de las mismas, a un flujo turbulento ( $Re = 10,000$ ).



(a)  $Re = 1.000$

(b)  $Re = 10.000$

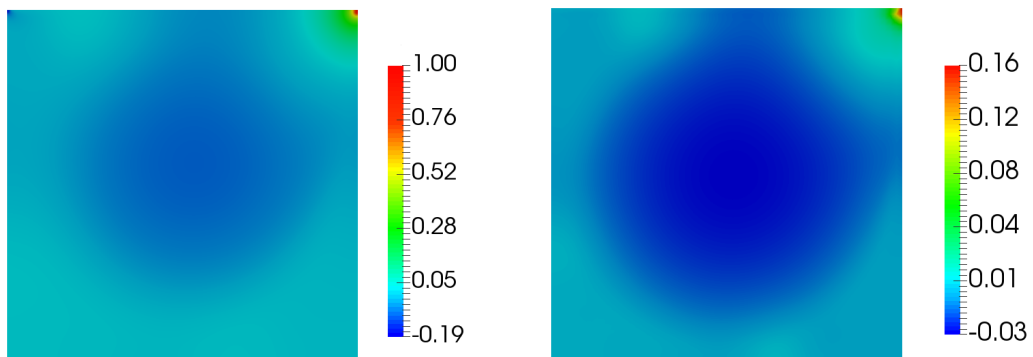
Figura 3: Módulo del vector velocidad. LDC



(a)  $Re = 1.000$

(b)  $Re = 10.000$

Figura 4: Líneas de corriente (*Streamlines*). LDC



(a)  $Re = 1.000$

(b)  $Re = 10.000$

Figura 5: Campo de presión. LDC

Una vez visualizados los resultados cualitativamente, es imprescindible compararlos con los datos bibliográficos cuantitativos. Para orquestar dicho contraste cuantitativo se escogen los perfiles de velocidad a lo largo de los ejes de simetría de la cavidad. Tal y como se muestra en la figura 6, el perfil de velocidad horizontal a través del eje de simetría vertical,  $u(y)$ , y el perfil de velocidad vertical a través del eje de simetría horizontal,  $v(x)$ .



Figura 6: Lugares geométricos donde se obtienen los perfiles de velocidad. LDC

Los perfiles patrón asimilados como representación consistentemente fiel de la realidad han sido tomados de contrastados artículos de investigación en fluidodinámica computacional, concretamente, de [5] para simulaciones en régimen laminar y de [12] para régimen turbulento. Evidentemente, la tendencia de los resultados propios a estos perfiles patrón se considerará un factor de éxito y un argumento para validar las modificaciones del algoritmo.

Las simulaciones propias reflejadas en el presente apartado se han realizado con la malla óptima, detallada, representada y justificada su elección en el apartado 3.3. Concretamente se incluyen resultados obtenidos para diferentes valores de  $Re$ , en primer lugar, correspondientes al régimen laminar (100, 400 y 1.000, figuras 7, 8 y 9) para verificar que las modificaciones realizadas en pos de predecir los flujos turbulentos no alteran la validez de los resultados laminares y a continuación, las comparativas más relevantes, para  $Re$  elevados correspondientes inequívocamente a régimen turbulento (10.000, 20.000 y 50.000, figuras 10, 11 y 12). En todas las figuras mostradas en las páginas 17 y 18, los resultados propios se representan con las curvas de color negro.

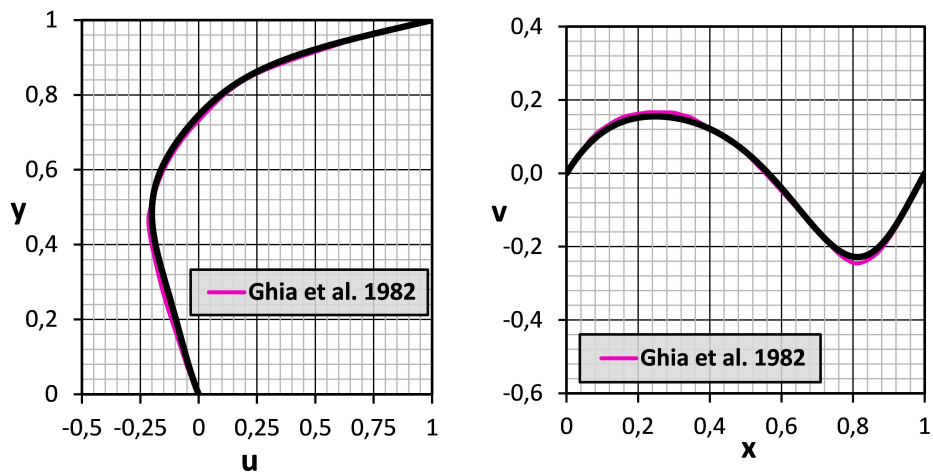


Figura 7: Comparativa resultados LDC (Re = 100)

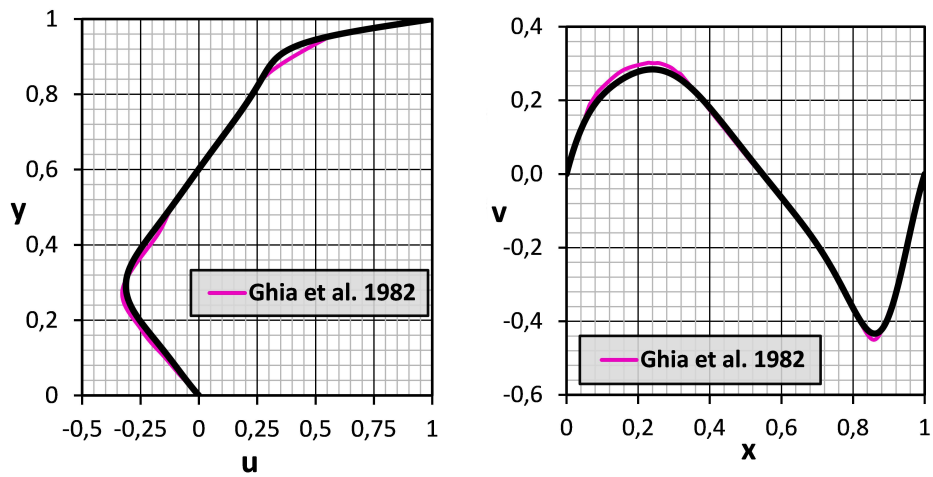


Figura 8: Comparativa resultados LDC (Re = 400)

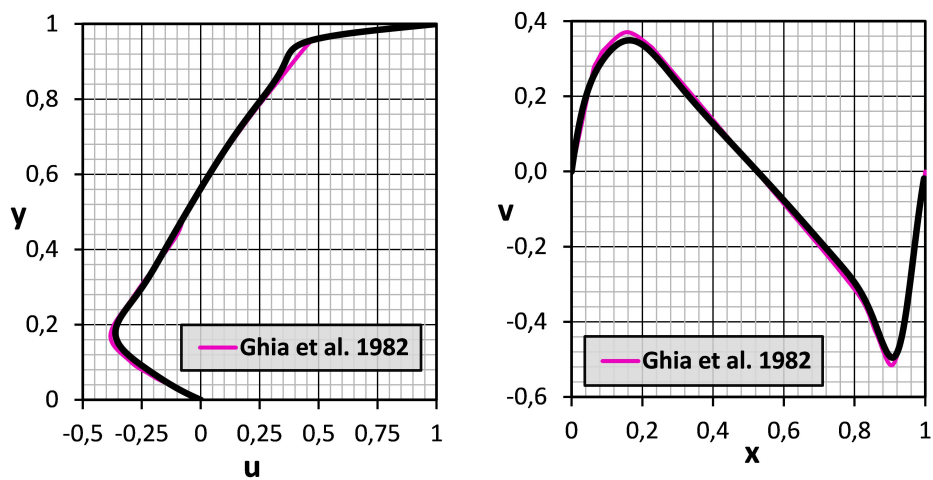
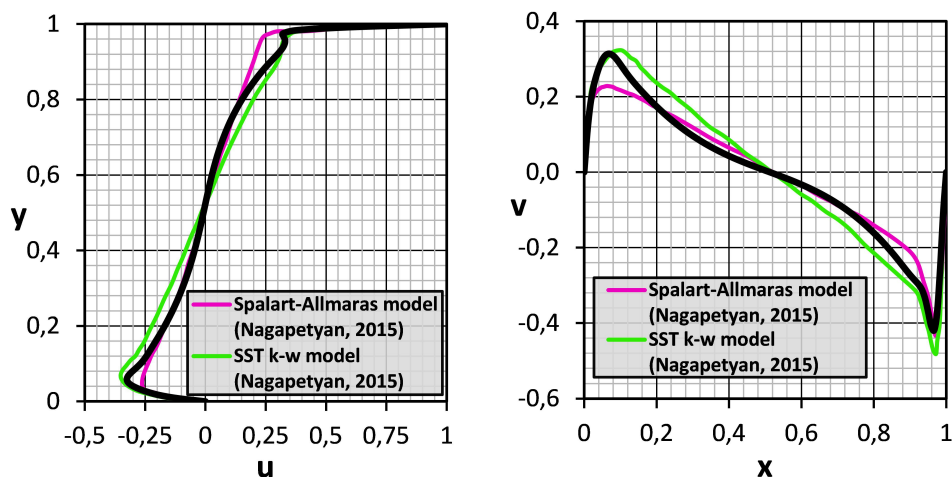
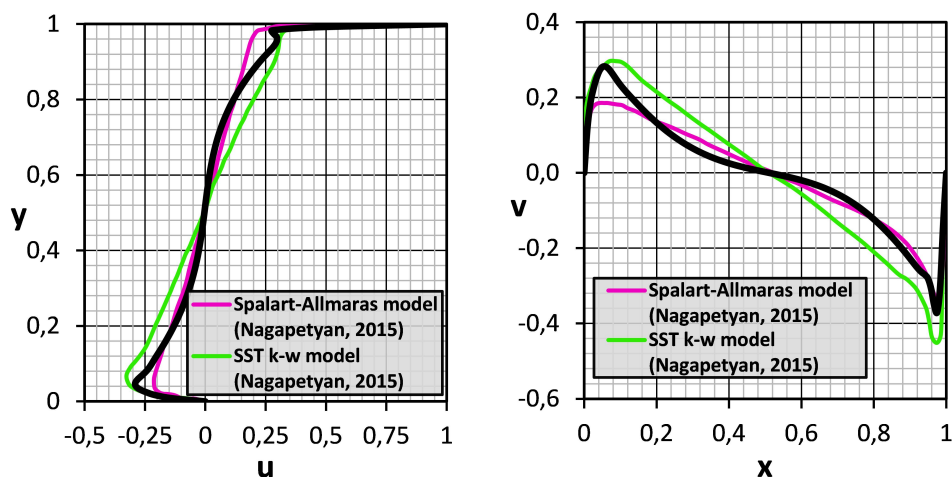
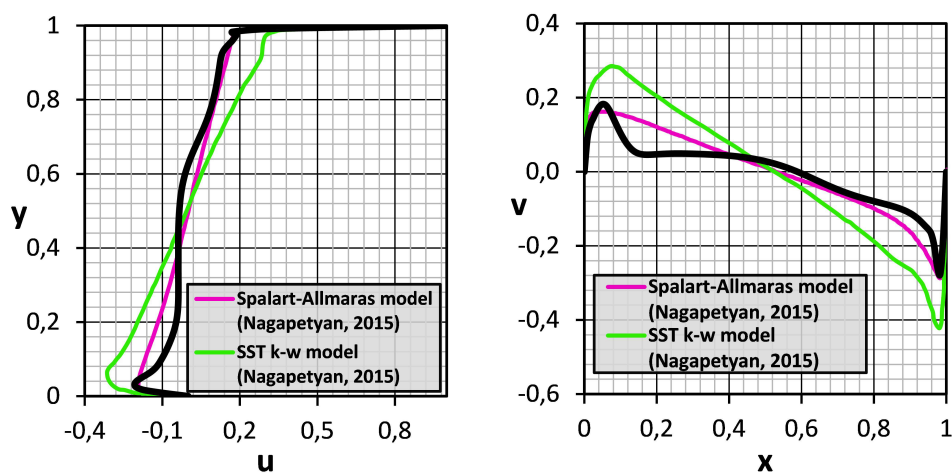


Figura 9: Comparativa resultados LDC (Re = 1.000)

Figura 10: Comparativa resultados LDC ( $Re = 10.000$ )Figura 11: Comparativa resultados LDC ( $Re = 20.000$ )Figura 12: Comparativa resultados LDC ( $Re = 50.000$ )

Las primeras valoraciones son muy positivas. Los resultados de las simulaciones son sin duda coherentes y próximos a las referencias adoptadas para todo el abanico de  $Re$ , tanto en régimen laminar, como en el turbulento.

Llegados a este punto, con el objetivo de cuantificar la calidad de los resultados expuestos en las figuras 7 - 12, es muy interesante plasmar numéricamente el error cometido con respecto a los perfiles de velocidad en los ejes de simetría tomados como referencias. También es relevante averiguar y cuantificar la mejora de precisión del solver con respecto a la versión anterior del algoritmo sin términos turbulentos incorporados. Para lograr ambos propósitos se propone el cálculo del error absoluto medio mediante (20).

$$\bar{E}_{abs} = \frac{1}{2} \left( \frac{1}{L} \int_0^L |v(x)_{ref} - v(x)| dx + \frac{1}{L} \int_0^L |u(y)_{ref} - u(y)| dy \right) \quad (20)$$

La expresión (20) es viable únicamente en caso de disponer de las funciones analíticas  $v(x)$  y  $u(y)$ . Puesto que tanto los resultados propios como los de las referencias se articulan mediante parejas de valores numéricos, es imprescindible discretizar la integración. Dicho proceso, aproximando las integrales por sumatorios discretos de regiones trapezoidales, transforma (20) en (21).

$$\begin{aligned} \bar{E}_{abs} \simeq & \frac{1}{2L} \sum_{i=1}^{n-1} \frac{1}{2} (|v_{i+1}^{ref} - v_{i+1}| + |v_i^{ref} - v_i|)(x_{i+1} - x_i) \\ & + \frac{1}{2L} \sum_{i=1}^{n-1} \frac{1}{2} (|u_{i+1}^{ref} - u_{i+1}| + |u_i^{ref} - u_i|)(y_{i+1} - y_i) \end{aligned} \quad (21)$$

La expresión (21) es fácilmente programable en un algoritmo auxiliar de post-procesado, cuyas entradas son las parejas de valores numéricos correspondientes a los perfiles de velocidad referencia y los propios. La totalidad de las líneas de este algoritmo en lenguaje *Fortran95* se incluyen en el anexo B.

El error absoluto calculado mediante (21), permite visualizar la exactitud de los resultados en función de los diferentes valores de  $Re$  y cuantificar la evolución del solver mediante la comparación de las funciones de *Error absoluto (Re)* del código modificado y del original. Ver figura 13.

En la figura 13 se puede observar que la introducción de términos turbulentos en el seno del código no altera la precisión que ya poseía anteriormente el mismo en la simulación de flujos laminares. Además, es un signo muy positivo que la divergencia comience a ser apreciable después de los valores de  $Re$  para los que Kosseff y Street [6] constatan la aparición del régimen transitorio (6.000 - 8.000) en el LDC y no antes. Con respecto al régimen turbulento la mejora se acentúa a medida que aumenta  $Re$ . Todos estos signos son reflejo de la buena dirección que, inequívocamente, ha tomado el algoritmo en su persecución del objetivo de predecir flujos turbulentos.

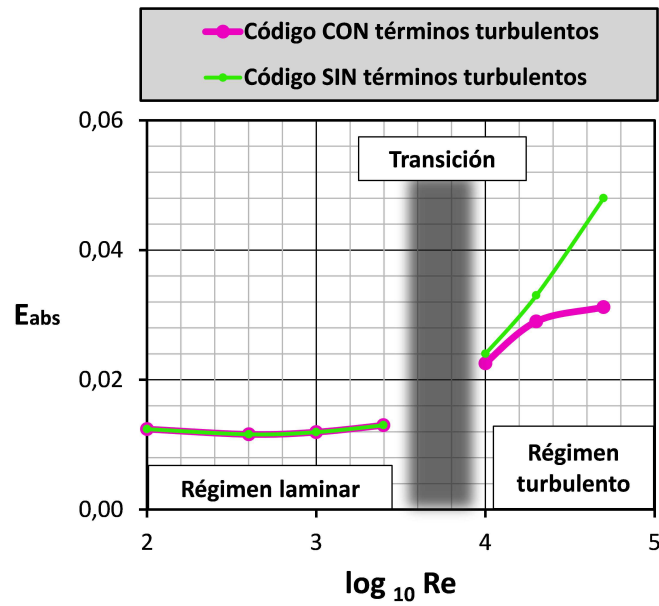


Figura 13: Error absoluto en función de  $Re$  para ambas versiones del solver. LDC

A modo de aclaración se debe mencionar que se ha empleado la misma malla en las simulaciones de ambos códigos para no distorsionar el efecto real de las evoluciones y que las curvas patrón empleadas para el cálculo de los errores absolutos mostrados en la figura 13 son las provenientes de [5] para régimen laminar y de [12] (*Spalart-Allmaras model*) para régimen turbulento.

