POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

An Automated CFD Workflow for Mating Gears Design Optimization

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria Aeronautica

Alessandro Tommasi, 10700642

Abstract: Geared systems interacting with a working fluid are critical components across several industries, from aerospace to petrochemicals, with external gear pumps representing a prominent example. While traditional numerical investigations have historically relied on analytical formulations or lumped-parameter models for their computational efficiency, these approaches often fail to capture the three-dimensional coupling of complex physical mechanisms. In contrast, Computational Fluid Dynamics (CFD) provides a robust framework to investigate localized phenomena such as internal leakage, cavitation, and aeration, which become increasingly relevant as designs evolve toward higher compactness and rotational speeds.

This work presents a fully automated CFD workflow specifically developed to address the challenges posed by the continuous deformation of the internal geometry. To manage the dynamic domain, a hybrid strategy combining mesh morphing and mesh replacement is adopted, prioritizing simulation accuracy while maintaining a balanced computational cost, and overcoming the limitations of simpler approaches such as the Multiple Reference Frame and non-conservative overlapping grid techniques.

The developed routine automates gear geometry generation from design parameters and employs the *snappyHexMesh* utility to produce surface-conforming grids. Topology changes are handled through the *meshToMesh* utility, enabling parallel field remapping and the cyclic reuse of mesh sequences, significantly reducing preprocessing efforts.

The methodology is validated through comparison between the computed resistant torque and experimental measurements, showing excellent agreement and accurately reproducing key physical phenomena, including backflow and centrifugal pressure effects. Finally, to better replicate real operating conditions and demonstrate the flexibility of the proposed framework, the workflow is extended to multiphase simulations using a Volume of Fluid approach implementing Hirt's sub-grid aeration model.

Advisor:
Prof. Federico Piscaglia

Co-advisors:
Prof. Franco Concli

Academic year:
2024-2025

Key-words: External gear pumps, Gearbox, Mating gears, CFD, Mesh generation, Dynamic mesh

1. Introduction

This project focuses on simulating the flow about two or more mating gears, where the motion between the teeth leads to trapping and transport of fluid through the device body. This kind of apparatus is versatile and capable of handling a wide range of fluids, pressures, viscosity, and flow rates, and can be found in many engineering applications, such as gearboxes, lubricated transmissions, and external gear pumps. Consequently, research and development in this field aims to improve performance, reduce cost, and enhance reliability; goals that are increasingly supported through the application of Computational Fluid Dynamics. The focus of this research project is the development of an efficient, automated and user-friendly routine for mesh generation and motion, to discretize the problem.

Systems based on mating gears interacting with a working fluid are widely used across various industries, including petrochemical, food processing, and aerospace. In the aerospace sector, geared systems play a key role in power transmission, lubrication circuits, and hydraulic actuation, where reliability is a critical requirement. External gear pumps represent a relevant example within this broader class of devices and are appreciated for their relatively low cost compared to other rotary displacement machines, their ability to operate with relatively unfiltered hydraulic fluids, and the option to combine several units with different fixed displacement volumes on a single shaft. However, they also exhibit disadvantages such as a high level of flow rate ripple, elevated noise levels, a fixed displacement volume and limited repairing possibilities. Compared to internal gear pumps, external ones can achieve higher pressures.

CFD offers powerful tools for studying phenomena such as internal leakage, cavitation, aeration, lubrication, heat dissipation and fluid-structure interactions, particularly in regions where high pressure gradients and tight clearances are present. Furthermore, as the trend moves toward more compact geared systems, maintaining the same transmitted power or flow rate requires higher gear angular velocities, which in turn increases the relevance of working fluid aeration and cavitation. Additionally, CFD helps to streamline and reduce the costs of the design process, reducing the number of experimental campaigns and allowing quick design changes along the way.

However, systems involving mating gears pose substantial challenges for CFD simulations. The rotational movement of the gears means that the internal fluid domain changes continuously, which necessitates the use of mesh motion strategies. Moreover, when such devices operate under high loads or pressures, the simulations need to account for fluid compressibility, heat exchange and the intrinsic unsteadiness of the problem, increasing the computational cost. The investigation of cavitation and aeration requires multiphase and multi-fluid models, further increasing complexity.

Dynamic mesh handling, or mesh motion, is therefore a key aspect to accurately simulate flows in geared systems. This process involves continuously adapting the computational domain to follow the rotation of the gears, whose motion alters the geometry over time. This procedure offers a closer representation of the underlying physics of the problem compared to methods such as the Multiple Reference Frame (MRF) approach, which is widely employed, for example, in turbomachinery analysis. Capturing the small clearances between the gear teeth and the surrounding casing, as well as the steep pressure gradients that arise, requires a fine spatial resolution, increasing the computational cost. Several approaches exist to tackle the challenge of mesh motion.

This work focuses on a strategy that combines mesh morphing, i.e. the continuous deformation of the mesh as the gears rotate, and mesh replacement, where a new mesh is introduced when the old one becomes too distorted. This hybrid approach aims to balance simulation accuracy and computational efficiency, enabling the handling of complex gear geometries in a modular and automated way. Although this strategy is already in use, it typically requires an elaborate pre-processing phase, whose simplification is the objective of the present work. Starting from the geometric parameters provided by the user, such as normal module, pressure angle, tip diameter, etc., the entire geometry can be reconstructed using its periodic nature; in addition, the motion properties must be defined as input. The aim is to quickly generate a base mesh and then manage its motion through an optimized dynamic mesh handling algorithm. When the mesh quality deteriorates due to excessive deformation, a topological mesh change is triggered. This involves switching to a new mesh that better conforms to the updated geometry and transferring the solution fields from the previous mesh to the new one via remapping algorithms.

This work aims to improve both the mesh motion strategy and the performance of the topological switching process, including the way memory is accessed and managed during these transitions, together with the implementation of a fully automated mesh generation routine. Historically, fluid-filled gear systems, and gear pumps in particular, have been the subject of significant scientific and industrial interest, as their accurate performance prediction has always represented a major research challenge. Early numerical investigations primarily relied on simplified one-dimensional or semi-empirical models, occasionally extended to two-dimensional analyses focusing on specific phenomena such as cavitation, aeration, or pressure fluctuations. Nevertheless, these effects are inherently coupled and strongly influenced by three-dimensional geometrical features, including leakage through side clearances. With the advent of high-performance computing, it has become feasible to perform

fully three-dimensional simulations capable of capturing the complex interactions among these phenomena with much greater fidelity.