### 3.3. Análisis de sensibilidad. La influencia de la malla

Como ya se ha mencionado y observado en el apartado 3.2, a pesar de la simple geometría del problema LDC, el flujo en el interior de la cavidad conserva una rica física que se manifiesta mediante múltiples vórtices giratorios y regiones de recirculación en las esquinas íntimamente relacionados con el número de Reynolds. Caracterizar correctamente todos estos patrones relativamente complejos depende en gran medida del tipo de malla y de su resolución. El objetivo principal del presente apartado es estimar la influencia de la malla sobre los resultados y averiguar cuál es la óptima (precisión vs coste computacional) para obtener los resultados de validación expuestos en el apartado 3.2.

En primer lugar, se matiza que todas las mallas propuestas a continuación están compuestas por elementos cuadriláteros estructurados. A partir de ahí, la primera idea, y la más intuitiva y simple, es generar mallas uniformes en todo el dominio. En segundo lugar, se propone recurrir a las escalas de Kolmogórov (introducidas teóricamente en el apartado 2.3) para comprender cualitativamente hacia dónde sería interesante distribuir un mayor número de elementos.



Cabe recordar que las escalas de Kolmogórov representan las dimensiones más pequeñas de los torbellinos (eddys) que pueden existir sin que sean destruidos por la viscosidad. De modo que puede ser interesante refinar lo máximo posible la malla en las regiones del dominio cuyas escalas sean menores y viceversa, aumentar el tamaño de los elementos en las regiones con mayores escalas de Kolmogórov, obteniendo de este modo mallas estructurada pero no uniformes.

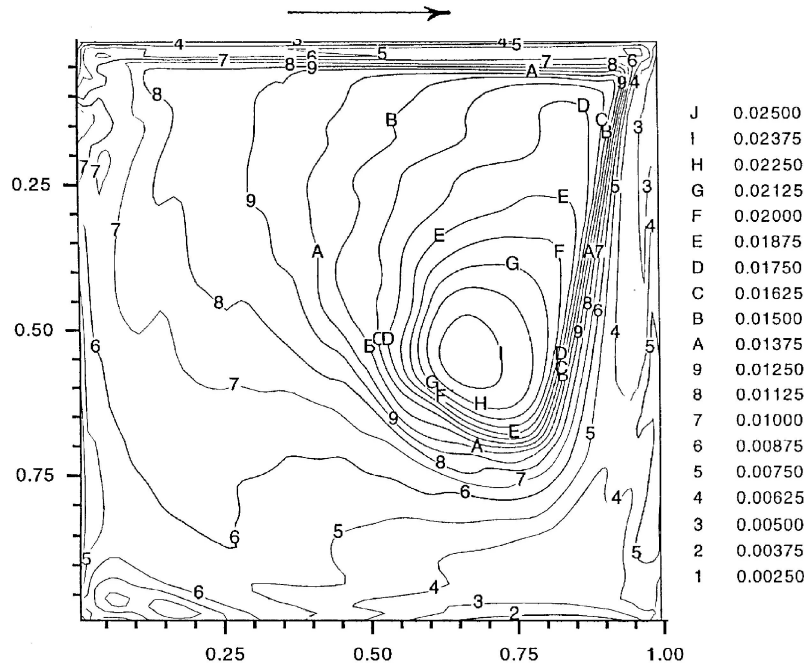


Figura 14: Microescalas espaciales de Kolmogórov ( $\eta$ ) en el dominio del LDC,  $Re = 10.000$ , [8]

A partir de la figura 14 se puede abstraer la idea de que las escalas espaciales de Kolmogórov son menores, a rasgos generales, en las zonas cercanas a las paredes y las esquinas y mayores en la región central. En base a ello la tipología de malla denominada de aquí en adelante como “distribuida” constará de una mayor densidad de elementos en las proximidades del contorno y una menor densidad en el centro del dominio.

A modo de presentación visual se incluye la figura 15, representando la tipología de malla distribuida. Se considera innecesario añadir la tipología uniforme puesto que simplemente es una retícula homogénea conformada por cuadrados idénticos en todo el dominio. Con el objetivo de facilitar la visibilidad y puesto que el cometido de la figura 15 es plenamente ilustrativo, se muestra una malla con resolución mucho más grosera (20x20 elementos) que las realmente empleadas.

Para conocer el orden de magnitud adecuado del número de elementos se recurre a diferentes artículos que estudian la simulación numérica del LDC ([5], [10], [14]...),

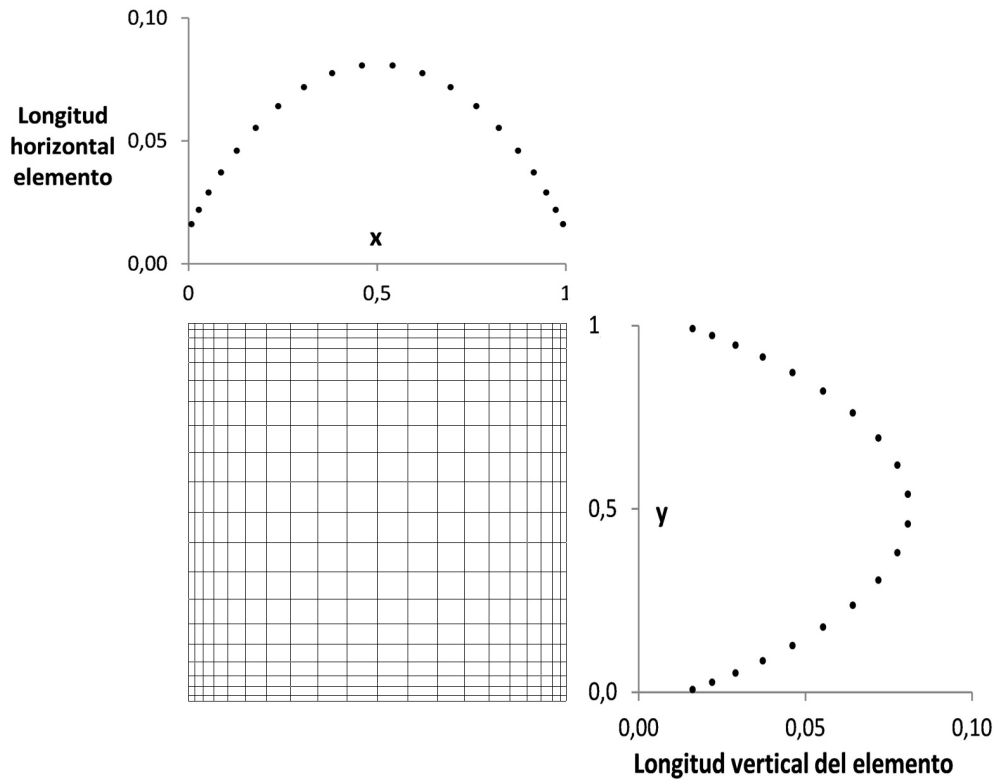


Figura 15: Tipología de malla distribuida

en base a ellos, se generan mallas con una resolución de 60x60, 120x120, 180x180 y 240x240 elementos, tanto para las uniformes como para las distribuidas.

Sin embargo, queda por determinar cuál es el grado de distribución óptimo para la tipología de malla mostrada en la figura 15. Para cuantificar el grado de distribución se emplea el parámetro  $\delta$  definido como el cociente entre la longitud del elemento más grande y la del más pequeño.

$$\delta = \frac{L_{max}}{L_{min}} = \frac{H_{max}}{H_{min}} \quad (22)$$

Nótese que a partir de un valor de  $\delta$  deja de ser eficiente aumentar más la distribución. Aumentar la región ocupada por elementos menores que la media se torna progresivamente más costoso con respecto al tamaño que adquieren los elementos de la región central. En otras palabras, a medida que aumenta  $\delta$  reducir mínimamente los elementos pequeños supone aumentar en gran medida los mayores.

En la gráfica izquierda de la figura 16 se puede visualizar la engañosa circunstancia de que aumentar  $\delta$  incrementa indefinidamente el número de elementos con un tamaño menor que el elemento medio de la malla uniforme con la misma resolución, sin embargo, en la gráfica derecha de la misma figura, se puede observar el hecho

ya comentado de que incrementar la región ocupada por elementos menores que la media empeora en mayor medida la región central. En base a ello se deduce que existe un  $\delta$  óptimo dependiente del número de elementos de las aristas.

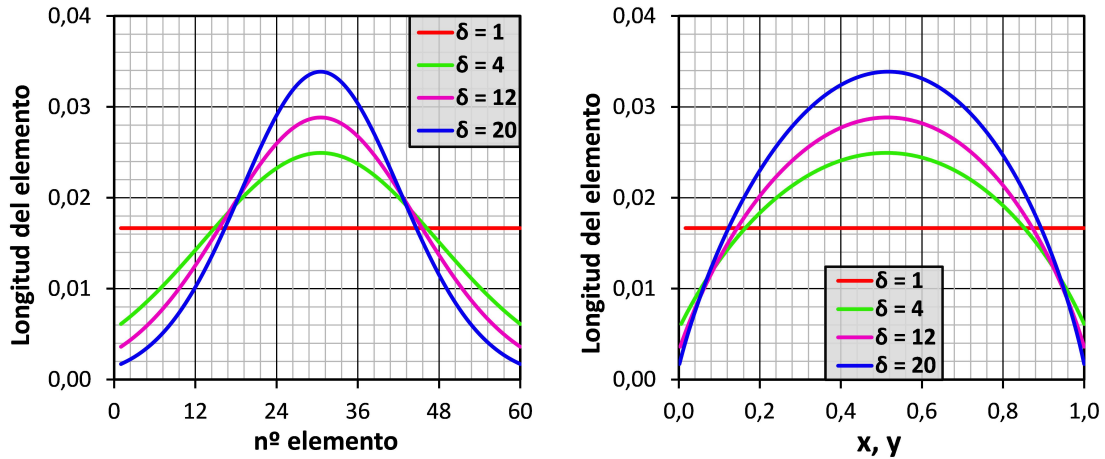


Figura 16: Efecto del parámetro  $\delta$  en la distribución de elementos. Resolución 60x60

Con el objetivo de averiguar el valor de  $\delta_{optimo}$  en función de la resolución se calcula el error absoluto cometido por todas las mallas distribuidas ( $\delta = 8, 4, 2$ ) y la uniforme ( $\delta = 1$ ) según la expresión (21) de la página 19 con respecto a la curva de referencia publicada en [5] para el valor de  $Re = 1.000$  (escogido arbitrariamente por su menor coste computacional con respecto a valores superiores). El compendio de estos resultados se muestra en la figura 17.

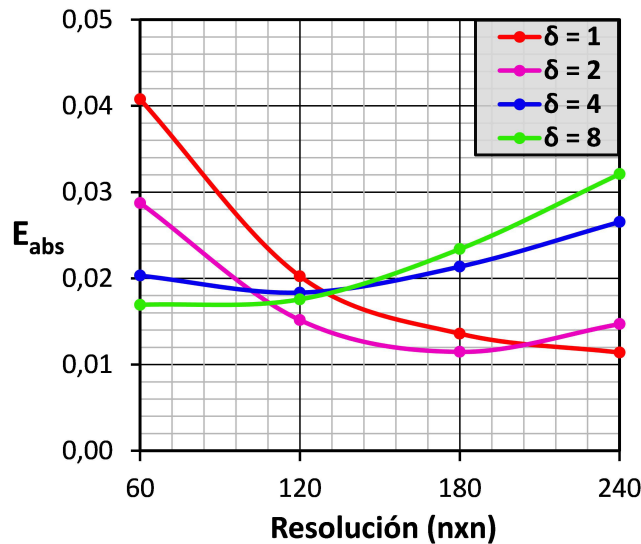


Figura 17: Error absoluto en función de la distribución y la resolución de la malla ( $Re = 1.000$ )

De lo expuesto en la figura 17 se pueden extraer varias conclusiones. Se confirma la ineficacia de aumentar indefinidamente el grado de distribución cuando la resolución es elevada, prevista tras la visualización del efecto del parámetro  $\delta$  en la figura 16 y se obtiene el valor idóneo de  $\delta$  en función de la resolución (ver tabla 1). Además, es relevante mencionar que la tipología uniforme es la única que siempre mejora los resultados al aumentar el número de elementos que la componen.

Resolución	Distribución óptima
60x60	$\delta = 8$
120x120	$\delta = 2$
180x180	$\delta = 2$
240x240	$\delta = 1$ (Uniforme)

Tabla 1: Distribución óptima en función de la resolución de la malla

Finalmente, se está en disposición de completar el objetivo primigenio del presente apartado; averiguar qué malla aporta el mejor equilibrio entre precisión y coste computacional en función del número de Re. Para abordar el análisis final se trabaja con las mallas cuyos parámetros de resolución y distribución se reflejan en la tabla 1.

La representación gráfica de la precisión frente al coste computacional para diferentes valores de Re se adjunta en la figura 18. Para generar esta gráfica se ha cuantificado la precisión de cada simulación de nuevo mediante el error absoluto cometido con respecto a los perfiles patrón de [5] para los valores de Re laminares y de [12] (*Spalart-Allmaras model*) para los turbulentos (calculado siempre mediante la expresión (21) de la página 19). Con respecto al coste computacional se realiza la aproximación heurística de que, para condiciones idénticas de un determinado flujo, el tiempo de resolución es proporcional al número de elementos que conforman la malla empleada en la simulación.

En la figura 18 se observa el aumento de la precisión que se logra a cambio de incrementar el coste computacional (patrón común independiente del número de Re). Sin embargo, la mejora de la precisión se estanca progresivamente, no haciéndolo así el aumento del coste computacional. Para cada valor de Re este estancamiento de la mejora de la precisión se produce en un punto diferente, siguiendo una tendencia cualitativa proporcional; número de elementos óptimo  $\propto$  Re

A modo de conclusión del análisis de sensibilidad se incluye la figura 19 mostrando la malla óptima aproximada a emplear en función del valor de Re, incluyendo resolución y grado de distribución. En base a estas relaciones se han elegido las mallas empleadas en las simulaciones (ver tabla 2) que han dado origen a los resultados de validación del apartado 3.2.

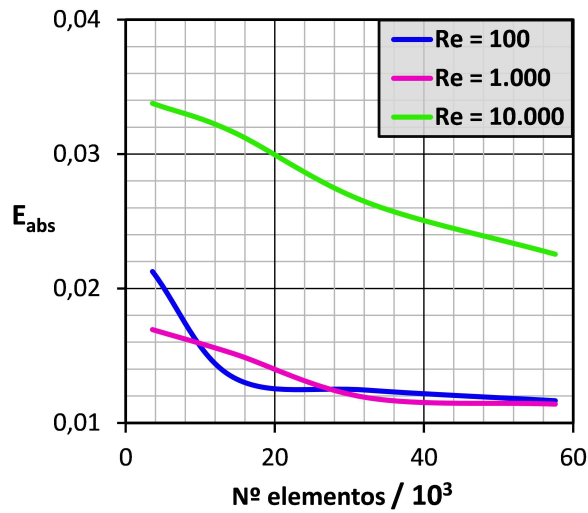


Figura 18: Precisión vs Coste computacional en función de Re

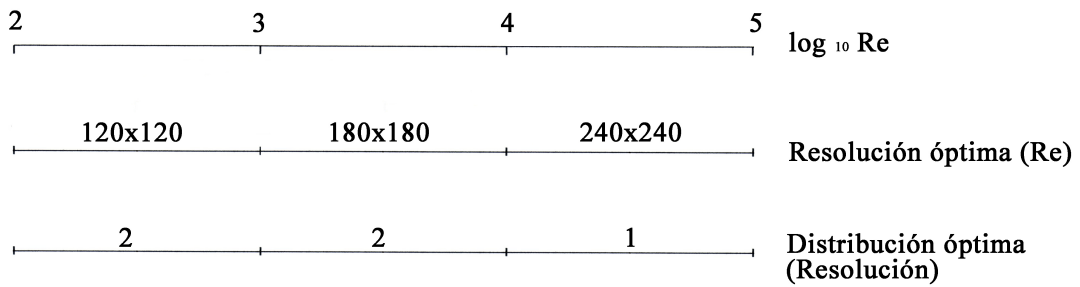


Figura 19: Resolución y distribución de la malla óptima en función de Re

Re	Malla
100	120x120 ( $\delta = 2$ )
400	120x120 ( $\delta = 2$ )
1.000	180x180 ( $\delta = 2$ )
2.500	180x180 ( $\delta = 2$ )
10.000	240x240 ( $\delta = 1$ )
20.000	240x240 ( $\delta = 1$ )
50.000	240x240 ( $\delta = 1$ )

Tabla 2: Mallas empleadas en las simulaciones del apartado 3.2 en función de Re

---

## 4. Simulación de la turbomáquina

Expuestos los conceptos teóricos básicos que fundamentan el algoritmo en la sección 2 y validadas las modificaciones del modelo turbulento en la sección 3 se está en disposición de comenzar a utilizar el solver en aplicaciones tecnológicas.

La aplicación práctica escogida es la simulación del flujo en el interior de una turbomáquina, concretamente, se aborda la simulación de un ventilador centrífugo con álabes curvados hacia adelante (*Forward curved centrifugal fan*) cuyo flujo es turbulento en la realidad. Este tipo de turbomáquinas son ampliamente utilizadas por su buen comportamiento y su simpleza a la hora de ser manufacturadas, además de por su escaso nivel sonoro. Para integrar el caso de estudio en el marco de aplicación del solver es imprescindible generar una simplificación (2-D) del modelo geométrico real.

Comenzando con unas breves pinceladas del método MRF (*Multiple Reference Frames*), empleado en la resolución de flujos que abarcan dominios con velocidades relativas entre ellos, y tras mostrar el diseño del modelo 2-D, se presenta el código de post-procesado, crucial para lograr presentar propuestas de optimización y realizar comparaciones cuantitativas de los resultados. Finalmente se incluyen los resultados más significativos de las simulaciones.

### 4.1. Multiple Reference Frame

Para aplicar el método MRF, en primer lugar, el dominio global de cálculo debe ser dividido en los diferentes subdominios que lo componen, siendo el factor diferenciador la existencia de velocidad relativa entre ellos. En el caso concreto de la turbomáquina que se simula en el presente apartado existen únicamente dos subdominios, uno rotante  $\Omega_t^1$  con una determinada velocidad angular constante  $\omega$ , correspondiente a la región del rotor, y otro fijo  $\Omega_t^0$ , asociado a la voluta. La interfaz entre ambos se denota  $\Gamma_t^1$ . Ver figura 20.

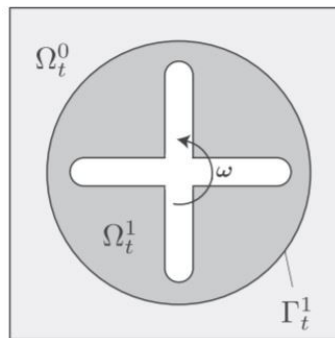


Figura 20: Subdominios e interfaz. Enfoque MRF

El flujo en cada subdominio se resuelve utilizando las ecuaciones del marco de referencia correspondiente. Si un subdominio está acelerado, se resuelve utilizando las ecuaciones de Navier-Stokes en el marco de referencia no inercial solidario al respectivo dominio (en este caso concreto corresponden a (23) y (24) para el subdominio  $\Omega_t^1$ ), obteniendo el campo de velocidades relativas a su referencia,  $\mathbf{u}_R$ ,

$$\rho \left( \frac{\partial \mathbf{u}_R}{\partial t} + (\mathbf{u}_R \cdot \nabla) \mathbf{u}_R + 2\boldsymbol{\omega} \times \mathbf{u}_R + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \right) = \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{u}_R \quad \text{en } \Omega_t^1 \quad (23)$$

$$\nabla \cdot \mathbf{u}_R = 0 \quad \text{en } \Omega_t^1 \quad (24)$$

Cabe señalar que los términos a la izquierda del igual en (23) surgen de la relación entre la aceleración absoluta y la aceleración relativa referida a un marco no inercial (25) donde  $\mathbf{a}_I^0$  corresponde a la aceleración absoluta de la referencia no inercial, simplificada por las particularidades de la presente aplicación;  $d\boldsymbol{\omega}/dt = \mathbf{0}$ ,  $\mathbf{a}_I^0 = \mathbf{0}$  y  $\mathbf{r}_R = \mathbf{r}_I$  ( $\forall t, \Omega_t^1$ ).

$$\rho \mathbf{a}_I = \rho \left( \mathbf{a}_R + \mathbf{a}_I^0 + 2\boldsymbol{\omega} \times \mathbf{u}_R + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_R) + \frac{d\boldsymbol{\omega}}{dt} \times \mathbf{r}_R \right) \quad (25)$$

En caso de que el subdominio sea fijo, las ecuaciones que deben resolverse corresponden a las expresiones ya expuestas en el apartado 2.1 y recogidas de nuevo aquí ((26) y (27) aplicadas en  $\Omega_t^0$ ), válidas para el marco de referencia inercial, obteniendo el campo de velocidades absolutas,  $\mathbf{u}_I$ ,

$$\rho \frac{\partial \mathbf{u}_I}{\partial t} + \rho (\mathbf{u}_I \cdot \nabla) \mathbf{u}_I = \rho \mathbf{f} - \nabla p + \mu \nabla^2 \mathbf{u}_I \quad \text{en } \Omega_t^0 \quad (26)$$

$$\nabla \cdot \mathbf{u}_I = 0 \quad \text{en } \Omega_t^0 \quad (27)$$

Una vez conocidos todos los campos de velocidades relativas en el seno de los subdominios rotantes y el campo de velocidad absoluta del subdominio estático, a la solución en los nodos de los subdominios no inerciales se les agrega vectorialmente la velocidad de arrastre propia de su referencia. De este modo se obtiene una solución de velocidad absoluta continua en todo el dominio  $\Omega_t$ . La relación general es (28) donde  $\mathbf{u}_I^0$  corresponde a la velocidad absoluta de la referencia no inercial, sin embargo, puesto que en el caso de la turbomáquina abordada aquí  $\mathbf{u}_I^0 = \mathbf{0}$  y  $\mathbf{r}_R = \mathbf{r}_I$  ( $\forall t, \Omega_t^1$ ), se aplica (29).

$$\mathbf{u}_I = \mathbf{u}_I^0 + \mathbf{u}_R + \boldsymbol{\omega} \times \mathbf{r}_R \quad (28)$$

$$\mathbf{u}_I = \mathbf{u}_R + \boldsymbol{\omega} \times \mathbf{r} \quad (29)$$

Es relevante destacar lo sensible que es la interfaz  $\Gamma_t^1$ . Dado que los nodos localizados en ella pertenecen a ambos dominios y que durante la computación existe una discontinuidad entre la solución de los subdominios se presenta la problemática de tener dos valores de velocidad distinta para cada uno de estos nodos. Puesto que no es posible que almacenen los dos valores, sólo se guarda el valor del sistema inercial, de este modo se permite que las variables del flujo en una zona se empleen para calcular las del límite del subdominio adyacente.

Una vez señaladas las bases del método, es necesario tener presente que el MRF es claramente una aproximación. En concreto una aproximación de estado estacionario, es decir, la malla permanece fija para el cálculo, lo que equivale a congelar el movimiento de la parte móvil en una posición específica y observar el campo de flujo instantáneo con el rotor en esa posición. Por lo tanto, el MRF a menudo se denomina “enfoco de rotor congelado”.

Pese a ello, este enfoque es capaz de proporcionar un modelo razonable del flujo para muchas aplicaciones cuando las interacciones rotor-estator son relativamente débiles y el flujo es relativamente sencillo en la interfaz. Sin embargo, si es necesario simular realmente los transitorios que pueden ocurrir en interacciones fuertes entre rotor y estator debe emplearse el modelo de malla deslizante (*sliding mesh model*).

Como se especificará en el apartado 4.2, en la aplicación que se aborda en el presente trabajo no existe un estator con álabes, el flujo pasa directamente del rotor a la voluta de la turbomáquina, resultando una mínima interacción rotor-estator. Por tanto, es viable aceptar la aproximación que supone el MRF.

## 4.2. Diseño del modelo 2-D y de su malla. Geometría base

Ya ha sido mencionado en el apartado 2 que el marco de aplicación del algoritmo de elementos finitos se ciñe a flujos bidimensionales e incompresibles, tanto en su versión con el modelo turbulento incorporado como en la anterior. La limitación de flujo bidimensional conlleva la obligación de generar un modelo 2-D del ventilador centrífugo tridimensional real. El objetivo es crear una fiel reproducción de la sección media de la turbomáquina, perpendicular al eje de giro del rotor y en base a los parámetros de diseño empleados en la realidad. Este tipo de ventilador centrífugo no incluye estator con álabes fijos, por lo que la arquitectura constructiva que tiene influencia sobre el flujo en su interior se puede modelar mediante únicamente dos elementos principales; la voluta y el rotor.

Comenzando con la voluta, se ha modelado la pared interior de su sección media en base a la expresión (30), correspondiente a una curva de espiral logarítmica.

$$r(\theta) = r_0 e^{(\theta-\theta_0) \operatorname{tg}(\alpha_v)} \quad (30)$$



El parámetro  $r_0$  simboliza el radio externo del dominio del rotor (52 mm),  $\theta_0$  corresponde al ángulo en radianes donde comienza la espiral y  $\alpha_v$  es el parámetro que controla la apertura de la misma, su valor en esta turbomáquina es de  $7^\circ$ . Las dimensiones de la boca de salida también han sido adoptadas de la geometría real y quedan reflejadas en la figura 21.

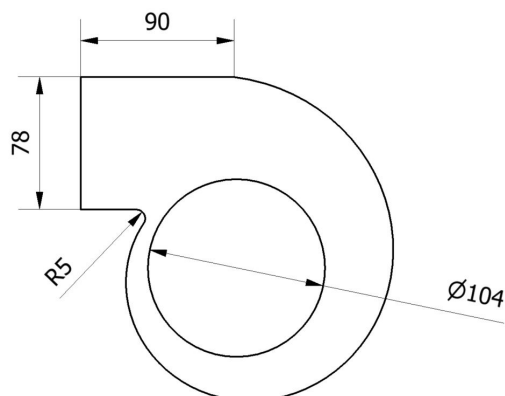


Figura 21: Modelo 2D de la voluta. Cotas en milímetros

Respecto al rotor, lo más relevante es modelar correctamente el álabe, en especial su ángulo de entrada ( $\beta_1 = 90^\circ$ ) y de salida ( $\beta_2 = 150^\circ$ ) puesto que influyen en gran medida en el desarrollo del flujo. Los 35 álbes individuales que componen el rotor se encuentran situados en una franja o corona circular acotada por radios de 40 y 50 mm y repartidos de forma homogénea, separados 10,3 grados cada uno de ellos. Los valores numéricos del radio exterior del dominio del rotor (coincidente con el radio interno de la voluta), del radio interior (su circunferencia equivale a la superficie de entrada del fluido en la turbomáquina) y del espesor de los álbes se muestran en la figura 22.

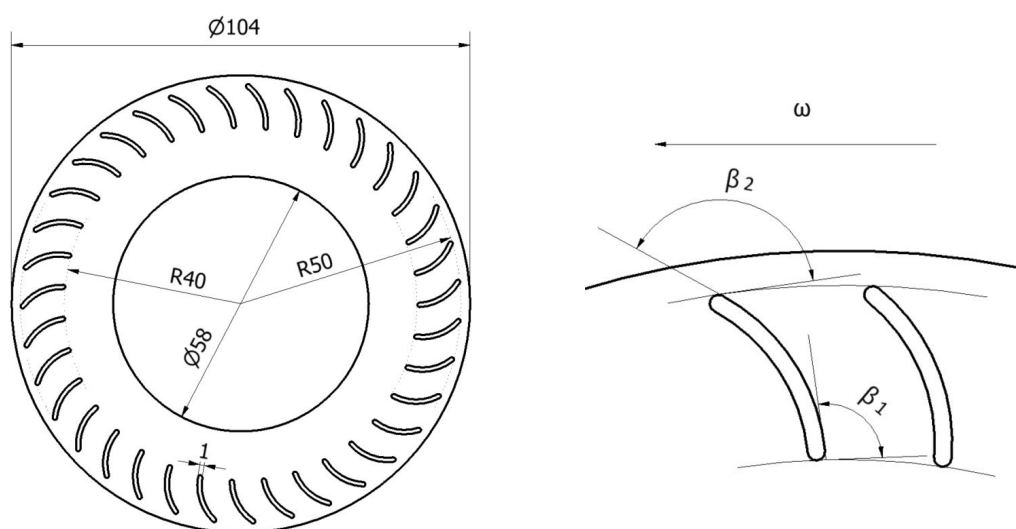


Figura 22: Modelo 2D del rotor. Cotas en milímetros

Una vez definida la geometría del modelo 2D, se pueden concretar el resto de detalles necesarios para completar el pre-proceso y estar en disposición de comenzar a lanzar simulaciones. Queda por estipular las condiciones de contorno a imponer, los parámetros del fluido y el mallado de ambos subdominios.

Respecto a las condiciones de contorno, se impone presión uniforme y de valor nulo a lo largo de la arista que modela la superficie de salida del fluido (color rojo en la figura 23), condición de no deslizamiento en todo el contorno de la voluta y sobre los álabes (color azul en la misma figura), lo que se traduce en  $\mathbf{u}_I = (u_I, v_I) = \mathbf{0}$  para la pared de la voluta y en  $\mathbf{u}_R = (u_R, v_R) = \mathbf{0}$  para la superficie de los álabes. Finalmente, en la circunferencia que modela la superficie de entrada del fluido a la turbomáquina (color verde en la figura 23), se impone velocidad absoluta en dirección radial de módulo uniforme a lo largo de toda ella;  $|\mathbf{u}_I| = Q/2\pi br_{ent} = U \equiv \mathbf{u}_I = (U\cos\alpha, U\sin\alpha)$  con  $\alpha = \text{arctg}(c_y/c_x)$ . El parámetro  $b$  simboliza la anchura tridimensional de la turbomáquina, siendo su valor real 40 mm.

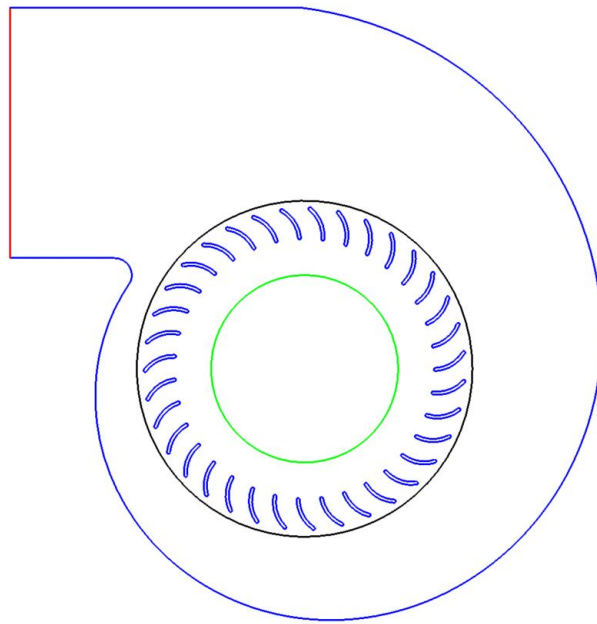


Figura 23: Condiciones de contorno sobre el conjunto del modelo 2D

El fluido que se pretende simular en el ventilador es aire atmosférico a 20°C de temperatura, bajo las hipótesis de flujo incompresible y despreciando los efectos térmicos, los únicos parámetros del fluido necesarios para resolver las ecuaciones de Navier-Stokes en todo el dominio son la densidad y la viscosidad de dicho fluido (aire atmosférico) a dicha temperatura (20°C);

$$\rho = 1,22 \text{ Kg/m}^3 \quad \mu = 1,835 \cdot 10^{-5} \text{ Ns/m}^2$$

Finalmente, respecto a la malla, se pretende emplear la misma en todas las simulaciones de 4.4 y 4.5 con el objetivo de eliminar su influencia sobre las comparativas que se realizan en dichos apartados. Para concretar los detalles, se parte de las siguientes premisas.

Las mallas de voluta y rotor se deben generar por separado para poder aplicar el método MRF. Además se debe predisponer que los elementos de la interfaz entre ambos dominios sean coincidentes, puesto que, en el contexto de dicho método, esta coincidencia es de vital importancia para lograr la continuidad de la solución entre el dominio inercial y el no inercial y la comunicación correcta entre ambas mallas. Por tanto, el tamaño de elemento en la circunferencia externa del rotor debe ser igual que el de la circunferencia interna de la voluta.

La malla del rotor debe ser capaz de alterar lo menos posible la geometría original de los álabes, por ello se refina en mayor medida, con respecto a la media del resto de la malla, la región próxima a los mismos con el objetivo de reducir todo lo posible el error cordal. Ver figura 24.

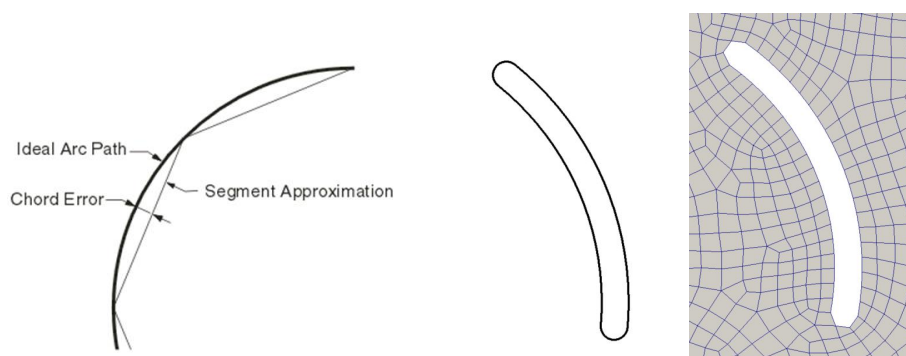


Figura 24: Control del error cordal en el mallado de los álabes

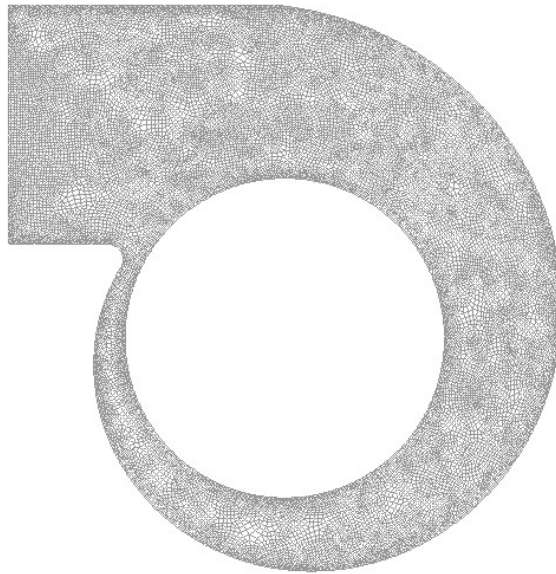
Respecto a la voluta, partiendo de la imposición de tamaño de elemento en su interior por parte del rotor, se decide introducir un mayor número de elementos cerca del contorno exterior con la idea de capturar lo mejor posible los efectos de pared que se generen y sacrificando resolución en regiones intermedias con un flujo más uniforme.