2. State of the art

In this chapter, an overview of the state of the art concerning the methods employed in the simulation of flow in mating gears is provided. The discussion begins with the earliest approaches, which relied on analytical or semi-empirical models, proceeds with more advanced techniques, and concludes with the presentation of the meshing strategies currently adopted in the CFD field.
As already mentioned, the use of CFD in the study of mating gears has been gaining increasing attention. Nevertheless, numerical approaches have long played a central role, spanning from simple analytical formulations to fully detailed three-dimensional simulations. The first attempts to describe the phenomena associated with gear motion were based on analytical models calibrated against experimental data. Concli and Gorla [4] reviewed the methods developed to investigate power losses due to churning and pocketing, highlighting several analytical models derived from experimental observations. Among these, the Ohlendorf method, later refined into its most complete version, known as the Mauz formulation, represents the first of its kind. More recent approaches include the Changenet and Velex formulation, as well as the ISO/TR 14179-1 model. Studies on windage power losses have also been proposed, one of the earliest being the work of Anderson and collaborators. In all these cases, the outcomes are expressed as equations for power loss or resisting torque, obtained by fitting analytical models to experimental results [8].
Devendran and Vacca [6] introduced an original analytical method for investigating variable displacement gear pumps. This approach is extremely computationally efficient and capable of providing accurate, though limited, predictions, since it only describes flow rate and torque as functions of time. While such a method proves valuable during preliminary design or sizing phases, it offers no insights into the underlying physical mechanisms.
What clearly emerges from the literature is that these analytical or semi-empirical approaches are significantly more computationally efficient compared to CFD-based methods. However, they generally address only a restricted set of phenomena, neglecting their intrinsic interdependence, and are applicable within a narrow range of conditions, being strongly constrained by the limited experimental datasets on which they are based [8]. Consequently, they often suffer from reduced accuracy and are best suited for preliminary assessments or approximate design studies.
The subsequent stage in the evolution of modeling approaches is represented by lumped parameter models [1, 14, 21]. In these methods, the fluid domain is discretized into control volumes (CVs) with variable spatial resolution, typically dependent on the specific case under study but in any case far coarser than that employed in conventional CFD. Each control volume is treated as an open thermodynamic system, capable of exchanging mass and energy with its surroundings. The mass flow between adjacent control volumes is generally evaluated through a variable orifice equation, whereas the pressure, considered as a function of time only, is obtained by integrating a mass conservation ordinary differential equation (ODE) together with a fluid equation of state.
The fundamental distinction of this approach lies in replacing partial differential equations (PDEs) with simplified ODEs, leading to a single representative value for each variable associated with the entire control volume. In lumped parameter models, the inlet and outlet regions are typically represented by a single control volume each; a natural evolution of these methods is their coupling with one-dimensional approaches, which allow these regions to be discretized into a greater number of control volumes. This refinement enables the inclusion of friction and inertia effects, thus making it possible to simulate pressure wave propagation and evaluate pressure ripple [21]. The usage of lumped parameters has proven effective in predicting global quantities such as average pressures, flow rate, and volumetric efficiency. However, it does so at the expense of spatial resolution, with localized phenomena such as cavitation and leakage being incorporated only through dedicated sub-models.
In the following part of this review, the focus shifts to CFD-based models for the study of geared systems, that is, numerical approaches based on the solution of the Navier–Stokes (N–S) equations. Within the CFD framework, several methodologies can be employed, and here some of the most widely adopted strategies are presented. A first broad distinction can be drawn between mesh-based and meshless techniques. In meshless approaches, the spatial domain is not discretized into cells but is instead represented through a set of nodes, often referred to as particles, within a Lagrangian reference frame. The main advantage of such methods is the complete avoidance of mesh generation and dynamic mesh handling [11, 13].
Among meshless approaches, the most widely used is Smooth Particle Hydrodynamics (SPH) [13]. In SPH, the properties of each particle are computed as weighted sums of the corresponding properties of surrounding nodes, belonging to a predefined region of influence, with the weights determined by the relative distance. This method has demonstrated broad applicability to complex fluid systems; however, its effectiveness has proven greater in predicting lubricant flow patterns rather than in providing accurate flow field quantities [4, 13].
Despite these advantages, the majority of CFD applications rely on mesh-based models. In such approaches, the fluid domain is discretized into finite CVs, within which the governing equations are solved. The N–S equations,

being partial differential equations in conservation form, are particularly well suited to the finite-volume (FV) method, which has thus become the most widely adopted approach in commercial solvers. The remainder of this section will therefore refer exclusively to FV-based methods.

Within the FV framework, two reference frames may be employed: Lagrangian and Eulerian. In the Lagrangian description, the computational cells move following the fluid motion, whereas in the Eulerian approach their position remains fixed in space. An intermediate formulation, known as Arbitrary Lagrangian–Eulerian (ALE), has also been developed, aiming to combine the advantages of both approaches [7, 20]. In ALE methods, computational nodes can either remain stationary, as in the Eulerian framework, or move consistently with the continuum, as in the Lagrangian one. This hybrid strategy provides the ability to adapt to domain deformations while retaining the higher accuracy typically associated with a Lagrangian representation [7].

In practice, within the FV framework, the purely Eulerian description is by far the most widely used. Consequently, in the remainder of this review, the discussion will focus on the most common Eulerian-based techniques for the numerical simulation of mating gears, highlighting their respective trade-offs in terms of computational cost and accuracy.

The simplest CFD strategies for geared systems simulation employ a static mesh. The most basic implementation prescribes a velocity boundary condition either on the gears or directly on the fluid, accounting for inertial effects. This approach is typically used to investigate steady phenomena arising during gear rotation [13]. An evolution of this technique is the Multi-Reference Frame (MRF) method, which still relies on a fixed mesh but introduces separate rotating reference frames for each gear [13]. These methods are computationally efficient but provide limited accuracy.

Higher-fidelity simulations require moving-mesh techniques, which enable the actual gear geometry to be represented during rotation, allowing the capture of transient effects. The Sliding Mesh (SM) method encloses the rotating bodies within axisymmetric zones, in which the mesh is rigidly rotated. The interface between stationary and rotating regions is handled via Arbitrary Mesh Interface (AMI) [4, 13]. The main limitation of SM is the inability to accurately represent the engagement zone, which requires mesh topology changes [8, 13]. Consequently, geometric simplifications, such as gear scaling or tooth removal, are often introduced to avoid meshing issues, as proposed by Concli and Gorla [3].

The Overlapping Grids (OG) method, also referred to as Chimera grids, represents an alternative moving-mesh strategy. Separate meshes are generated for each body and allowed to overlap within a background mesh. Cells can be classified as hole (inside a solid region), calculated (non-overlapping), or interpolated (within overlap regions). In the latter case, field values are obtained via interpolation from neighboring cells. Although OG provides greater geometric flexibility, interpolation is non-conservative and may reduce solution accuracy [4, 8, 13, 17, 18].

More advanced strategies rely on remeshing, either locally (Local Remeshing Approach, LRA) or globally (Global Remeshing Approach, GRA) [8]. In LRA, as time advances and the boundaries move, cells are displaced according to a smoothing function based on Hooke's law; when quality criteria are violated, the affected elements are removed and regenerated. In GRA, cell motion is typically governed by a Laplace smoothing equation, and once mesh quality falls below the threshold, the entire grid is regenerated [4, 13, 17].