Cabe añadir que únicamente se contempla el empleo de elementos cuadriláteros, obligadamente desestructurados debido a la sinuosa geometría, para el mallado de ambos dominios.

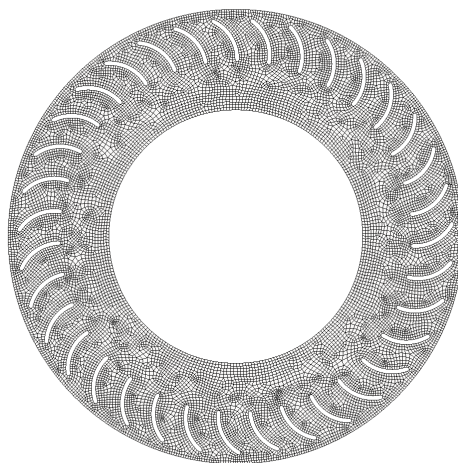
En este punto sólo falta por concretar el tamaño medio de los elementos y su número global aproximado que permita alcanzar un equilibrio adecuado entre la calidad de la solución y el coste computacional. Se recurre al análisis de sensibilidad recogido en [13] para un modelo bidimensional de una turbomáquina de idéntica

arquitectura constructiva y dimensiones muy similares, en el cuál se alcanza la conclusión de que la solución comienza a no mostrar variaciones significativas al emplear elementos con longitud media de arista igual a 1 mm, lo que supone la discretización de ambos dominios con un total de 30.000/40.000 elementos.

En base a las premisas mencionadas anteriormente, se genera la dupla de mallas mostrada en la figura 25. Con elementos de 0,5 mm sobre el contorno de los álabes, 1 mm en la pared externa de la voluta e imponiendo un tamaño medio para el resto de 1,8 mm, se obtiene un total de 33.158 elementos, repartidos en 16.640 de la voluta y 16.518 del rotor.



(a) Malla de la voluta



(b) Malla del rotor

Figura 25: Mallas de la geometría base del modelo 2D

### 4.3. Código de post-procesado

Una vez descrito el pre-proceso en el apartado 4.2 y antes de mostrar los resultados de las simulaciones en el 4.4, es preciso comentar brevemente la forma en la que se extrae la mayor información posible de dichos resultados vertidos por el solver (campos de presión y de velocidad), o en otras palabras, comentar cómo se post-procesa la información.

Con el objetivo de realizar comparaciones entre simulaciones y de generar propuestas de optimización se desarrolla íntegra y originalmente el algoritmo de post-procesado cuyas bases se presentan y detallan a continuación (la totalidad del código fuente en lenguaje *Fortran95* se incluyen en el anexo A). La función de este algoritmo es calcular, entre otros parámetros, el rendimiento o el salto energético del fluido entre la entrada del rotor y la salida de la voluta, de vital importancia para lograr comparar y optimizar mediante estudios cuantitativos.

Uno de los parámetros de mayor interés y con más valor comparativo es el rendimiento global del ventilador. Su definición general para una turbomáquina impulsora es,

$$\eta_{tot} = \frac{\Delta \dot{E}}{\dot{W}_{Iny}} = \frac{\dot{E}_{Sal Voluta} - \dot{E}_{Ent Rotor}}{\Gamma_{Alabes} \cdot \omega_{Rotor}} \quad (31)$$

De los cuatro términos necesarios, únicamente es un parámetro constante y conocido a priori  $\omega_{Rotor}$ . El resto deben calcularse a partir de los campos de presión y velocidad mediante las expresiones (32), (33) y (34). Las dos primeras permiten conocer la energía que atraviesa las respectivas superficies de control por unidad de tiempo y la última, el par mecánico que realiza el rotor sobre el fluido en las condiciones del flujo resultado despreciando los esfuerzos viscosos sobre los álabes,

$$\dot{E}_{Sal Voluta} = \int_{S_{sal}} \left( p + \frac{1}{2} \rho |\mathbf{u}|^2 \right) [(\mathbf{u} - \mathbf{u}_S) \cdot \mathbf{n}] dS = \int_{y_0}^{y_n} \left( p(y) + \frac{1}{2} \rho |\mathbf{u}(y)|^2 \right) u(y) dy \quad (32)$$

$$\dot{E}_{Ent Rotor} = \int_{S_{ent}} \left( p + \frac{1}{2} \rho |\mathbf{u}|^2 \right) [(\mathbf{u} - \mathbf{u}_S) \cdot \mathbf{n}] dS = \int_0^{2\pi} \left( p(\theta) + \frac{1}{2} \rho U^2 \right) U r_{ent} d\theta \quad (33)$$

Breve inciso para recordar que  $U$  ha sido definido previamente en la página 30 como el módulo del vector velocidad absoluta (y cuya dirección es radial) a lo largo de la superficie de entrada del rotor.

$$\Gamma_{Alabes} = \sum_{j=1}^{n^{\circ}alabes} \left( \int_{S_j} (\mathbf{r}_{dS} \times [-p \mathbf{n} dS]) \right) \quad (34)$$

Estas tres expresiones analíticas deben ser discretizadas para poder incluirlas en el algoritmo. En todo este proceso se considerarán los perfiles de presión y de las componentes de  $\mathbf{u}$ ,  $(u, v)$ , como aproximaciones lineales entre los valores conocidos de dos nodos consecutivos .

Con esta aproximación se pueden realizar las integrales analíticas entre cada pareja de nodos para todos los términos recogidos en (32), (33) y (34). Se muestra a continuación el proceso de discretización brevemente detallado para el término de presión de  $\dot{E}_{Sal Voluta}$  (35) a título de ejemplo, para el término de energía cinética de  $\dot{E}_{Sal Voluta}$  (36), para los dos de  $\dot{E}_{Ent Rotor}$  (37),(38) y para el de  $\Gamma_{Alabes}$  (39) se incluye únicamente la expresión discretizada final con el objetivo de no postergar la conclusión de la exposición.

Referido a la nomenclatura, se aclara que de aquí en adelante;  $\phi_i \equiv$  propiedad  $\phi$  del nodo  $i$  ,  $\Delta\phi \equiv \phi_{i+1} - \phi_i$ . En la curva que modela la superficie de entrada y en la de cada álabe, al ser cerradas,  $\phi_{n+1} \equiv \phi_1$ . Además, en base a las simulaciones posteriores se ha constatado que para todos los nodos que conforman la superficie de salida  $v \approx 0$  , por lo que en (36) se ha aplicado la aproximación  $\mathbf{u}(y) \approx (u(y), 0)$ . En consecuencia,

$$\begin{aligned} \int_{y_0}^{y_n} p(y)u(y)dy &\approx \sum_{i=1}^{n-1} \int_{y_i}^{y_{i+1}} \left( \frac{p_{i+1} - p_i}{y_{i+1} - y_i} (y - y_i) + p_i \right) \left( \frac{u_{i+1} - u_i}{u_{i+1} - u_i} (y - y_i) + u_i \right) dy \\ &= \sum_{i=1}^{n-1} (y_{i+1} - y_i) \left( \frac{1}{3} \Delta p \Delta u + \frac{1}{2} (u_i \Delta p + p_i \Delta u) + u_i p_i \right) \end{aligned} \quad (35)$$

$$\int_{y_0}^{y_n} \frac{1}{2} \rho |\mathbf{u}(y)|^2 u(y) dy \approx \frac{\rho}{2} \sum_{i=1}^{n-1} (y_{i+1} - y_i) \left( \frac{1}{4} (\Delta u)^3 + u_i (\Delta u)^2 + \frac{3}{2} u_i^2 \Delta u + u_i^3 \right) \quad (36)$$

$$\int_0^{2\pi} U r_{ent} p(\theta) d\theta \approx U \sum_{i=1}^n \frac{1}{2} (p_{i+1} + p_i) |\mathbf{r}_{i+1} - \mathbf{r}_i| \quad (37)$$

$$\int_0^{2\pi} \frac{1}{2} \rho U^3 r_{ent} d\theta = \pi \rho U^3 r_{ent} \quad (38)$$

$$\sum_{j=1}^{n^{\circ}alabes} \left( \int_{S_j} (\mathbf{r}_{dS} \times [-p \mathbf{n} dS]) \right) \approx \sum_{j=1}^{n^{\circ}alabes} \left( \sum_{i=1}^n (\mathbf{r}_i + \alpha_i \mathbf{l}_i) \times \left( -\frac{1}{2} |\mathbf{l}_i| (p_{i+1} + p_i) \mathbf{n}_i \right) \right) \quad (39)$$

En (39)  $\alpha_i \mathbf{l}_i$  corresponde al punto de aplicación;  $\frac{p_i + \frac{2}{3}(p_{i+1} - p_i)}{(p_{i+1} + p_i)} (\mathbf{r}_{i+1} - \mathbf{r}_i)$ .

También en (39),  $\mathbf{n}_i$  representa el vector unitario perpendicular a  $(\mathbf{r}_{i+1} - \mathbf{r}_i)$  orientado hacia el exterior del álabe.

Las ecuaciones discretizadas (35)-(39) han sido incluidas en el algoritmo, rechazando la idea de realizar simples promedios, con el fin de alcanzar la mayor precisión posible pese a complicar ligeramente las expresiones teóricas.

Para poder programar las expresiones anteriores, los nodos involucrados deben ser identificados entre todos los que componen la malla, separados por superficies (entrada, salida y cada álabe individual) y ordenados dentro de cada superficie. Se identifican y separan por el factor diferenciador de las condiciones de contorno y se ordenan mediante la conectividad de la malla.

Por último, otros parámetros relevantes muy relacionados con el incremento energético del fluido, y empleados en comparaciones posteriores, son el incremento de presión total  $\Delta p_{tot}$ , el incremento de presión total a estática  $\Delta p_{tot-est}$  (o  $\Delta p_{est}$  para simplificar de aquí en adelante) y el rendimiento estático  $\eta_{est}$  calculados trivialmente tras conocer los términos de presión estática y cinética de  $\dot{E}_{Sal Voluta}$  y  $\dot{E}_{Ent Rotor}$  según (40),

$$\Delta p_{tot} = \frac{\Delta \dot{E}}{Q} ; \quad \Delta p_{tot-est} = \frac{\dot{E}_{est Sal} - (\dot{E}_{est Ent} + \dot{E}_{cin Ent})}{Q} ; \quad \eta_{est} = \frac{Q \Delta p_{tot-est}}{\dot{W}_{iny}} \quad (40)$$

## 4.4. Resultados principales

En esta sección se incluyen diferentes resultados de las simulaciones relacionadas con el modelo 2D del ventilador centrífugo y su respectiva malla (caso base). En primer lugar, se presentan detalladamente los resultados obtenidos con el algoritmo modificado y validado en el apartado 3. Posteriormente se incluyen comparaciones cuantitativas entre los resultados del código modificado, del original (sin el modelo turbulento incorporado) y datos experimentales del ventilador real.

### 4.4.1. Caso base

Respecto al caso base, es decir, geometría de las figuras 21 y 22 y mallas de la 25, simulado con el código modificado y con los parámetros nominales de funcionamiento del ventilador real ( $N = 4000 \text{ rpm}$ ,  $Q = 0,049 \text{ m}^3/\text{s}$ ), se incluyen de una forma gráfica y visual los resultados del campo de velocidad absoluta, en las figuras 26 y 27a, del campo de velocidad relativa al subdominio rotante en la figura 27b y del campo de presiones, en las figuras 28 y 29a junto con la gráfica del perfil de presión sobre el álabe del detalle (29b).