Hybrid approaches have also been developed. Duzel et al. [12] generated structured meshes within cylindrical enclosures around each gear, which deform to represent the mating region. Only the inner regions of the enclosures are remeshed, using an iterative elliptic solver, thus avoiding full-domain regeneration. Martinez Rubio [20] proposed an alternative method where the fluid domain is divided into three regions: a fixed zone containing the inlet and outlet, and two moving regions, one per gear, that overlap in the engagement area. Unlike Chimera grids, the solution in the overlapping regions is not obtained via interpolation; instead, the two meshes are projected onto an interface line (or surface in 3D) computed at each time step, eliminating mesh superposition.

Additional criteria also influence the choice of the most suitable method, such as mesh generation. In many studies, gear geometries are derived from CAD models and subsequently processed through commercial software [18] or automated mesh generation tools such as *snappyHexMesh* in OpenFOAM [23]. Other works rely on point lists or algorithms to directly define the gear profile, significantly streamlining and accelerating the meshing process compared to CAD-based workflows [20].

Another critical aspect concerns the treatment of the contact point. In real applications, the clearance between gears is zero, as one gear drives the other. This represents a major challenge for dynamic mesh handling. In theory, only the Overlapping Grid (OG) method can accurately reproduce the zero-clearance contact point [17]. Other techniques often introduce an artificial kinematic viscosity contribution, gradually increasing toward the contact region to suppress fluid penetration between the teeth [20]. Orlandi et al. [18] compared OG and mesh-replacement techniques against experimental results, concluding that the latter is more effective for predicting global pump performance and cavitation, while OG provides higher accuracy in reproducing pressure evolution within the trapped fluid volume between gear teeth, owing to its treatment of zero-clearance contact.

The final aspect examined in this review concerns the treatment of local phenomena, with particular emphasis on aeration. Aeration refers to the entrainment of air at the oil–air free surface and arises as a consequence of

flow turbulence. This phenomenon leads to increased power losses and demonstrates that considering only the lubricant density and viscosity is insufficient for an accurate estimation of power dissipation; moreover, aeration significantly affects the lubricating properties of the oil [15, 16].
Simplified approaches, such as lumped-parameter models, can account for aeration through dedicated sub-models. In the most widely adopted implementations, the fraction of separated air is evaluated by assuming instantaneous separation and dissolution processes governed by Henry's law, together with a constant total gas content throughout the hydraulic circuit. More advanced lumped-parameter models extend this framework by introducing gas transport between capacitive elements and non-instantaneous release and dissolution dynamics; however, their application remains limited [21].
Building upon these concepts, the model proposed by Shah et al. [22] introduces a fully dynamic formulation in which the temporal evolution of gas and vapor phases directly influences both pressure build-up and flow through hydraulic connections. This approach overcomes the limitations associated with static equilibrium assumptions and enables a more realistic prediction of inlet pressure dynamics and incomplete filling phenomena in gear pumps. However, as already mentioned, lumped-parameter models are characterized by poor spatial resolution, which also limits their ability to accurately capture aeration phenomena.
CFD-based models allow the non-homogeneous distribution of air within gears chambers to be resolved and enable the identification of regions where aeration and cavitation occur. However, their application requires advanced fluid models characterized by a large number of parameters, whose estimation under operating conditions is often challenging [21]. In Sec. 5, air entrainment is implemented based on the model proposed by Hirt [10], which has already been adopted by Mastrone and Concli [8, 16].

3. Methodology

From the analysis presented in the previous chapter, it emerges that CFD methods often involve a trade-off between the accuracy of the approach and the complexity of the pre-processing stage, for instance, methods such as Chimera grids offer simplified pre-processing but with the drawback of a lack of accuracy, whereas there are other approaches involving a complex pre-processing phase, as an example the necessity of creating several meshes, but on the contrary provide higher accuracy. In this project, a method of the latter type has been chosen, in order to prioritize the accuracy of the results while seeking the simplification of the pre-processing stage as much as possible.
The starting point of this work is an outdated set-up, originally developed for older versions of OpenFOAM, which relies on the mesh morphing and replacement method anticipated in the introduction (Sec. 1). This configuration is managed by a `bash` script that merely executes an ordered sequence of simulations, each characterized by its own mesh topology and by the use of mesh morphing as the computation progresses. Once a simulation is completed, the results of its final iteration are employed as the initial conditions for the subsequent one through a remapping algorithm.
The first and most critical aspect of this set-up is that the geometry definition and mesh generation of the gears are handled externally. This approach, renders the overall process less streamlined and requires more frequent user intervention during the pre-processing stage. As discussed in the literature review (Sec. 2), many methodologies depend on commercial softwares either to generate the computational grid or to obtain a CAD representations of the geometry. In contrast, the present work aims to develop a fully automated routine capable of constructing a 2D or 3D geometry directly from the geometrical parameters of the gears, and subsequently producing a mesh with adequate quality metrics.
Secondary aspects, yet still relevant in the overall streamlining of the methodology, are related to the memory access when the topology is changed and to the post-processing phase. The access to the mesh files is not optimized, during the remapping stage both meshes, the old one and the new one, must be simultaneously available. Consequently, at the start of each new simulation, the new grid must be loaded while the previous one must be reloaded, since it has been cleared from memory after the preceding run is concluded, this aspect is treated more in detail later in (Sec. 3.4). Regarding the second aspect, the overall output is not provided as a single continuous simulation but rather as multiple separate runs, which complicates the post-processing phase, in the new version the results are available in the form of a unique simulation.
A key improvement to streamline this process is the adoption of the *meshToMesh* topology changer, a class introduced in OpenFOAM v6 and further developed in later versions. This utility enables the mapping of field variables onto a new mesh or sequence of meshes at specified time instants. Moreover, it allows the meshes to be cyclically remapped, which is particularly suited for symmetric forward–reverse motion or multi-cycle engine simulations, as is precisely the case in the present study. Indeed, the sequence of meshes employed here describes the meshing phase between a pair of gear teeth, a phenomenon that recurs after a rotation of $2\pi/Z_p$, where Z_p denotes the number of teeth of the pinion.

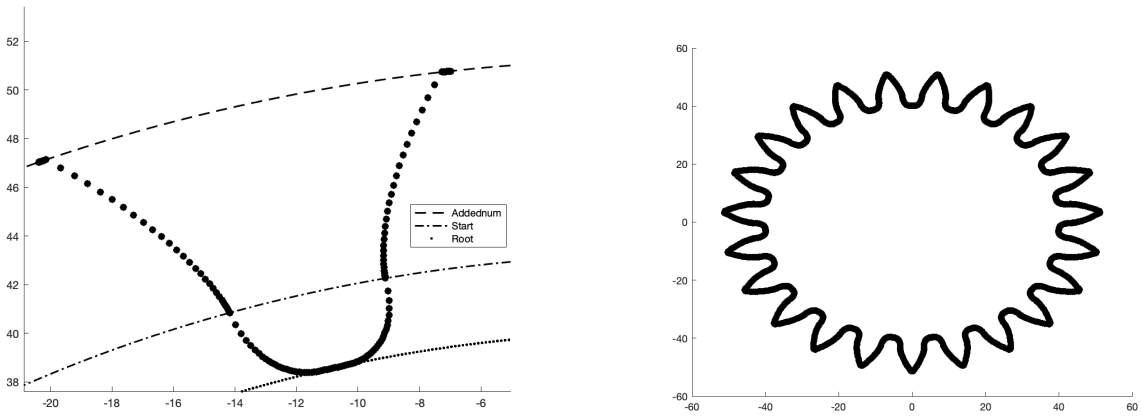
3.1. Gear generation

Before addressing the actual mesh generation, the gear geometry must be constructed starting from a set of fundamental design parameters, such as the number of teeth, the normal module, and the pressure angle. To this end, a dedicated in-house application is employed. The application takes the above-mentioned parameters as inputs and generates the geometry of a single tooth by describing its profile through a finite set of points expressed in polar coordinates. Exploiting the inherent periodicity of spur gears, the complete wheel geometry is subsequently obtained by rigidly rotating this initial set of points.
The generation process begins with the definition of the tooth profile, which is subdivided into several distinct geometric portions. The tooth flank is composed of two involute-based sections, both generated as involutes of the base circle. The involute curve is parametrically defined by the scalar parameter q, which represents the angular displacement of the tangent line unwrapped from the base circle. The Cartesian coordinates of the involute are obtained from the base radius R_b using the classical involute equations:

$$\begin{cases} x(q) = R_b\,(\cos q + q \sin q) \\ y(q) = R_b\,(\sin q - q \cos q) \end{cases} \qquad q(r) = \sqrt{\frac{r^{\,2}}{R_b} - 1} \tag{1}$$

The involute is subdivided into a working and a non-working region. For the working flank, the parameter q is bounded between two values, $q(r_{start})$ and $q(r_{head})$, where r_{start} corresponds to the diameter at which the involute profile is truncated in order to avoid interference with the root region, while accounting for profile shift, backlash, and chamfer effects. Conversely, the non-working portion of the involute extends from the base circle to the r_{start}, and is therefore defined over the interval $q \in [0, q(r_{start})]$. Both involute segments are discretized by uniformly sampling the parameter q within their respective intervals, yielding a finite set of points that approximate the continuous involute profile.
The tooth tip is represented by uniformly sampling an arc of the addendum circle spanning half of the total angular thickness of the tooth. Similarly, the tooth root is closed by an arc of the root (dedendum) circle, extending from the involute exit angle up to $\gamma = \pi/Z$ where Z denotes the number of teeth.
The opposite flank is obtained by symmetry. This procedure ensures a geometrically consistent representation of the complete tooth profile. At this stage, the set of points shown in Fig. 1a is obtained. The complete gear geometry is finally constructed by replicating this set of points $Z - 1$ time, each time applying an incremental angular rotation given by Eq.2, leading to Fig. 1b.

$$\theta_k = k\frac{2\pi}{Z} \qquad k = 1, ...Z - 1 \tag{2}$$



(a) (b)

Figure 1: a) Tooth profile; b) Gear profile

3.2. Mesh generation

In this paragraph the automated geometry discretization process is dealt. At this stage the gears are described by a cloud of N points (N=n·Z) each. To get a three dimensional object, the planar figure is duplicated and offset to obtain the required thickness, a stereolithography (STL) file is automatically generated resorting to a

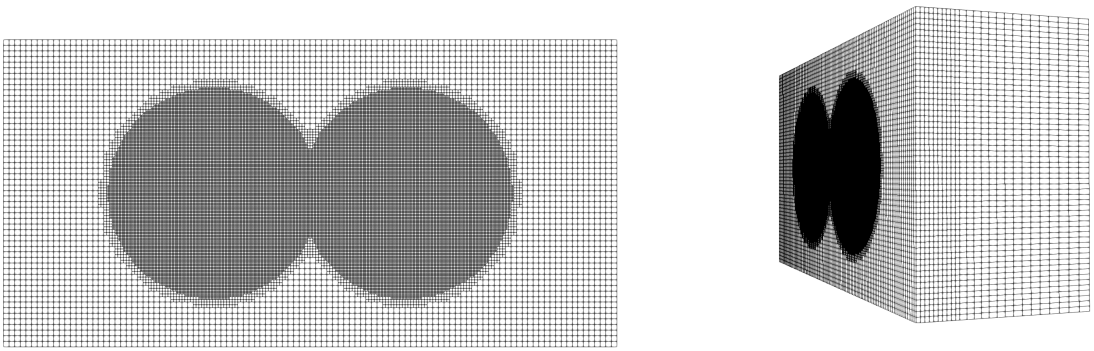
custom Python code. To get a proper pairing, one of the gears is translated, on the x-axis to the nominal center distance according to Eq. 3 to guarantee a consistent tooth–slot correspondence between the two gears.

$$\Delta_x = \frac{(m + x_1) \cdot Z_1 + (m + x_2) \cdot Z_2}{2} + \delta_x \tag{3}$$

where m is the normal module, Z_i the teeth number of teeth per each gear and δ_x an additional, user defined, axial shift. Before proceeding in the actual mesh generation, the engagement of the wheels is initialized by rotating the driving gear by an angle $\vec{\theta_p}(0)$ while the driven one by an angle $\vec{\theta_g}(0)$ according to:

$$\vec{\theta_p}(0) = -\frac{\pi}{2}\hat{z} \qquad\qquad \vec{\theta_g}(0) = -\vec{\theta_p}(0) + \frac{\pi}{N_g}\hat{z} \tag{4}$$

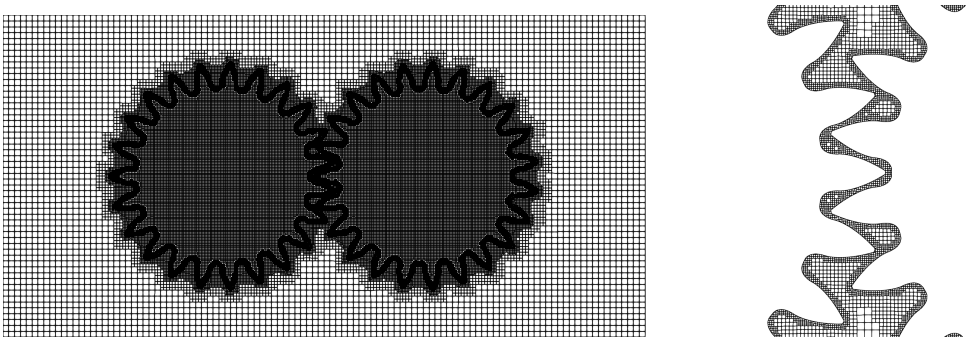
The process begins with the construction of a background mesh, generated with the *blockMesh* utility. This grid consists of hexahedral cells covering the computational domain and does not need to match the geometry. Instead, it serves as the foundational structure upon which subsequent refinement and deformation operations are performed to reproduce the desired geometry. The background mesh must nevertheless be sufficiently regular, particularly in regions expected to undergo deformation. Such mesh is composed of two blocks, one extending for the gears thickness and a second one that represents the casing which cells presents a gradually growing dimension in the z-diretion as we move toward the farfield and where a high spacial resolution is no longer required, avoiding sudden changes, which might provoke steep gradients. In the gears block, are practiced two cylindrical refinement regions extending from each center with a diameter equal to the pitch diameter of the gears. The features just explained can be observed in Fig. 2.