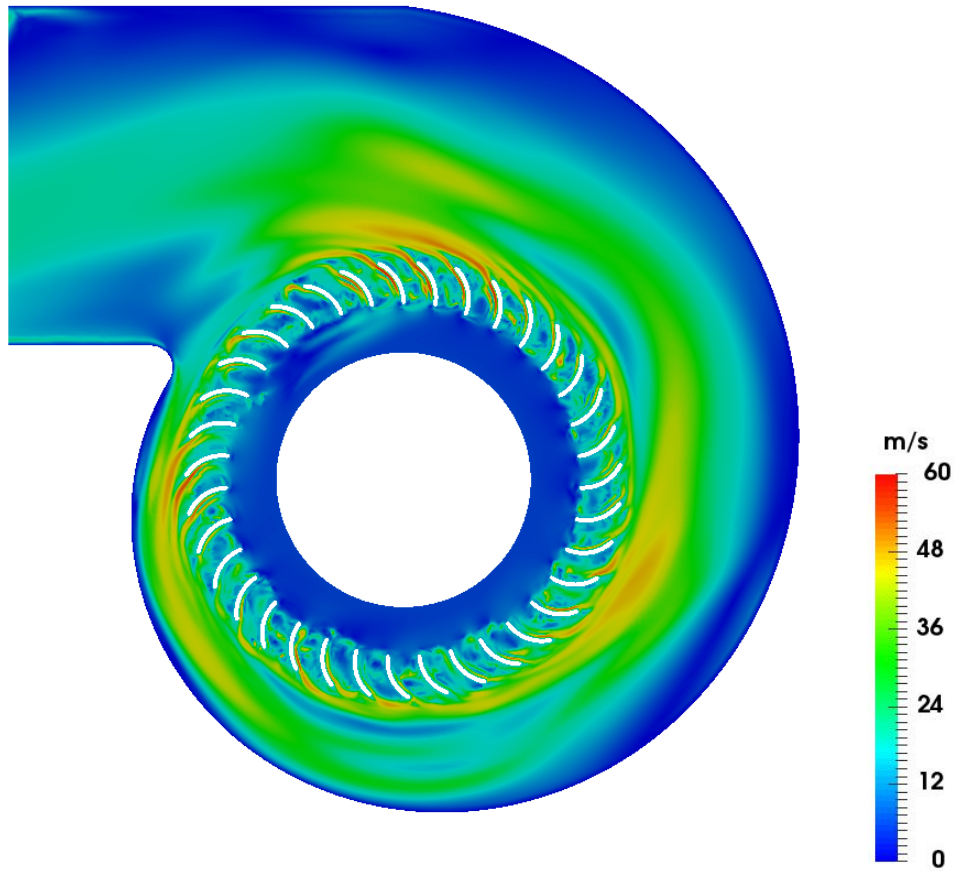


Figura 26: Campo global del módulo de la velocidad absoluta. Caso base

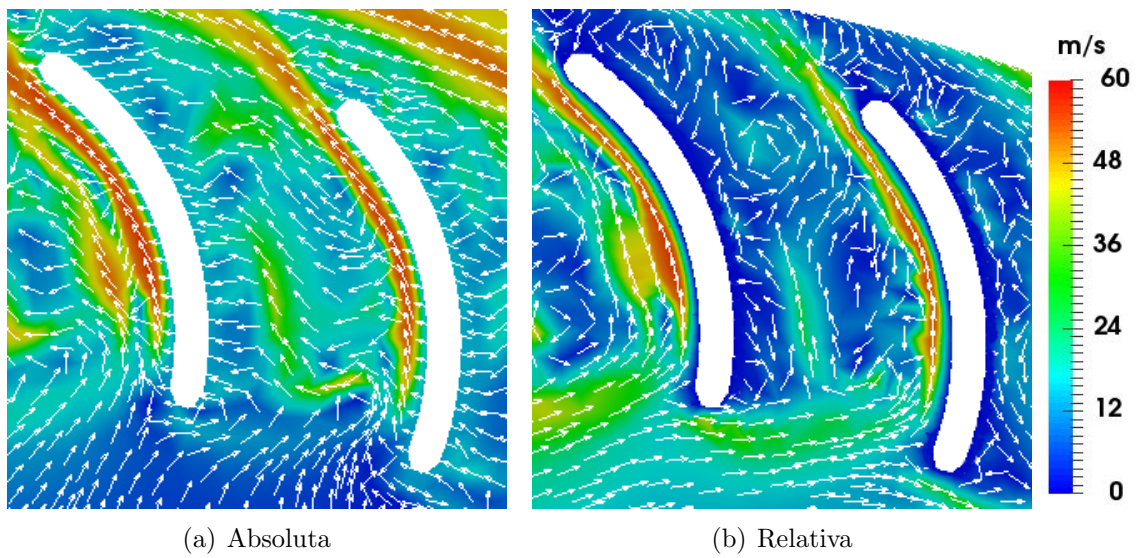


Figura 27: Campo de velocidades detallado en el álabe. Caso base



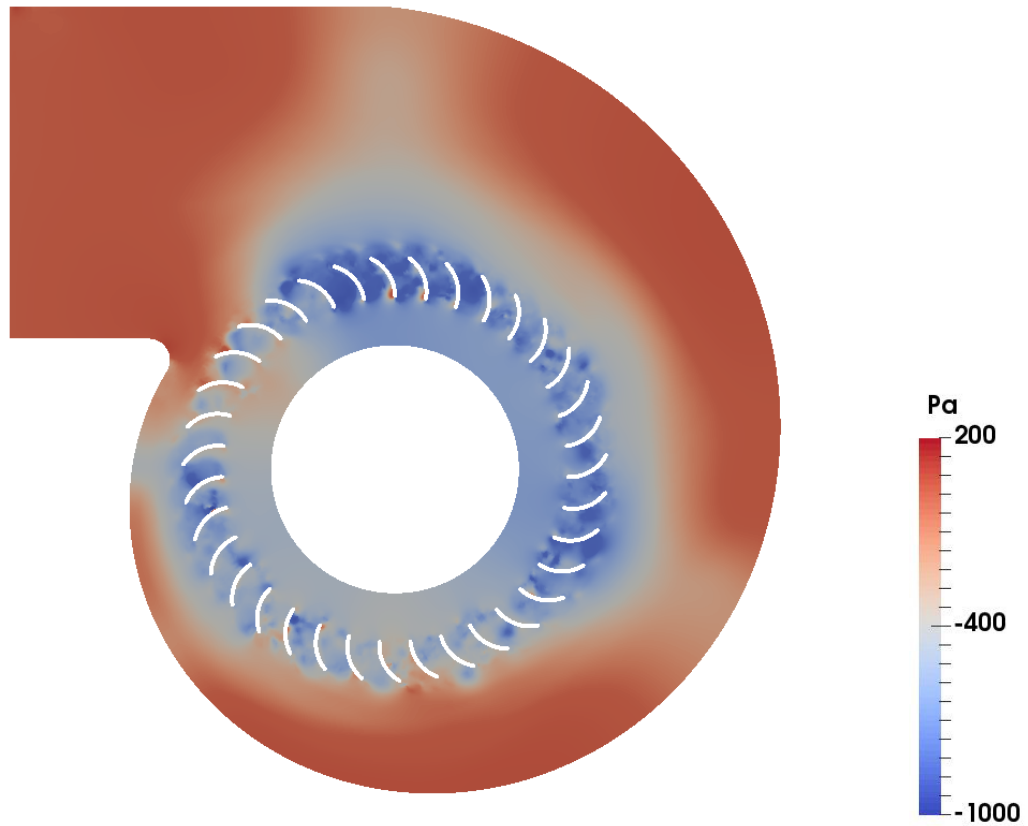


Figura 28: Campo global de presiones. Caso base

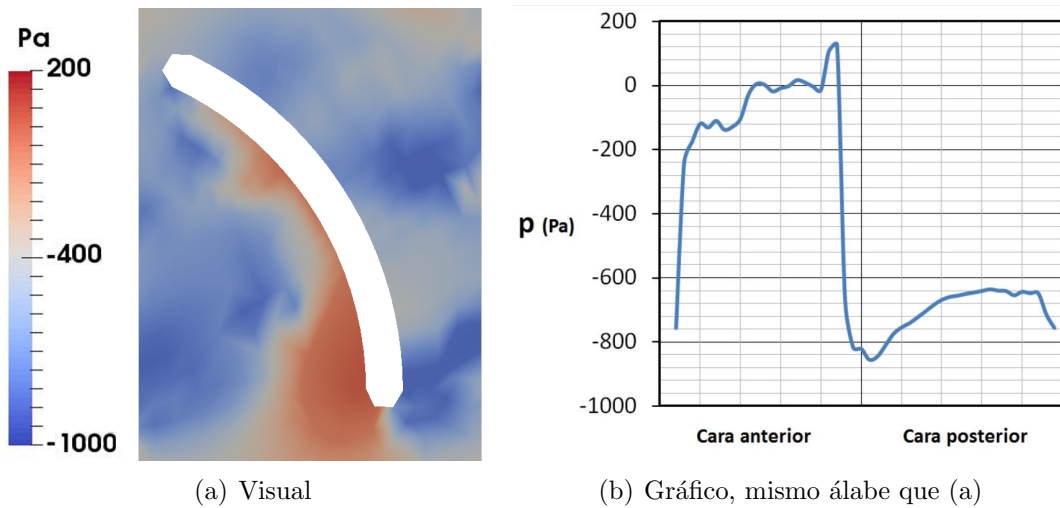


Figura 29: Campo de presiones detallado en el álabe . Caso base

Comentar en primer lugar que los resultados mostrados en las figuras 26 - 29 son coherentes en términos generales con el flujo esperado, lo que corrobora el correcto funcionamiento del algoritmo y del MRF.

Entrando en detalle, es interesante observar en la figura 27 el desprendimiento del flujo tras los álabes generando vórtices de escalas muy variadas. Esta observación puede dar pie a posibles propuestas de optimización basadas en la modificación del ángulo de entrada ( $\beta_1 = 90^\circ$ ) y/o el de salida ( $\beta_2 = 150^\circ$ ) persiguiendo un menor desprendimiento del flujo. Para visualizar de una forma más nítida los vórtices que se desarrollan entre los álabes se incluye la figura 30 con las líneas de corriente sobreimpresas.

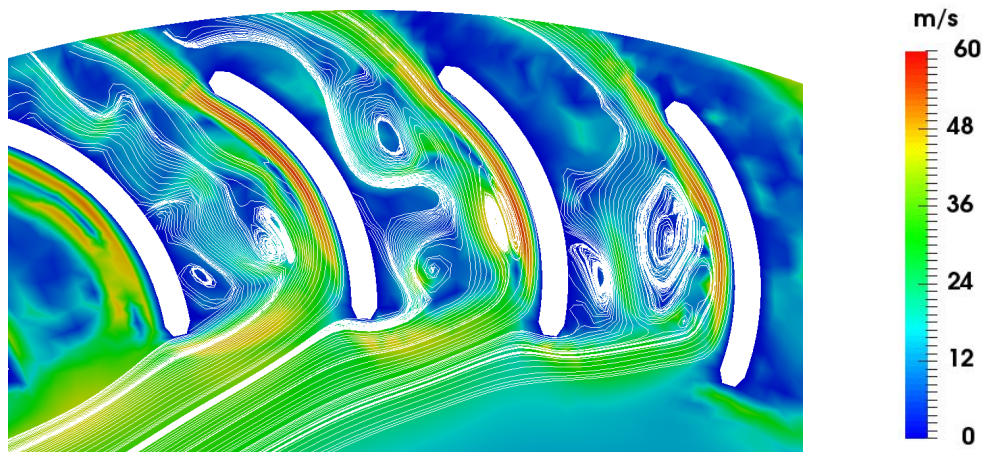


Figura 30: Vórtices entre los álabes. Líneas de corriente. Caso base

En la figura 28 se percibe el aumento general de presión que sufre el fluido tras pasar por los álabes y en el detalle de la figura 29a se observa la distribución de presiones sobre los álabes que genera el par mecánico de oposición al giro del rotor y en última instancia, la necesidad de inyectar potencia desde el exterior. El análisis cuantitativo de estos perfiles de presión a lo largo de los álabes mostrado en la figura 29b pretende ser una herramienta comparativa en la búsqueda del aumento del rendimiento, puesto que la reducción de la diferencia de presiones entre la cara posterior y la anterior de los álabes puede reducir el consumo del ventilador.

Para concluir la exposición de los resultados del caso base se concretan los aspectos relacionados con la etapa transitoria y la convergencia al régimen de flujo estacionario.

Al tratarse de un problema con un componente rotatorio y un elevado número de Re (40.000, 80.000) se puede prever que la solución no se corresponderá con un

estado estacionario puro independiente completamente del tiempo. Existirá una fase transitoria muy marcada durante las primeras iteraciones del solver para alcanzar a continuación un régimen aproximadamente estacionario pero oscilante dentro de unos límites.

Representar la duración de estas fases y acotar los límites entre los que oscila la solución estacionaria son los objetivos de la figura 31. En ella se muestra la evolución temporal de parámetros globales como  $\eta_{tot}$  y  $\Delta p_{tot}$  durante las primeras 200 iteraciones. Para obtener esta serie de datos se ha ejecutado la simulación con un paso temporal entre cada iteración de 0,2 milésimas de segundo.

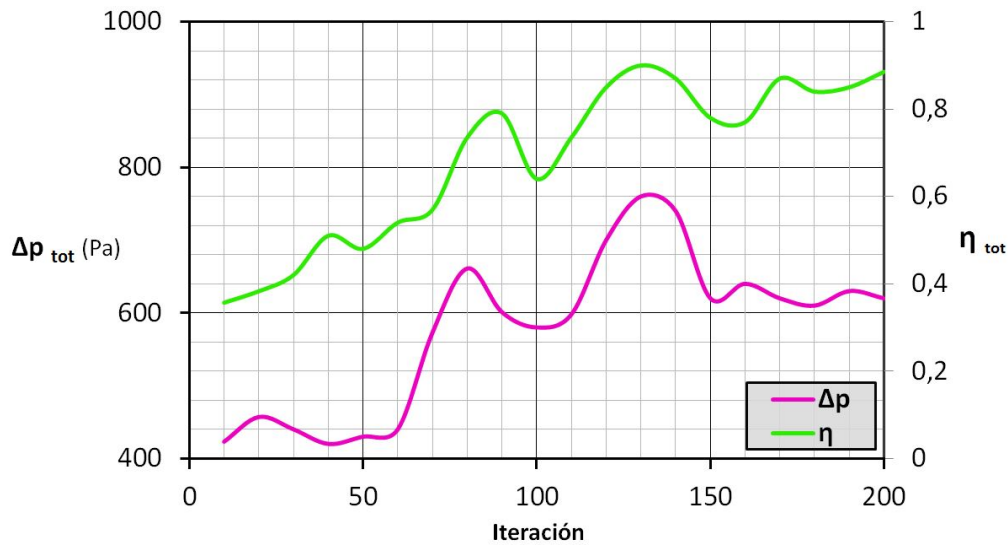


Figura 31: Evolución temporal de parámetros globales. Caso base

Se puede observar en la figura 31 que el régimen transitorio termina tras 150 iteraciones, equivalentes a dos revoluciones completas del rotor (0,030 segundos). Los resultados mostrados anteriormente en las páginas 36, 37 y 38 corresponden al paso 200, situándose lo suficientemente lejos del final del transitorio más marcado.

#### 4.4.2. Comparaciones con resultados laminares y experimentales

El caso base inspirado en la geometría real de la turbomáquina se simula también con la versión anterior del código para comparar los resultados turbulentos mostrados en el apartado 4.4.1 con los de la versión laminar.

En primer lugar, mediante comparaciones visuales se pueden determinar las primeras diferencias cualitativas. En la figura 32 se pueden observar los campos de velocidad, para el caso laminar tiene lugar una mayor uniformidad, en el desprendimiento del flujo se generan vórtices únicos detrás de cada álabe de una dimensión del orden de magnitud del propio álabe, mientras que en la simulación turbulenta

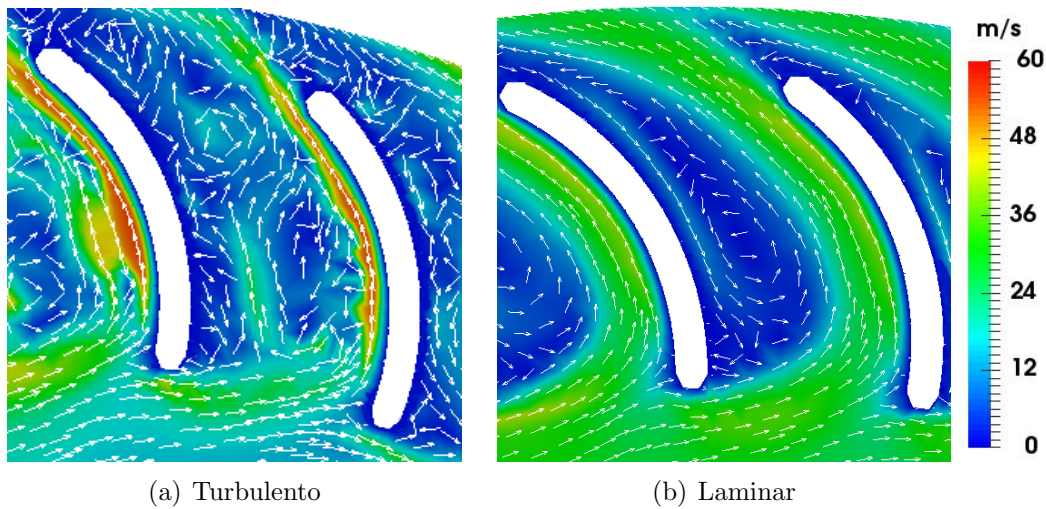


Figura 32: Velocidad relativa detallada en los álabes. Caso base

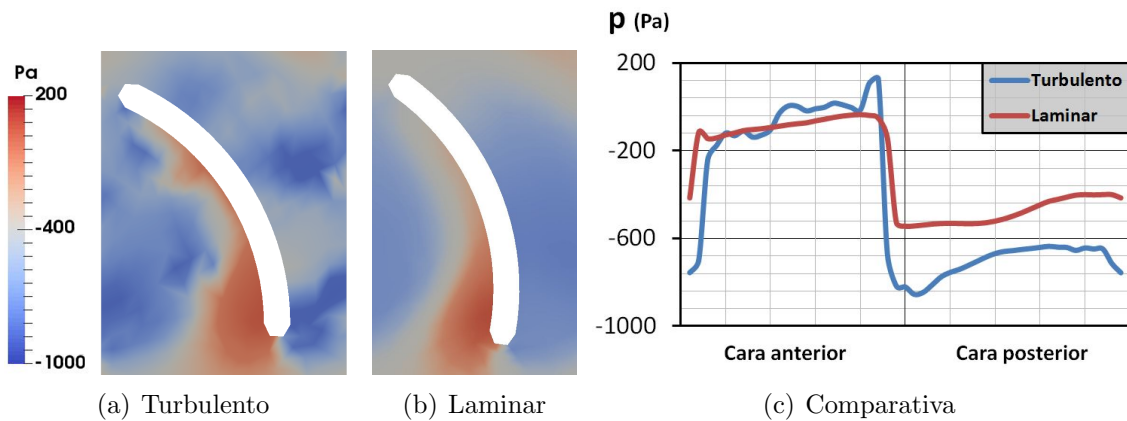


Figura 33: Presión detallada en los álabes. Caso base

aparecen numerosos vórtices de menor tamaño. Respecto al perfil de presiones, se puede apreciar en la figura 33 que el flujo laminar no genera grandes fluctuaciones sobre cada una de las caras del álabe, únicamente se produce al salto entre la cara posterior y la anterior, sin embargo, el flujo turbulento ocasiona fluctuaciones mucho más acentuadas dentro de una misma cara.

Además de las comparativas visuales de las figuras 32 y 33, se realizan simulaciones con ambas versiones del código para cuatro valores diferentes de caudal con el objetivo de obtener las curvas características  $\Delta p_{tot-est}(Q)$  y  $\eta_{est}(Q)$ , presentando de este modo comparaciones cuantitativas en la figura 34. Se incluyen también en dicha figura mediciones experimentales para contextualizar la precisión de todas las simulaciones.

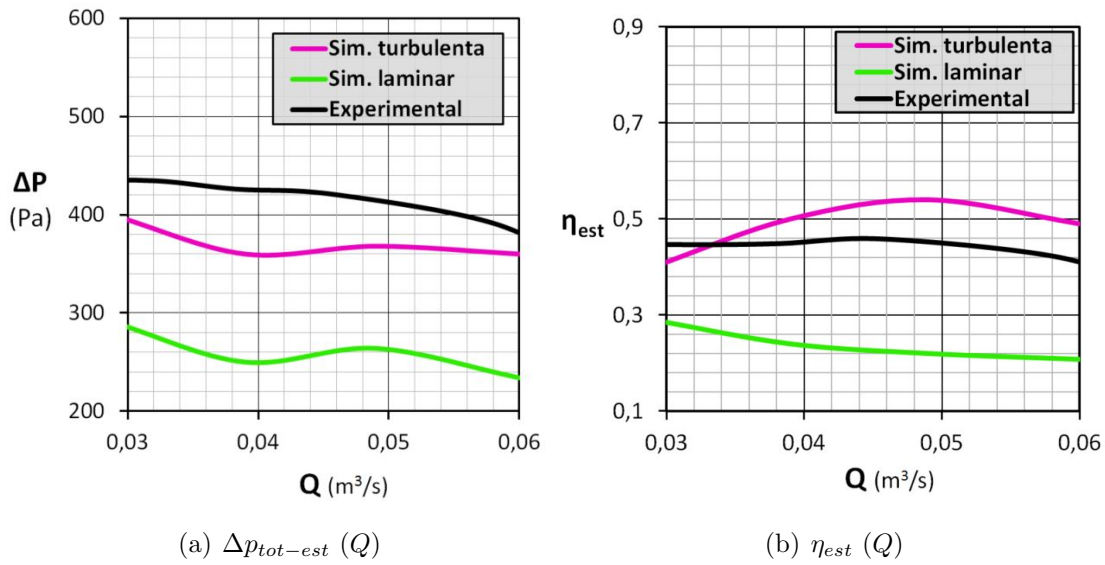


Figura 34: Curvas características de la turbomáquina

Sin contabilizar el error inherente a la simulación (método numérico, discretización del dominio, modelo turbulento, MRF...), existen discrepancias respecto a los datos experimentales asociadas a las limitaciones del modelo 2D. Las mediciones experimentales computan fenómenos como el flujo por los conductos hasta la entrada del rodete o como la recirculación, que son imposibles de tener en cuenta en un modelo bidimensional. Pese a todo ello, se puede observar en la figura 34 una mejora notable en la precisión con respecto a la versión para flujo laminar del algoritmo, obteniendo resultados mucho más próximos a los experimentales con el modelo de turbulencia, lo cual supone un indudable paso adelante.