(a) (b)

Figure 2: Background mesh

To avoid the necessity to have very small cells in the engagement zone, in particular in correspondence of the contact point, the gears are scaled by a scale factor of 0.995, this practice is justified by the fact that is beyond the interests of this work to capture phenomena at such small length scale. The discretization for the gears is obtained through the OpenFOAM's in built utility *snappyHexMesh* [9].
snappyHexMesh is OpenFOAM's automatic mesh generator that transforms an initial hexahedral background grid into a surface-conforming computational mesh based on triangulated geometries such as STL or OBJ files. Its methodology follows a structured sequence of operations aimed at adapting a simple initial discretization to complex geometrical domains. Geometric input is then supplied in the form of triangulated surfaces and feature edges that represent sharp geometric transitions. *snappyHexMesh* uses this information to determine refinement zones and to guide the adaptation of the mesh to the underlying geometry. The first major stage, castellation, performs local refinement of the background mesh. Cells are subdivided according to their proximity to the geometry and to geometric features, producing a discretization dense enough to capture relevant surfaces. After refinement, the algorithm removes all cells lying outside the region connected to the user-defined internal point, effectively isolating the computational domain. The subsequent snapping stage adjusts the refined mesh so that cell faces align with the triangulated geometry. This deformation is carried out using quality-preserving algorithms that minimize skewness and avoid numerical degradation. The result is a hexahedral-dominant mesh that conforms to the geometry while maintaining acceptable element quality. Given the laminar-flow assumption adopted in this study (Sec. 4), the mesh was generated without the addition of surface layers. *snappyHexMesh* is designed for automation: once the configuration file is defined, the entire workflow, from initial mesh to surface-fitted result, proceeds without further user intervention, yielding a robust and reproducible mesh suitable for CFD simulations.



(a) (b)

Figure 3: a) Gears mesh; b) engagement detail.

As previously mentioned, the gear geometry is periodic; therefore, it is sufficient to describe a single tooth engagement, since the same set of meshes can be reused for all subsequent engagements. To generate this set, a list of rotation angles must be specified. These angles represent the pinion rotation with respect to the initial configuration. The list can be generated automatically by the code when a constant angular increment between consecutive meshes is required, or alternatively, a custom list of angles can be provided by the user. It is worth recalling that the angular increment should be chosen sufficiently small to ensure the preservation of mesh quality; whereas selecting an excessively large increment leads to inefficiencies, which manifest as increased mesh generation time and unnecessarily frequent topology changes. When a new mesh is generated, each STL surface is rotated by the corresponding angle, as defined in Eq. 5, and the *snappyHexMesh* utility is then executed.

$$\theta_p(\alpha) = \theta_p(0) + \alpha \qquad \theta_g(\alpha) = \theta_g(0) - \frac{N_p}{N_g} \cdot \alpha \tag{5}$$

With α belonging to the previously mentioned list of angles.
A grid convergence study was conducted to quantify the discretization uncertainty associated with the numerical solution. The analysis follows the Grid Convergence Index (GCI) methodology proposed by Celik et al.[2] and recommended by ASME for the verification of CFD simulations. This approach is based on Richardson extrapolation, whose key feature is that it can be applied not only to pointwise solution variables but also to integrated quantities, such as forces or moments [19], moreover, it can also be applied in case where refinement ratios are not constant. The assumption is that the numerical solution for a given quantity of interest ϕ can be expressed as:

$$\phi(h) = \phi_{exact} + Ch^p$$

where ϕ_{exact} is the exact (continous) solution, h is a representative grid size, C is a constant and p is the apparent order of accuracy. The value of p does not necessarily coincide with the formal order of the numerical scheme, but rather represents the effective convergence rate of the solution for the specific quantity of interest. Three systematically refined and geometrically similar grids (coarse, medium, and fine) were employed. The quantity h_i was defined for each grid level i based on the average cell size as:

$$h_i = \sqrt[3]{\frac{V}{N_i}} \tag{6}$$

where V is the domain volume, equal for all the meshes, and N_i the number of cells. The grid refinement ratios (r_{ij}) and the difference of the computed quantity (ϵ_{ij}) were defined as:

$$r_{ij} = \frac{h_i}{h_j} \qquad \epsilon_{ij} = \phi_i - \phi_j \tag{7}$$

in this case ϕ corresponds to the resistant torque acting on the gears. The following equation is obtained:

$$\frac{\epsilon_{32}}{\epsilon_{21}} = \frac{r_{21}^p - s}{r_{32}^p - s} \tag{8}$$

$$s = 1 \cdot sgn(\frac{\epsilon_{32}}{\epsilon_{21}}) \tag{9}$$

where s, in Eq. 9, is equal to +1 in case of monotonic convergence or -1 in case of oscillatory behavior. Eq. 8 is solved implicitly for p. The following quantities can now be evaluated.

$$\phi_{\text{ext}}^{21} = \frac{r_{21}^{p}\,\phi_1 - \phi_2}{r_{21}^{p} - 1} \tag{10}$$

with ϕ_{ext}^{21} being an estimation of the value of an hypothetical infinitely refined grid, based in the fine and medium refined grids.

$$e_a^{21} = \left|\frac{\phi_1 - \phi_2}{\phi_1}\right| \qquad\qquad e_{\text{ext}}^{21} = \left|\frac{\phi_{\text{ext}}^{21} - \phi_1}{\phi_{\text{ext}}^{21}}\right| \tag{11}$$

where e_a^{21} quantifies the change in the numerical solution between successive grid levels (medium to fine) and provides a measure of the solution sensitivity to grid refinement, rather than a true error with respect to the exact solution. The extrapolated relative error (e_{ext}^{21}) estimates the distance between the fine-grid solution and the asymptotic solution predicted by Richardson extrapolation, and therefore represents an estimate of the remaining discretization error on the finest grid, although it is not used directly for standard uncertainty reporting. Lastly, the extrapolated solution and the Grid Convergence Index were computed, following the procedure of Celik et al., to estimate the numerical uncertainty associated with spatial discretization.

$$\text{GCI}_{\text{fine}}^{21} = \frac{1.25\,e_a^{21}}{r_{21}^{p} - 1} \tag{12}$$

In Tab.1 are reported the specifications for the grids while in Tab.2 are presented the outcome of the application of the grid convergence study.