## 4.5. Propuesta de optimización

Una vez analizados los resultados del caso base es el momento adecuado para explorar ciertas modificaciones de la geometría constructiva que desemboquen en una mejora de la eficiencia global.

Los principales aspectos sobre los que se puede actuar para lograr una optimización son la curvatura de la voluta y la geometría de los álabes. Está ampliamente comprobado que aumentar la apertura de la espiral de la voluta mejora la eficiencia de este tipo de turbomáquinas puesto que se modifica de una forma más progresiva y suave la dirección del flujo que sale del rotor, sin embargo, por limitaciones de espacio, la expansión de la voluta suele ser inviable. En consecuencia, lo más factible es actuar sobre los ángulos de entrada y de salida de los álabes.

La primera idea es reducir el ángulo de entrada de los álabes ( $\beta_1$ ) desde los 90° que adquiere en el caso base hasta los 22°, manteniendo el resto de parámetros inalterados. Ver la figura 22b de la página 29 para visualizar  $\beta_1$ . El nuevo valor

propuesto viene estipulado por la intención de que la velocidad relativa del fluido con respecto al álabe sea paralela a la geometría del mismo en su entrada, persiguiendo que el desprendimiento del flujo sea menor y no se generen vórtices tras los álabes que reduzcan la presión y aumenten el par mecánico sobre el rotor. Dicho valor de  $\beta_1$  se calcula mediante (41) en base al triángulo de velocidades (29) que se genera con los parámetros nominales de funcionamiento de la turbomáquina real ( $N = 4000 \text{ rpm}$ ,  $Q = 0,049 \text{ m}^3/\text{s}$ ), mostrado en la figura 35.

$$\tan\beta_1 = \frac{|\mathbf{u}_f|}{|\boldsymbol{\omega} \times \mathbf{r}|} = \frac{Q_n}{2\pi b r_{ent} Rot} \cdot \frac{1}{\omega_n r_{int} Alabes} \quad (41)$$

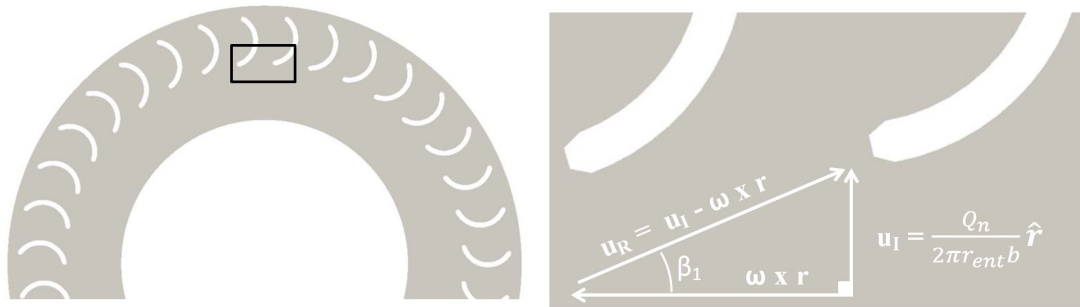


Figura 35: Triángulo de velocidades en la entrada de los álabes. 1ª propuesta.

Los resultados de las simulaciones ejecutadas sobre el rotor con la nueva geometría de álabe ( $\beta_1 = 22^\circ$ ,  $\beta_2 = 150^\circ$ ) desvelan que pese a lograr el objetivo de evitar el desprendimiento en la cara posterior, aparece un efecto no deseado en la cara anterior, un vórtice que dificulta el flujo precisamente en la zona donde debería ser impulsado, por lo que no extraña que los saltos energéticos y las eficiencias obtenidas en el post-proceso resulten ser mucho menores. Ver figura 36.

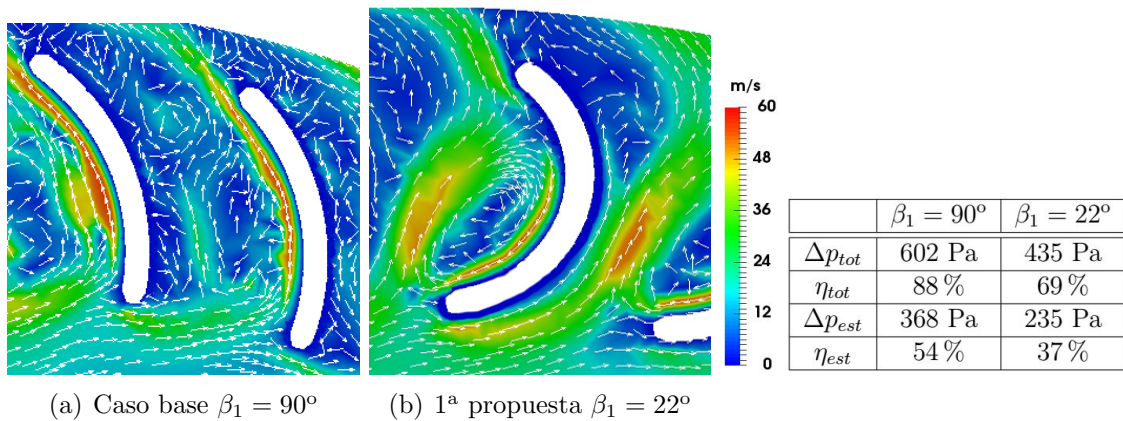


Figura 36: Velocidad relativa detallada en los álabes y comparación de los resultados post-procesados entre el caso base y la primera propuesta de optimización

Antes de abandonar el objetivo de evitar el desprendimiento del flujo tras los álabes mediante la reducción de  $\beta_1$ , se pretende subsanar el efecto del vórtice en la cara anterior eliminando una de sus posibles causas; la excesiva curvatura que se ha generado en el álabe al reducir  $\beta_1$  sin modificar  $\beta_2$ . De este modo se configura la segunda propuesta, cuyos álabes mantienen el ángulo de entrada a  $22^\circ$  y ven reducido el de salida a  $100^\circ$ .

Esta reducción del ángulo de salida también alberga un inconveniente, y es que la componente tangencial de la velocidad absoluta en la salida del rotor se reduce y la voluta se ve obligada a forzar un cambio de dirección más brusco (ver figura 37), por lo que idealmente y en caso de disponer del espacio necesario, la reducción de  $\beta_2$  debería ir acompañada de un aumento de la apertura de la espiral. En este caso no es así porque se ha asumido la restricción de que el espacio disponible para albergar la turbomáquina en su aplicación real ya está ocupado completamente.

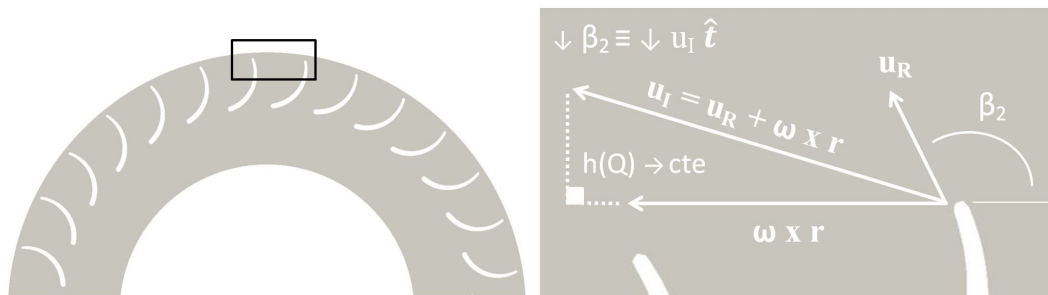


Figura 37: Triángulo de velocidades en la salida de los álabes. 2ª propuesta

Los resultados mostrados en la figura 38 refutan la idea de disminuir el ángulo de salida  $\beta_2$  puesto que no se logra ningún avance. Se siguen generando vórtices en la cada impulsora de los álabes y el empeoramiento de los resultados cuantitativos es rotundo. Se concluye por tanto que los ángulos del caso base ya están optimizados y se rechaza la idea de modificarlos.

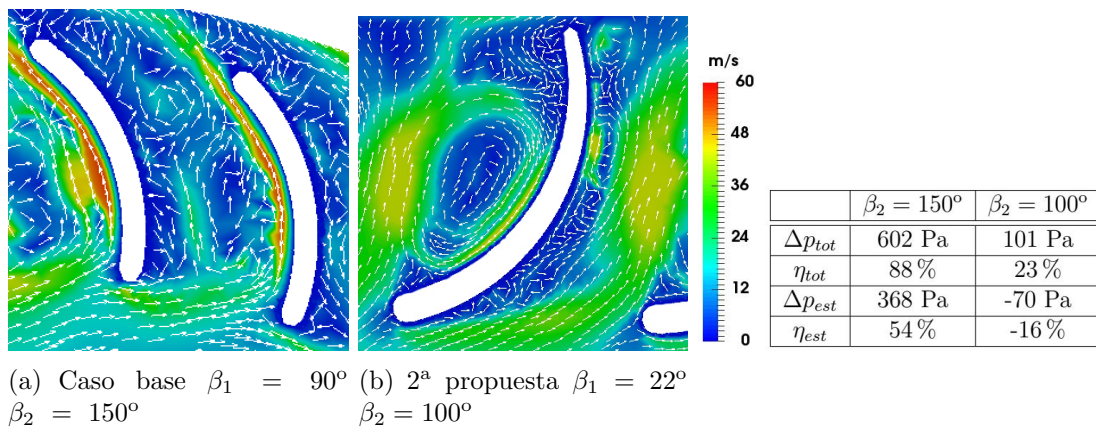


Figura 38: Velocidad relativa detallada en los álabes y comparación de los resultados post-procesados entre el caso base y la segunda propuesta de optimización

Para la tercera y última propuesta de optimización se recurre a una idea poco habitual inspirada en [13] y basada en emplear un espesor de álabe variable para lograr reducir el desprendimiento y la depresión en la cara posterior sin modificar los ángulos de la cara incidente. En la figura 39 se puede observar la geometría de los nuevos álabes, mediante la cual se pretende combinar los aspectos positivos del caso base y de la primera propuesta.



Figura 39: Geometría rotor. 3ª propuesta

Los resultados incluidos en la figura 40 señalan cierto éxito en el proceso de optimización. Se logra mayor ordenación en el flujo, sin embargo, el incremento de superficie de los álabes eleva el par mecánico un 50 % respecto al caso base (de 2 a 3  $Nm/m$ ). Estos efectos contrarios se traducen en un ligero descenso de  $\eta_{tot}$  y viceversa para  $\eta_{tot-est}$ . Además, es relevante el hecho de que se logran mantener altos niveles de eficiencia para saltos energéticos mayores, lo que puede ampliar el rango de aplicación de la turbomáquina hacia puntos de funcionamiento de mayor demanda energética antes limitados por la pérdida de rendimiento.

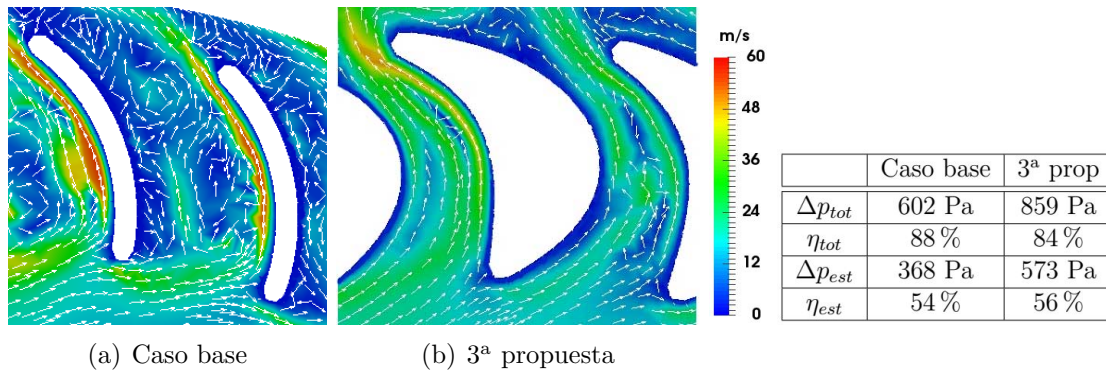


Figura 40: Velocidad relativa detallada en los álabes y comparación de los resultados post-procesados entre el caso base y la tercera propuesta de optimización

Matizar antes de concluir que las parejas de mallas empleadas en el presente apartado están constituidas con la misma densidad de elementos y en base a las mismas directrices que la dupla del caso base presentada en el apartado 4.2, con diferencias muy sutiles generadas únicamente por los cambios en la geometría de los álabes. Además todas las simulaciones han sido ejecutadas para los parámetros nominales de funcionamiento de la turbomáquina real ( $N = 4000 \text{ rpm}$ ,  $Q = 0,049 \text{ m}^3/s$ ).



---

## 5. Conclusiones generales

Este trabajo Fin de Grado, enmarcado en un proyecto de investigación para el desarrollo de herramientas avanzadas de simulación fluidodinámica computacional, ha perseguido dos objetivos principales. Por un lado la validación del modelo de turbulencia VMS (*Variational MultiScale*) implementado sobre un algoritmo de elementos finitos concebido para resolver flujos incompresibles bidimensionales. Y por otro, la aplicación del mismo en la simulación y optimización de una turbomáquina.

Durante la validación, se ha simulado el problema LDC (*Lid Driven Cavity*), comparando los resultados con la bibliografía existente y se ha realizado un análisis de sensibilidad para lograr emplear la malla óptima en la obtención de dichos resultados. La realización de este proceso ha permitido constatar la notable mejora que ha supuesto el modelo turbulento en la predicción de flujos turbulentos, la cuál se ha podido cuantificar mediante el algoritmo desarrollado de forma original adjunto íntegramente en el anexo B. La conclusión satisfactoria del proceso de validación se puede resumir y representar mediante la figura 41. En ella también se compendian los resultados del análisis de sensibilidad de la malla.

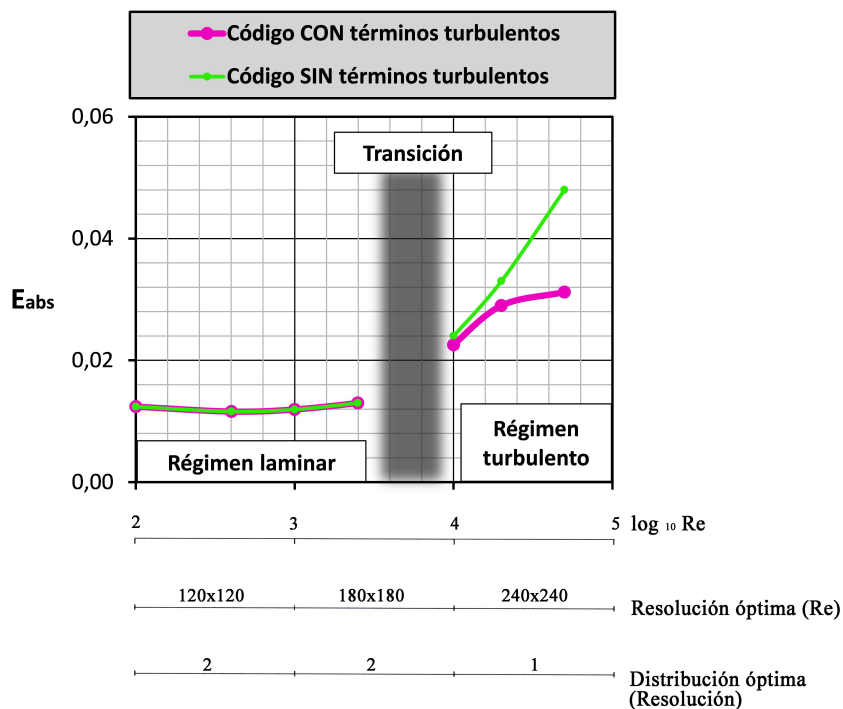


Figura 41: Error absoluto respecto a la solución de referencia en función de  $Re$  y resultados del análisis de sensibilidad de la malla. LDC

Respecto a la simulación de la turbomáquina cabe decir que, tras aproximar la geometría real de un ventilador centrífugo mediante un modelo 2D y programar el código de post-procesado que se incluye íntegro en el anexo A, se han obtenido resultados satisfactorios mediante el método MRF.

Los resultados principales del modelo 2D se han comparado con las curvas características reales de la turbomáquina tridimensional para constatar la precisión del solver de elementos finitos. Como se puede apreciar en la figura 42, estas nuevas comparaciones ratifican una mejora más que notable con respecto a la versión anterior sin el modelo turbulento.

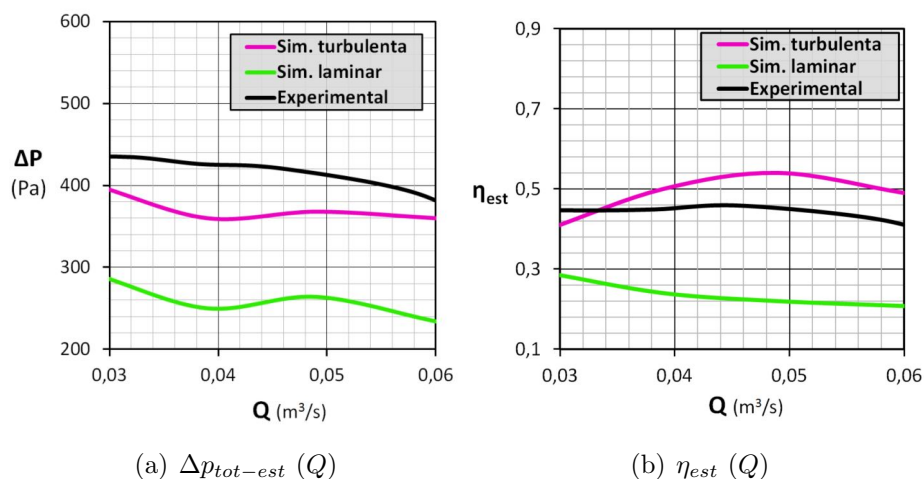


Figura 42: Curvas características de la turbomáquina

Finalmente, se ha desarrollado un proceso de optimización de los elementos constructivos de la turbomáquina, centrando el esfuerzo en la geometría de los álabes del rotor, puesto que la geometría de la voluta suele estar restringida por las limitaciones espaciales de cada aplicación concreta. En base a las premisas de la teoría de máquinas hidráulicas se han generado tres propuestas diferentes a partir del caso base, resultando carentes de éxito las dos primeras y albergando cierta mejora la última de ellas. Se incluyen en la figura 43 los resultados de la tercera propuesta.

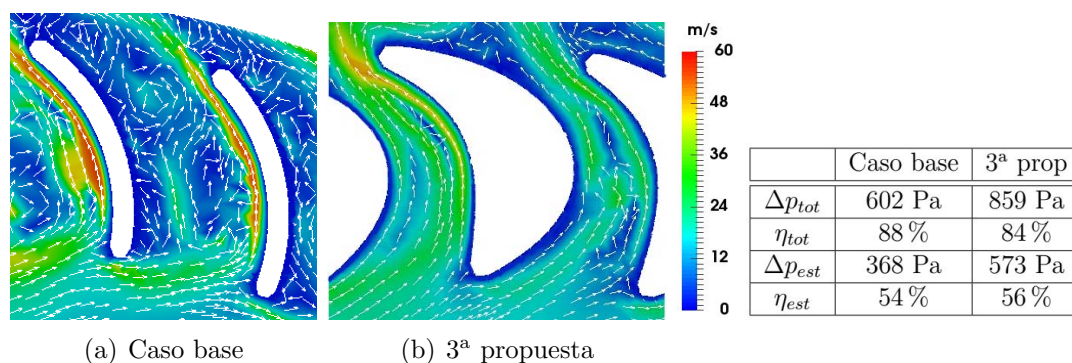


Figura 43: Resultados de la tercera propuesta de optimización

Se concluye de este modo el presente Trabajo Fin de Grado tras alcanzar el cumplimiento de los objetivos principales marcados para su realización.

## Bibliografía

- [1] G.HAUKE, *Introducción al Método de Elementos Finitos en Fluidodinámica Computacional* (1ª edición, 2014)
- [2] P.A.DAVIDSON, *Turbulence, An Introduction for Scientists and Engineers* (2ª edición, 2015)
- [3] J.E.MARSDEN, A.TROMBA *Cálculo vectorial* (6ª edición, 2018)
- [4] Y.A.CENGEL, *Mecánica de fluidos, fundamentos y aplicaciones* (4ª edición, 2019)
- [5] U.GHIA, K.N.GHIA, C.T.SHIN, *High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method* (1982)
- [6] J.R.KOSEFF, R.L.STREET, *The Lid-Driven Cavity Flow: A Synthesis of Qualitative and Quantitative Observations* (1984)
- [7] L.FRANCA, S.FREY, *Stabilized finite element methods: II. the incompressible Navier-Stokes equations* (1992)
- [8] M.D.DESHPANDE, S.G MILTON, *Kolmogorov scales in a driven cavity flow* (1998)
- [9] O.BOTELLA, R.PEYRET, *Benchmark spectral results on the lid-driven Cavity flow* (1998)
- [10] E.ERTURK, T.C.CORKE, C.GOKCOL, *Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers* (2005)
- [11] Y.BAZILEVS, V.M.CALO, J.A.COTTRELL, T.J.R.HUGHES, A.REALI, G.SCOVAZZI, *Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows* (2007)
- [12] H.J.NAGAPETYAN, *Application of Wray-Agarwal Model to Turbulent Flow in a 2D Lid-Driven Cavity and a 3D LidDriven Box* (2015)
- [13] A.L.LACUEY, *Simulación fluidodinámica de ventiladores centrífugos para climatización de vehículos* (2018)
- [14] F.BENITEZ, J.D'ELIA, M.A.STORTI, *Numerical solution of the Navier-Stokes equations on Multi-GPGPU* (2018)

## Índice de figuras

1.	Contexto del trabajo en el proyecto de investigación . . . . .	4
2.	Condiciones de contorno y geometría del problema LDC . . . . .	13
3.	Módulo del vector velocidad. LDC . . . . .	15
4.	Lineas de corriente ( <i>Streamlines</i> ). LDC . . . . .	15
5.	Campo de presión. LDC . . . . .	15
6.	Lugares geométricos donde se obtienen los perfiles de velocidad. LDC	16
7.	Comparativa resultados LDC (Re = 100) . . . . .	17
8.	Comparativa resultados LDC (Re = 400) . . . . .	17
9.	Comparativa resultados LDC (Re = 1.000) . . . . .	17
10.	Comparativa resultados LDC (Re = 10.000) . . . . .	18
11.	Comparativa resultados LDC (Re = 20.000) . . . . .	18
12.	Comparativa resultados LDC (Re = 50.000) . . . . .	18
13.	Error absoluto en función de Re para ambas versiones del solver. LDC	20
14.	Microescalas espaciales de Kolmogórov ( $\eta$ ) en el dominio del LDC, Re = 10.000, [8] . . . . .	21
15.	Tipología de malla distribuida . . . . .	22
16.	Efecto del parámetro $\delta$ en la distribución de elementos. Resolución 60x60 . . . . .	23
17.	Error absoluto en función de la distribución y la resolución de la malla (Re = 1.000) . . . . .	23
18.	Precisión vs Coste computacional en función de Re . . . . .	25
19.	Resolución y distribución de la malla óptima en función de Re . . . .	25
20.	Subdominios e interfaz. Enfoque MRF . . . . .	26
21.	Modelo 2D de la voluta. Cotas en milímetros . . . . .	29
22.	Modelo 2D del rotor. Cotas en milímetros . . . . .	29
23.	Condiciones de contorno sobre el conjunto del modelo 2D . . . . .	30
24.	Control del error cordal en el mallado de los álabes . . . . .	31
25.	Mallas de la geometría base del modelo 2D . . . . .	32
26.	Campo global del módulo de la velocidad absoluta. Caso base . . . .	36
27.	Campo de velocidades detallado en el álabe. Caso base . . . . .	36
28.	Campo global de presiones. Caso base . . . . .	37
29.	Campo de presiones detallado en el álabe . Caso base . . . . .	37
30.	Vórtices entre los álabes. Líneas de corriente. Caso base . . . . .	38
31.	Evolución temporal de parámetros globales. Caso base . . . . .	39
32.	Velocidad relativa detallada en los álabes. Caso base . . . . .	40
33.	Presión detallada en los álabes. Caso base . . . . .	40
34.	Curvas características de la turbomáquina . . . . .	41
35.	Triángulo de velocidades en la entrada de los álabes. 1 <sup>a</sup> propuesta. .	42
36.	Velocidad relativa detallada en los álabes y comparación de los resul- tados post-procesados entre el caso base y la primera propuesta de optimización . . . . .	42
37.	Triángulo de velocidades en la salida de los álabes. 2 <sup>a</sup> propuesta . . .	43

38.	Velocidad relativa detallada en los álabes y comparación de los resultados post-procesados entre el caso base y la segunda propuesta de optimización . . . . .	43
39.	Geometría rotor. 3 <sup>a</sup> propuesta . . . . .	44
40.	Velocidad relativa detallada en los álabes y comparación de los resultados post-procesados entre el caso base y la tercera propuesta de optimización . . . . .	44
41.	Error absoluto respecto a la solución de referencia en función de Re y resultados del análisis de sensibilidad de la malla. LDC . . . . .	45
42.	Curvas características de la turbomáquina . . . . .	46
43.	Resultados de la tercera propuesta de optimización . . . . .	46

## Índice de tablas

1.	Distribución óptima en función de la resolución de la malla . . . . .	24
2.	Mallas empleadas en las simulaciones del apartado 3.2 en función de Re . . . . .	25

# Anexos

Se incluye a modo de apéndices el código fuente de los algoritmos desarrollados de forma original durante la realización del Trabajo de Fin de Grado. En el anexo A se adjunta el algoritmo de post-proceso presentado en el apartado 4.3 de la memoria y empleado en el contexto de la simulación de la turbomáquina con el objetivo de permitir la realización de comparaciones cuantitativas. En el anexo B se incorpora el algoritmo de cálculo del error absoluto, introducido en la sección 3.2 y utilizado en el proceso de validación del modelo turbulento.

## A. Código fuente algoritmo de post-proceso. Turbomáquina

```
1 program eficiencia

!En primer lugar se calcula el par mecanico que aplican los alabes del
  rotor sobre el fluido para conocer la potencia inyectada y a
  continuacion se calcula la energia del flujo a la entrada del
  rotor y a la salida de la voluta para hallar la potencia util

integer i,j,k,m,n,s,p,a,b,z,elemento,next,numalabes,numnod,
  numnodccvelu,numnodccvelv,numelem,numnodent,numnodconj,
  numnodsals,numnodccp,e

integer auxiliar(2000),conecent(2000,2),nodoscorresp(4000)

real aux,plus,pa,pb,modFi,modl,Mi,Mtotal,req,rho,caudal,w,
  rint,modVent,PotEstEnt,PotCinEnt,PotEstSal,PotCinSal,minx,
  miny,efiglobal,efiestatica,dv,du,dp,vi,ui,pi

11 real lector(6),lect(6),nod(4,3),coorda(2),coordb(2),l(2),
  sonda(2,2),perp(2,2),dist(2),unitFi(2),Fi(2),ri(2),
  Malabe(150),nodosentrada(2000,4),nodossalida(2000,6),
  auxiliar2(2000,3)

open (10,file="solpNSdataext.dat")
open (11,file="clasificados.dat")
open (12,file="filtrados.dat")
open (14,file="dataext.dat")
21 open (15,file="kk11")
open (16,file="dataconj.dat")
open (17,file="solpNS.dat")
open (18,file="dataint.dat")
open (19,file="soluNS.dat")
open (20,file="solvNS.dat")
open (21,file="eficiencia.dat")
```

```

! lectura parametros globales de kk11

read (15,*)
read (15,*)
read (15,*)
read (15,*)
read (15,*)
read (15,*)
read (15,*)
41 read (15,*) rho
read (15,*)
read (15,*)
read (15,*)
read (15,*) caudal
read (15,*)
read (15,*) w
read (15,*) rint
read (15,*)

51 close (15)

```

!Existe un problema de correspondencia que se debe solucionar; para el filtrado es necesario emplear los archivos donde se encuentran las condiciones de contorno (dataext y dataint), sin embargo, las soluciones son vertidas con la referencia de los nodos de dataconj, por lo que cuando se quieran leer resultados, se debe hallar la equivalencia de las referencias entre dataint/ext y dataconj

```

! generacion del archivo extra de solucion de presiones con la
referencia dataext para solucionar el problema de la
correspondencia

61 read (14,*) j

do i = 1, j + 1

    read (14,*)

enddo

read (14,*) j
71 do i = 1, j

    read (14,*) nodoscorresp (i)

enddo

close (14)
open (14, file="dataext.dat")

```



```

81 read (14,*) j
    e = 1
    do i = 1, j
        print *, i
        read (14,*) lector(1), lector(2), lector(3), lector(4)
91    if (lector (1) == nodoscorresp (e)) then
            e = e + 1
            lector (3) = lector (3) * 10**3
            s = lector (3)
            lector (4) = lector (4) * 10**3
            p = lector (4)
101    a = 1
        close (16)
        open (16, file="dataconj.dat")
        close (17)
        open (17, file="solpNS.dat")
        read (16,*) numnodconj
        do while (a /= 0)
111            read (16,*) lect (1), lect (2), lect (3), lect (4)
                lect (3) = lect (3) * 10**6
                m = lect (3)
                lect (4) = lect (4) * 10**6
                n = lect (4)
                if ((s >= m - 1).and.(s <= m + 1).and.(p >=
121 n-1).and.(p <= n + 1)) then
                    a = 0
                    z = lect (1)
                    do k = 1, z - 1
                        read (17,*)
                    enddo
131            read (17,*) aux, plus
                z = lector (1)

```

```

                write (10,*) z, plus
            endif
        enddo
141     else
            z = lector (1)
            write (10,*) z, -1
        endif
    enddo
close (14)
151 close (17)
    close (16)

! Fin generacion del archivo extra de solucion de presiones con la
    referencia dataext

! PAR MECANICO (POTENCIA INYECTADA)
161 open (14, file="dataext.dat")
    read (14,*) numnod
    j = numnod + 1
    do i = 1, j
        read (14,*)
171     enddo
        read (14,*) numnodccvelu
        do i = 1, numnodccvelu + 1
            read (14, *)
        enddo
181     read (14,*) numnodccvelv
        do i = 1, numnodccvelv + 1
            read (14,*)
        enddo
    enddo

```

```

    read (14,*) numnodecp
191 do i = 1, numnodecp + 2
        read (14,*)
    enddo

    read (14,*) numelem

    read (11,*) numalabes
201 Mtotal = 0

    do i = 1, numalabes
        Malabe (i) = 0

        read (11,*)
        read (11,*) a
        read (11,*) b
211
        !!!!! determinacion del tipo de ordenacion que tienen los nodos de
        cada alabe (horaria o antihoraria)
        !!!!! clave para dirigir el vector unitario de la fuerza hacia el
        interior del alabe

        elemento = 0

        do while (elemento == 0)! busqueda del elemento que contiene a los
        dos primeros nodos del alabe

            read (14,*) lector (1), lector (2), lector (3), lector (4),
            lector (5), lector (6)

221         m = 0

            do j = 3, 6

                if ((lector(j) == a) .or. (lector(j) == b)) m = m + 1

            enddo

            if (m == 2) then

231                 elemento = lector (1)
                 nod(1,1) = lector (3)
                 nod(2,1) = lector (4)
                 nod(3,1) = lector (5)
                 nod(4,1) = lector (6)

            endif

        enddo

```

```

241   ! ahora es necesario leer las coordenadas de estos nodos. SE
      ALMACENAN EN EL VECTOR nod

      close (14)
      open (14, file="dataext.dat")

      read (14,*)

      do j = 1, numnod

251         read (14,*) lector (1), lector (2), lector (3), lector (4)

          do k = 1, 4

              if (lector (1) == nod (k,1)) then

                  nod (k,2) = lector (3)
                  nod (k,3) = lector (4)

              endif

261         enddo

          if (lector (1) == a) then

              coorda (1) = lector (3)
              coorda (2) = lector (4)

          endif

          if (lector (1) == b) then

271             coordb (1) = lector (3)
              coordb (2) = lector (4)

          endif

          enddo

      ! calculo de la localizacion de los dos puntos sonda almacenadas
      en el vector sonda

281     l(1) = coordb(1) - coorda(1)
        l(2) = coordb(2) - coorda(2)

        perp (1,1) = l(2)
        perp (1,2) = -l(1)

        perp (2,1) = -l(2)
        perp (2,2) = l(1)

291     sonda (1,1) = coorda(1) + l(1)/2 + perp (1,1)/2
        sonda (1,2) = coorda(2) + l(2)/2 + perp (1,2)/2

        sonda (2,1) = coorda(1) + l(1)/2 + perp (2,1)/2
        sonda (2,2) = coorda(2) + l(2)/2 + perp (2,2)/2

```

```

! que punto sonda esta en el interior del alabe u cual en el
interior del elemento

dist(1) = 0
dist(2) = 0
301  do j = 1, 2
        do k = 1, 4
                plus = ((sonda (j,1) - nod (k,2))**2 + (sonda (j,2) - nod
(k,3))**2)**(0.5)
                dist(j) = dist(j) + plus
        enddo
    enddo
311  if (dist(1) < dist (2)) then ! en este caso, perp2 marca el
interior del alabe (orden antihorario)
        s = -1
        else !perp 1 marca el interior del alabe (orden horario)
        s = 1
    endif
321  !!!!fin de la determinaion del sentido de la clasificacion

close (10)
open (10, file="solpNSdataext.dat")

do j = 1, a - 1
        read (10,*)
331  enddo

read (10,*) k, pa

close (10)
open (10, file="solpNSdataext.dat")

do j = 1, b - 1
        read (10,*)
341  enddo

read (10,*) k, pb ! leer presion de b del fichero de la solucion

do while (b /= -1)
!!! aqui comienza el calculo del par PARA CADA PAREJA de nodos