	Index (i)	N_i	ϕ_i	h_i
Coarse	3	317915	11.9	$1.147 \cdot 10^{-2}$
Medium	2	725918	12.65	$1.11 \cdot 10^{-2}$
Fine	1	1620196	12.95	$8.5 \cdot 10^{-3}$

Table 1: Mesh specifications.

p	ϕ_{ext}^{21}	e_{ext}^{21}	e_a^{21}	GCI_{fine}^{21}
3.266	13.1648	2.317%	1.632%	2.073 %

Table 2: Grid convergence.

The computed GCI is consistent with values commonly reported in the literature, indicating that grid convergence has been achieved for the present study. The relatively high apparent order of convergence is attributed to the integral nature of the monitored quantity, as quantities such as forces or moments are known to exhibit higher apparent convergence rates than pointwise solution variables [19].
In Alg. 1 is summarized the algorithm explained in this paragraph.

Algorithm 1 Mesh generation

1: Mesh generation
   Input: `gearsGenDict`, N mesh times
   Output: N meshes.
2: 2D geometry generation ← `gearsGen`(`gearsGenDict`)
3: 2D geometry extrusion & stereolithography generation
4: Background mesh generation ← `blockMesh`
5: **for** $i = 1 \rightarrow N$ **do**
6: Gears rotation ← $\theta_p = N(i)$
7: Features extraction
8: Mesh generation ← `snappyHexMesh`
9: **end for**

3.3. Mesh motion strategy

The motion of the gears is defined through the law of motion of the driving wheel, from which is derived the one for the driven one, of the kind:

$$\vec{\theta_p}(t) = \vec{\omega_p} \cdot t + \vec{\theta_p}(0) \qquad \vec{\theta_g}(t) = \vec{\omega_p} \cdot \frac{N_p}{N_g} \cdot t + \vec{\theta_g}(0) \tag{13}$$

therefore it is possible to know the angular position for each time instant. During the simulation for each time-step, the position of each point belonging to the gears patches is computed and imposed through a *point-Displacement* boundary condition. The data required as input are the axis and the center of rotation for each wheel and the angular velocity of the driving gear, whereas that of the driven gear can be determined from the gear ratio. The shape of the gears is defined as the mean of a cloud of points for each of which a displacement vector is applied at each time interval. The displacement vector is computed according to the Rodrigues formula as follows:

$$\vec{v} = \vec{p_0} \cdot cos(\theta) + (\hat{a} \times \vec{p_0}) \cdot sin(\theta) + \hat{a} \cdot (\hat{a} \cdot \vec{p_0}) \cdot (1 - cos(\theta)) \tag{14}$$

where $\vec{v}$ is the displacement vector, $\vec{p_0}$ is the position vector of the points, in the undisplaced mesh configuration, with respect to the center of rotation and it is initialized after each topological change, θ is the rotation angle relative to the initialized configuration and it is calculated by multiplying the angular velocity by the time elapsed from the change of topology and $\hat{a}$ represents the rotation axis versor. Once the geometry is updated, hence the points belonging to the wheels patches have been displaced, the gird in proximity of the gears undergoes a distortion, which may affect the quality of the mesh and consequently the convergence and the reliability of the simulation. To overcome such problem, it is a widely used method in the CFD framework to perform some mesh morphing iterations. This procedure consists in solving a diffusion equation for the points position, of following kind:

$$\nabla \cdot (\gamma \nabla \vec{d}) = 0 \tag{15}$$

where $\vec{d} = (\vec{x}(t + \Delta t) - \vec{x}(t))$ is the mesh points displacement vector and γ is a diffusivity term. In mesh motion formulations, the inclusion of a diffusive term in the Laplacian of the displacement field is essential to ensure a smooth and physically consistent propagation of boundary deformations into the interior of the mesh. This term prevents excessive mesh distortion by gradually distributing the imposed displacements, thereby enhancing numerical stability and preserving mesh quality during large or highly non-uniform motions.
In this work, the diffusion coefficient γ is defined as directional and set equal to (0.1,0.1,0). These values were obtained empirically by comparing numerical results with experimental measurements. Diffusion along the z-direction is set to zero, since the mesh does not undergo deformation along this direction. Regarding the boundary conditions for Eq. 15, a prescribed displacement is enforced on the pinion and gear patches according to Eq. 14. On the patches in contact with the gears, a zero-gradient condition is imposed to allow free rotation of the bodies, while a zero-displacement condition is applied on the remaining patches. Furthermore, as discussed in the validation chapter (Sec. 4) a symmetry condition is imposed on the front patch, in accordance with the adopted problem formulation.
Figure 4 illustrates the mesh motion steps occurring at each time advancement.
It is possible to observe in Fig. 4b how even a slight rotation of the boundaries, without the morphing step, would lead to a non valid mesh, while in Fig. 4c are displayed the actual displacement vectors applied to the cell centers. When dealing with moving grids, the cells faces sweep a volume of fluid, behavior that generates a numerical flux through the face giving rise to source terms that compromise the conservation of the transported quantities. To account for the mesh motion, the mass transport equation should be written in the following form:

$$\frac{\partial}{\partial t} \int_{V(t)} \rho\phi \, dV + \int_{S(t)} \rho\phi_f (\vec{u} - \vec{u_b}) \cdot \vec{n} \, dS = 0 \tag{16}$$

where $\vec{u_b}$ corresponds to the boundary velocity and the subscript f stands for a value on the face, that in a co-located grid arrangement, as in the OpenFOAM's framework, is obtained by interpolation. What emerges from Eq. 16 is that in the advection term, the flux given by the cell motion is subtracted. In the case that the boundary velocity is equal to the advection velocity, the advection term cancels out and the Lagrangian formulation is obtained. If the cell motion is not known a priori, the sole calculations using the grid velocities is not enough to guarantee the conservation of mass, therefore, an additional conservation law, called space conservation law (Eq. 17), is added to account for volume changes due to the boundary velocity.

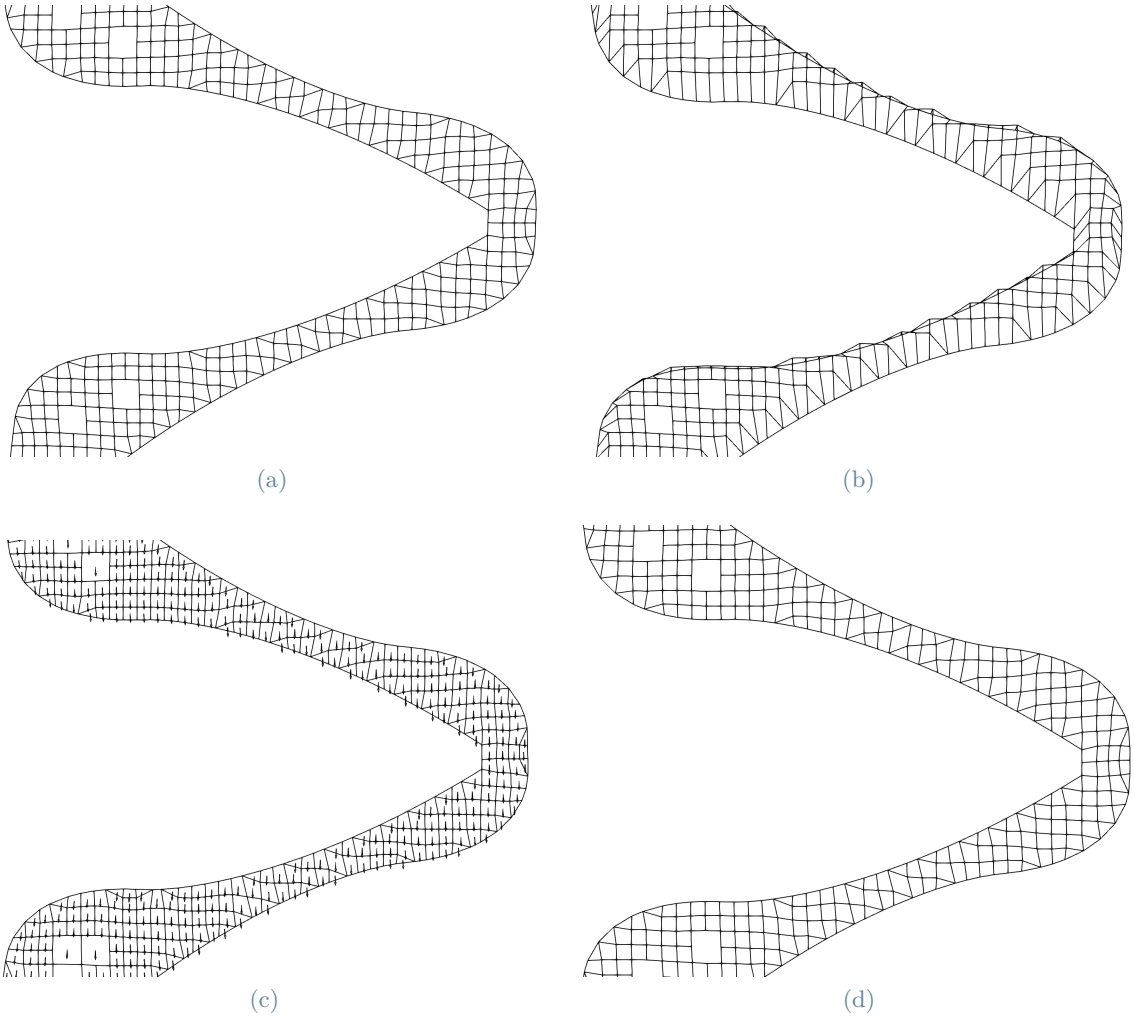


Figure 4: a)Undisplaced grid, b)boundaries displacement, c)Laplacian displacement, d)cell displacement.

$$\frac{\partial}{\partial t}\int_{V(t)} dV - \int_{S(t)} \vec{u_b} \cdot \vec{n}\, dS = 0 \tag{17}$$

Algorithm 2 Mesh motion

1: Mesh motion
   Input: $\vec{x}(t)$, $\theta(t)$, γ
   Output: $\vec{x}(t + \Delta t)$.
2: Boundaries rotation $\leftarrow$ Rodriguess rotation Eq. (14)
3: Laplacian displacement $\leftarrow$ $\nabla \cdot (\gamma \vec{\nabla} \vec{d}) = 0$
4: **Return:** $\vec{x}(t + \Delta t)$

3.4. Topology change

As discussed in the previous sections, during the pre-processing phase a set of meshes must be generated, each corresponding to a distinct instant of the tooth engagement phase. When these instants are reached, a topological change is required in order to allow the simulation to continue on the updated numerical grid. In the context of topological changes a key role is played by the mapping method, whose critical task is to project the fields computed at a certain time instant on a source mesh (srcMesh) to a target one (tgtMesh), preserving the conservation of the conservative fields. There are several mapping algorithms implemented in OpenFOAM,

the first one presented is called intersection, adopted by default in this utility, and is based on the geometrical intersection between the volume of a source cell (identified by the subscript i) and the one of a target cell (identified by the subscript j). For each source cell, a preliminary candidate search is performed on the target mesh using a bounding-box–based spatial acceleration structure. The axis-aligned bounding box of the source cell is first computed and used to query an octree built over the target mesh cell bounding boxes. This query returns only those target cells whose bounding boxes intersect the source cell bounding box, thereby excluding geometrically distant cells from further consideration. Since any actual cell–cell intersection necessarily implies an overlap between their bounding boxes, this coarse screening guarantees completeness while reducing the search complexity from a linear scan over all target cells to a logarithmic-time query (from O(N) to O(logN) where N is the number target cells). The resulting set of candidate target cells is then subjected to an exact geometric intersection test to identify the first valid source–target cell pair, which is used to initialise the subsequent mapping procedure. Once this coarse screening is completed, a finer one is conducted on the flagged cells identified in the previous step. The volume of the source cell (V_i) and the one of the flagged cell (V_j) are computed, and the intersecting volume (V_{ij}) is computed and, if it satisfies relation (18), where ϵ is a tolerance coefficient (equal by default to 10^{-6}) introduced to avoid numerical errors, the cell j is marked as intersecting and a weighting coefficient is computed according to Eq. (19).

$$V_{ij} > \epsilon \cdot V_i \qquad (18) \qquad\qquad w_{ij} = \frac{V_{ij}}{V_i} \qquad (19)$$

Once all the intersections and weights have been computed, the fields can be projected onto the target mesh according to Eq. (20).

$$\phi_j = \frac{1}{V_j} \sum_i \phi_i \, w_{ij} V_{ij} \qquad (20)$$

In order for the method to be fully conservative, the error defined in Eq. (21) should vanish. In practice, however, this condition is not strictly satisfied for two main reasons. The first one is common in every aspect of CFD and derives from the discretized approach to the problem. The second is specific to the intersection-based mapping approach and is related to the inequality in Eq. (18), which leads to the neglect of cell superimpositions whose volume falls below the prescribed tolerance. This approximation introduces an additional source of error that contributes to the continuity error, defined as a numerical measure of the degree to which mass conservation is satisfied during the simulation, thereby affecting the overall solution accuracy.