```

```

read (12, *)
! leer coordenadas de a y de b del archivo de filtrados
351
m = 0
n = 0
p = 0

do while (m == 0)

    read (12,*) lector (1), lector (2), lector (3)

    if (lector (1) == a) then
361
        n = 1
        coorda (1) = lector (2)/1000
        coorda (2) = lector (3)/1000

    endif

    if (lector (1) == b) then

371
        p = 1
        coordb(1) = lector (2)/1000
        coordb(2) = lector (3)/1000

    endif

    if ((p == 1) .and. (n == 1)) m = 1

enddo

381
! fin lectura de las coordenadas de las parejas de nodos y
! comienzo del calculo de los parametros

l(1) = (coordb(1) - coorda(1))
l(2) = (coordb(2) - coorda(2))
modl = ( l(1)**2 + l(2)**2 ) **(0.5)

modFi = modl * (pa + 0.5*(pb-pa))

if (s == 1) then
391
    unitFi (1) = l(2) / modl
    unitFi (2) = -l(1) / modl

endif

if (s == -1) then

    unitFi (1) = -l(2) / modl
    unitFi (2) = l(1) / modl
401

endif

```

```

Fi (1) = modFi * unitFi (1)
Fi (2) = modFi * unitFi (2)

req = (0.5 * pa + 0.333333333333 * (pb - pa))/(0.5 * (pb + pa))

if ((pa == 0).and.(pb == 0)) req = 0.5

411 ri (1) = coorda (1) + l (1) * req
    ri (2) = coorda (2) + l (2) * req

Mi = (ri (1) * Fi (2) - Fi (1) * ri (2))

Malabe (i) = Malabe (i) + Mi

!nueva lectura de la siguiente pareja

421 a = b
    pa = pb
    read (11,*) b

    close (10)
    open (10, file="solpNSdataext.dat")

    do j = 1, b - 1

431     read (10,*)

    enddo

    read (10,*) k, pb ! leer presion de b del fichero de la
solucion

    close (12)
    open (12, file="filtrados.dat")

    enddo

441 ! nueva disposcion del archivo dataext.dat a la altura de la
conectividad para el comienzo del bucle con el proximo alabe

    close (14)
    open (14, file="dataext.dat")
j = 1 + numnod + 1 + 1 + numnodccvelu + 1 + 1 + numnodccvelv + 1 +
1 + numnodccp + 1 + 1 + 1

    do k = 1, j

451     read (14,*)

    enddo

Mtotal = Mtotal + Malabe (i)

print *, Malabe (i)

```

```

enddo

Mtotal = abs (Mtotal)
461 ! ENERGIA DEL FLUJO A LA ENTRADA (POTENCIA UTIL)

! en la seccion de entrada a la TM se imponen las condiciones de
! contorno en velocidad de forma uniforme
! por lo que el caudal tambien es uniforme en el dominio existente
! entre cada pareja de nodos (Qi),
! y ademas el valor Qi asociado a cada pareja es el mismo para todas
! ellas
! esto no es asi en la salida de la TM

471

! lectura de los nodos de la entrada (malla interior con condicion de
! contorno en velocidad distinta de cero)
! y clasificado en orden mediante la conectividad en el vector
! "nodosentrada"

close (14)
open (14, file="dataext.dat")

481 do i = 1, numnod + 3

    read (14,*)

enddo

j = 1

do i = 1, numnodccvelu
491    read (14,*) lector (1), lector (2)

    if (lector (2) /= 0) then

        auxiliar (j) = lector (1) ! se guardan los nodos en el vector
        auxiliar porque todavia no estan ordenados
        j = j + 1

    endif

enddo

501 numnodent = j - 1

do i = 1, 1 + 1 + numnodccvelu + 1 + 1 + numnodccp + 1 + 1 + 1

    read (14,*)

```



```

    enddo

    k = 1
511  do i = 1, numelem

        read (14,*) lector (1), lector (2), lector (3), lector (4), lector
        (5), lector (6)

        s = 0

        do j = 1, numnodent

            if (auxiliar (j) == lector (3)) then
521                conecent (k,s+1) = lector (3)
                    s = s + 1

            endif

            if (auxiliar (j) == lector (4)) then

                conecent (k,s+1) = lector (4)
                    s = s + 1
531            endif

            if (auxiliar (j) == lector (5)) then

                conecent (k,s+1) = lector (5)
                    s = s + 1

            endif

541            if (auxiliar (j) == lector (6)) then

                conecent (k,s+1) = lector (6)
                    s = s + 1

            endif

        enddo

        if (s == 2) k = k + 1
551    enddo

    next = conecent (1,1)
    conecent (1,1) = 0

    do j = 1, numnodent

        s = 0
561        i = 0

```

```

do while (s == 0 )
    i = i + 1
    if (conecent (i,1) == next) then
        nodosentrada (j,1) = conecent (i,1)
        next = conecent (i,2)
571     conecent (i,2) = 0
        s = 1
    endif
    if (conecent (i,2) == next) then
        nodosentrada (j,1) = conecent (i,2)
        next = conecent (i,1)
581     conecent (i,1) = 0
        s = 1
    endif
enddo

enddo

nodosentrada (numnodent + 1,1) = nodosentrada (1,1)
591

do i = 1, numnodent + 1 ! inclusion en el vector nodosentrada la
    posicion de cada nodo
    ! inclusion en el vector nodosentrada la
    solucion de presion

    close (10)
    close (14)
    open (14, file="dataext.dat")
    open (10, file="solpNSdataext.dat")

601     k = nodosentrada (i,1)

    do j = 1, k

        read (14,*)

    enddo

    read (14,*) lector (1), lector (2), lector (3), lector (4)

611     nodosentrada(i,2) = lector (3)/1000
        nodosentrada(i,3) = lector (4)/1000

    do j = 1, k - 1

```

```

        read (10,*)

    enddo

    read (10,*) lector (1), lector (2)
621   nodosentrada(i,4) = lector (2)

enddo

!!!! fin del clasificado

modVent = caudal / (2 * 3.141592 * rint)

631   ! calculo de la energia del fluido a la entrada asociada a la
      presion

    PotEstEnt = 0

    do i = 1, numnodent

        l(1) = nodosentrada (i+1,2) - nodosentrada (i,2)
        l(2) = nodosentrada (i+1,3) - nodosentrada (i,3)

        modl = ( l(1)**2 + l(2)**2 ) **(0.5)

641   plus = 0.5 * modVent * modl * (nodosentrada (i+1,4) +
      nodosentrada (i,4))

        PotEstEnt = PotEstEnt + plus

    enddo

    ! calculo de la energia cinetica del fluido a la entrada

651 PotCinEnt = 3.141592 * rho * rint * ( modVent ** 3 )

! ENERGIA DEL FLUJO A LA SALIDA (POTENCIA UTIL)

661   ! lectura e identificacion de los nodos de la salida con la
      condicion de que deben ser los nodos cuya coordenada x mas
      "negativa"
      ! y clasificado en orden mediante la coordenada vertical en el
      vector "nodossalida"

    close (16)
    open (16, file="dataconj.dat")

```

```

    read (16,*) numnodconj

    minx = 0
671 do i = 1, numnodconj

        read (16,*) lector (1) , lector (2), lector (3)

        aux = minx
        minx = min (lector(3), aux)

    enddo

    close (16)
681 open (16, file="dataconj.dat")

    read (16,*)

    j = 0

    do i = 1, numnodconj

        read (16,*) lector (1) , lector (2), lector (3), lector (4)
691     if (lector (3) < (minx + 0.0003)) then

            j = j + 1
            auxiliar2 (j,1) = lector (1)
            auxiliar2 (j,2) = lector (3)
            auxiliar2 (j,3) = lector (4)

        endif

    enddo

701 numnodsals = j

    ! una vez identificados se pueden ordenar por su coordenada y

    do i = 1, numnodsals

        miny = 1000

        do j = 1, numnodsals
711         miny = min (miny, auxiliar2 (j,3))
            if (miny == auxiliar2 (j,3)) k = j

        enddo

        nodossals (i,1) = auxiliar2 (k,1)
        nodossals (i,2) = auxiliar2 (k,2)
        nodossals (i,3) = auxiliar2 (k,3)

721     auxiliar2 (k,3) = 1001

```

```

enddo

! solo queda extraer la informacion de los archivos solucion de los
  nodos ordenados

do i = 1, numnodsalsal

  z = nodossalida (i,1)

731  close (17)
     close (19)
     close (20)

     open (17, file="solpNS.dat")
     open (19, file="soluNS.dat")
     open (20, file="solvNS.dat")

741  do j = 1, z - 1

         read (17,*)
         read (19,*)
         read (20,*)

     enddo

     read (17,*) aux, nodossalida (i,4)
     read (19,*) aux, nodossalida (i,5)
751  read (20,*) aux, nodossalida (i,6)

     nodossalida (i,5) = - nodossalida (i,5)

enddo

!!!!!!! fin del clasificado

calculo de la energia del fluido a la salida asociada a la presion y
  a la velocidad

761  PotEstSal = 0
     PotCinSal = 0

do i = 1, numnodsalsal - 1

     l(1) = nodossalida (i+1,2) - nodossalida (i,2)
     l(2) = nodossalida (i+1,3) - nodossalida (i,3)

     modl = ( l(1)**2 + l(2)**2 ) **(0.5)

771  du = nodossalida (i+1,5) - nodossalida (i,5)
     dv = nodossalida (i+1,6) - nodossalida (i,6)
     dp = nodossalida (i+1,4) - nodossalida (i,4)

     ui = nodossalida (i,5)
     vi = nodossalida (i,6)

```

```

pi = nodossalida (i,4)

plus = modl * (dp*du/3 + 0.5*ui*dp + 0.5*pi*du + ui*pi)

781 PotEstSal = PotEstSal + plus

plus = 0.5*rho*modl * (0.25*du*dv*dv + dv*dv*ui/3 + 2*dv*du*vi/3 +
dv*vi*ui + 0.5*du*vi*vi + vi*vi*ui)
aux = 0.5*rho*modl * (0.25*du*du*du + du*du*ui + 1.5*du*ui*ui +
ui*ui*ui)

PotCinSal = PotCinSal + plus + aux

enddo

791 ! EFICIENCIA GLOBAL Y ESTATICA

efiglobal = ((PotEstSal + PotCinSal) - (PotEstEnt +
PotCinEnt))/(Mtotal * w)

efiestatica = (PotEstSal - (PotEstEnt + PotCinEnt))/(Mtotal * w)

write (21,*) "Eficiencia global", efiglobal
write (21,*) "Eficiencia estatica", efiestatica
write (21,*) "Par mecanico", Mtotal
write (21,*) "Potencia inyectada", Mtotal * w
801 write (21,*) "Salto de potencia util",
(PotEstSal + PotCinSal) - (PotEstEnt + PotCinEnt)
write (21,*) "Potencia fluido en entrada", PotEstEnt + PotCinEnt,
"asociada a la presion", PotEstEnt, " y cinetica", PotCinEnt
write (21,*) "Potencia fluido en salida", PotEstSal + PotCinSal,
"asociada a la presion", PotEstSal, " y cinetica", PotCinSal

endprogram

```

## B. Código fuente algoritmo de cálculo del error. LDC

```
1 program calculoError

integer nodosuP, nodosvP, nodosuR, nodosvR, i, j, k
real sumaUdif, sumaUref, a
real uPropio(1000,2), vPropio(1000,2), uRef(1000,2), vRef(1000,2),
    uAux(1000,2), vAux(1000,2), aux (1000,2)

open (10, file="uPropio.dat")
open (11, file="vPropio.dat")
open (12, file="uRef.dat")
open (13, file="vRef.dat")
11

!!! Muy importante, todos los archivos de entrada deben estar
ordenados de menor a mayor en su coordenada espacial, ademas, la
coordenada espacial debe ser la primera de las dos en todos los
archivos;      (1) x o y      (2) v o u !!

!en primer lugar, leer la informacion y almacenarla en vectores

j = 0

do i=1,1000
21    read(10,*,END=1) uPropio(i,1), uPropio(i,2)
    j = j + 1

end do

1 nodosuP = j

31
j = 0

do i=1,1000

    read(11,*,END=2) vPropio(i,1), vPropio(i,2)
    j = j + 1

end do

41 2 nodosvP = j

j = 0

do i=1,1000

    read(12,*,END=3) uRef(i,1), uRef(i,2)
```

```

51     j = j + 1

end do

3  nodosuR = j

j = 0

do i=1,1000
61     read(13,*,END=4) vRef(i,1), vRef(i,2)
        j = j + 1

end do

4  nodosvR = j

71 ! automatizacion del orden de los archivos propios. Siempre vienen del
    codigo con la coordenada espacial a la izquierda, pero a veces en
    orden decreciente

if (uPropio (1,1) == 1) then

    do i = 1, nodosuP

        aux (i,1) = uPropio (i,1)
        aux (i,2) = uPropio (i,2)
81     enddo

    do i = 1, nodosuP

        uPropio(i,1) = aux (nodosuP-i+1,1)
        uPropio(i,2) = aux (nodosuP-i+1,2)

    enddo

91 endif

if (vPropio (1,1) == 1) then

    do i = 1, nodosvP

        aux (i,1) = vPropio (i,1)
        aux (i,2) = vPropio (i,2)

101    enddo

    do i = 1, nodosvP

```



```

vPropio(i,1) = aux (nodosvP-i+1,1)
vPropio(i,2) = aux (nodosvP-i+1,2)

    enddo

endif
111

!crear los perfiles auxiliares con las coordenadas espaciales del
    perfil con mas resolucio n , primero para u y luego para v

j = 1
sumaUdif = 0
sumaUref = 0
sumaVdif = 0
sumaVref = 0

121 uAux (1,1) = uPropio (1,1)
    uAux (1,2) = uPropio (1,2)
    print *, uAux (1,1), uAux (1,2)

if (nodosuR > nodosuP) then ! el fijo pasa a ser el de referencia.
    Coordenadas espaciales del auxiliar con la referencia y valores de
    u del auxiliar con el propio. Se modifica el perfil propio

        do i = 2, nodosuR

            do while (uRef(i,1) > uPropio(j,1))

131                j = j + 1

            enddo

            uAux(i,1) = uRef(i,1)
            uAux(i,2) = uPropio(j-1,2) +
            (uPropio(j,2)-uPropio(j-1,2))/(uPropio(j,1)-uPropio(j-1,1)) *
            (uAux(i,1)-uPropio(j-1,1))

            enddo

141        do i = 1, nodosuR - 1 ! aqui se calcula ya ambos sumandos
            correspondientes a u(y)

                sumaUdif = sumaUdif + 0.25*(abs(uRef(i+1,2) - uAux(i+1,2))
            + abs(uRef(i,2) - uAux(i,2)))*(uRef(i+1,1) - uRef(i,1))

            enddo

151        else ! el fijo pasa a ser el propio. Coordenadas espaciales del
            auxiliar con el propio y valores de u del auxiliar con el de
            referencia. Se modifica la referencia

```

```

do i = 2, nodosuP

do while (uPropio(i,1) > uRef(j,1))

j = j + 1

enddo

161      uAux(i,1) = uPropio(i,1)
      uAux(i,2) = uRef(j-1,2) +
(uRef(j,2)-uRef(j-1,2))/(uRef(j,1)-uRef(j-1,1)) *
(uAux(i,1)-uRef(j-1,1))

enddo

do i = 1, nodosuP - 1 ! aqui se calcula ya ambos sumandos
correspondientes a u(y)

      sumaUdif = sumaUdif + 0.25*(abs(uAux(i+1,2) -
uPropio(i+1,2)) + abs(uAux(i,2) - uPropio(i,2)))*(uAux(i+1,1) -
uAux(i,1))

171      enddo

end if

vAux (1,1) = vRef (1,1)
vAux (1,2) = vRef (1,2)
print *, vAux (1,1), vAux (1,2)

181 j = 1

if (nodosvR > nodosvP) then ! el fijo es el de referencia. Coordenadas
espaciales del auxiliar con la referencia y valores de v del
auxiliar con el propio. Se modifica el perfil propio

do i = 2, nodosvR

do while (vRef(i,1) > vPropio(j,1))

j = j + 1

191      enddo

      vAux(i,1) = vRef(i,1)
      vAux(i,2) = vPropio(j-1,2) +
(vPropio(j,2)-vPropio(j-1,2))/(vPropio(j,1)-vPropio(j-1,1)) *
(vAux(i,1)-vPropio(j-1,1))

enddo

```

```

201         do i = 1, nodosvR - 1 ! aqui se calcula ya ambos sumandos
correspondientes a u(y)

                sumaVdif = sumaVdif + 0.25*(abs(vRef(i+1,2) - vAux(i+1,2))
+ abs(vRef(i,2) - vAux(i,2)))*(vRef(i+1,1) - vRef(i,1))

                enddo

211     else ! el fijo pasa a ser el propio. Coordenadas espaciales del
auxiliar con el propio y valores de u del auxiliar con el de
referencia. Se modifica la referencia

                do i = 2, nodosvP

                        do while (vPropio(i,1) > vRef(j,1))

                                j = j + 1

                        enddo

221                vAux(i,1) = vPropio(i,1)
                vAux(i,2) = vRef(j-1,2) +
(vRef(j,2)-vRef(j-1,2))/(vRef(j,1)-vRef(j-1,1)) *
(vAux(i,1)-vRef(j-1,1))

                enddo

                do i = 1, nodosvP - 1 ! aqui se calcula ya ambos sumandos
correspondientes a u(y)

                        sumaVdif = sumaVdif + 0.25*(abs(vAux(i+1,2) -
vPropio(i+1,2)) + abs(vAux(i,2) - vPropio(i,2)))*(vAux(i+1,1) -
vAux(i,1))

231                enddo

        end if

        print *, "Error absoluto = ", sumaVdif + sumaUdif
        print *, sumaVdif, sumaUdif

241     endprogram

```