$$\sum_i \phi_i V_i - \sum_j \phi_j V_j = e \qquad (21)$$

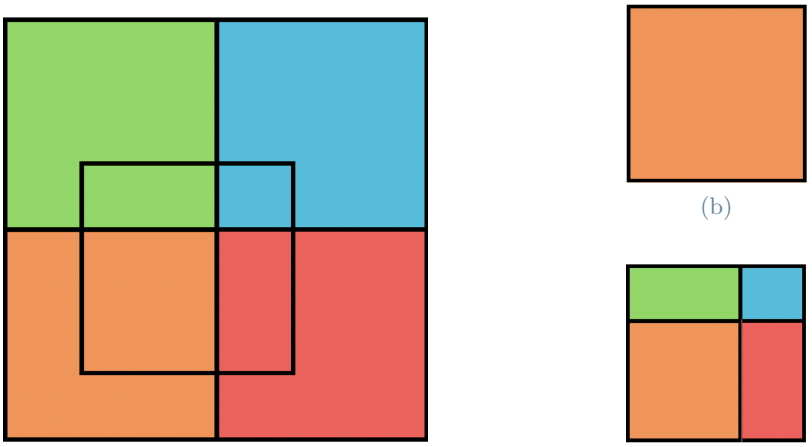
An alternative approach to the method previously described is the use of the nearest-neighbour interpolation. The initial procedure is analogous to that of the intersection-based method, in which the most likely candidate cells are first identified through an octree-based spatial search. Unlike the intersection approach, however, no geometric overlap is evaluated. Instead, the target cell is selected by minimizing the Euclidean distance between the center of the source cell and the centers of the candidate target cells located at the lowest level of the bounding-box tree, as expressed in Eq. (22), where $\mathbf{C}$ denotes the Cartesian coordinates of the cell-center position.

$$d^2_{\min} = \min |\mathbf{C}_i - \mathbf{C}_j|^2 \qquad (22)$$

If a single target cell is associated with multiple source cells, only the closest source cell is selected, while the others are discarded. These discarded cells are flagged as unmapped and may later be assigned to the next-closest target cells. When the target mesh contains fewer cells than the source mesh, some source cells remain unmapped, which can affect conservation properties. Conversely, if the target mesh has more cells than the source mesh, the unmapped target cells are iteratively assigned the values of their nearest source cell.
Once all source-target pairs are established, the fields are mapped on a 1:1 basis, as shown in Eq. (23). A simplified 2D representation of the two interpolation schemes is illustrated in Fig. 5.

$$\phi_i = \phi_j \qquad (23)$$

In order to compare the two mapping methods presented above, two simulations were conducted using pre-existing meshes derived from previous studies. The objective of these tests is to assess and compare the overall performance of the proposed topology-change methodology in terms of execution time and accuracy. The simulations considered here are characterized by numerical settings analogous to those adopted in the validation section (Sec. 4), where they are discussed in greater detail. For the present comparison, it is sufficient



(a) (b) (c)

Figure 5: a)Source cells and target cell; b)nearest; c)intersection.

to note that the simulations are time-dependent and are performed under the incompressible flow assumption, using identical time-step sizes over a time interval corresponding to the engagement of a pair of gear teeth, that is, a complete cycle of the mesh sequence.
One of the most relevant indicators of conservation quality, and therefore of simulation accuracy, is the continuity error, which quantifies the deviation from exact mass conservation arising from the numerical solution of the discretized continuity equation. For the incompressible Navier-Stokes equations, mass conservation is expressed by the following relation:

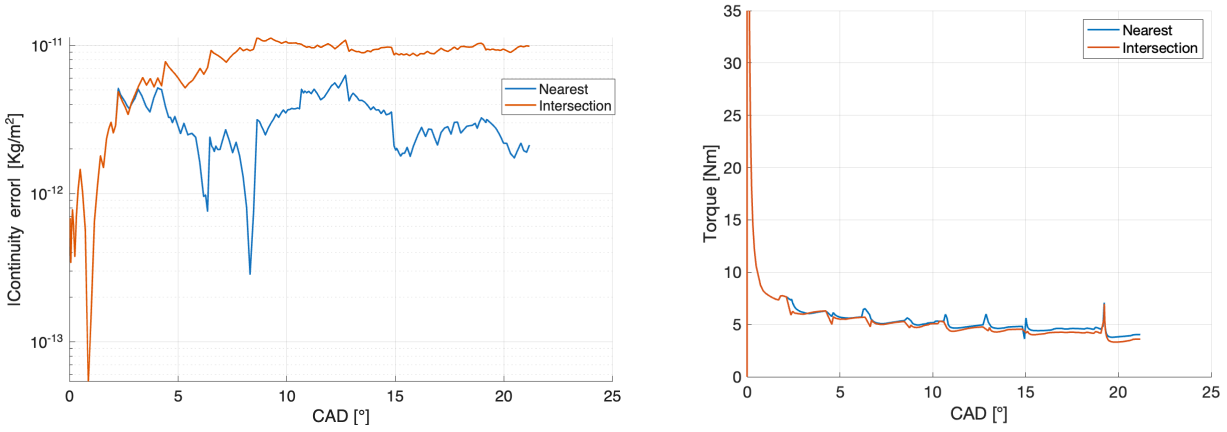
$$\nabla \cdot \vec{U} = 0 \tag{24}$$

the continuity error is defined in Eq. 25, while Eq. 26 is the discretized form normalized with respect to the cell volume, that is computed every time interval after the governing equations are solved.

$$\text{continuityError} = \int_V \nabla \cdot \vec{U}\, dV \tag{25}$$

$$\text{continuityError} = \sum_j \frac{\sum_i \rho_i \vec{U}_i \cdot \vec{n}_i S_i}{V_j} \Delta t \tag{26}$$

Where, in Eq. 26, the sub-index j cycles all the cells while, whereas i cycles all the cell's faces. In the following figure are presented the results.



(a) (b)

Figure 6: a) Cumulative continuity error; b) gears resisting torque.

In Fig. 6a are plotted the discretized cumulative continuity error for each time-step, what emerges is that the nearest interpolation method is characterized by a lower error with respect to intersection, that conversely shows a much more stable behavior when convergence is reached, while the former presents marked peaks and troughs.

The convergence study is performed by evaluating the resisting torque on the gears, as this quantity represents an integral measure of the overall fluid–mechanical behavior within the apparatus and is highly sensitive to mesh refinement and numerical accuracy, it is given by the pressure and viscous forces acting of the teeth sides, computed according to Eq.30, results are laid out in Fig. 6b.

	Nearest[s]	**Intersection**[s]	**Baseline**[s]		**Nearest**[s]	**Intersection**[s]	**Baseline**[s]
0	21.62	356.72	47.07	**5**	18.81	600.89	47.99
1	18.36	562.05	58.11	**6**	19.22	587.95	48.82
2	18.23	570.43	50.77	**7**	20.34	605.94	47.52
3	17.75	581.15	44.05	**8**	18.85	607.27	52.37
4	17.96	584.77	44.47				

Table 3: Remapping execution times.

In Table 3 are listed the topology change execution times required by each method, it emerges clearly that the nearest interpolation method requires much less time than the intersection, that required time interval comparable with the mesh lifespan, ranging from being 16.49 times faster up to 32.75 times.
In light of the results just presented, it is possible to observe that both methods produce similar results, meaning that in this specific case, the adoption of a more accurate method does not offer any relevant advantage. This outcome was likely facilitated by the use of meshes that, although differing in topology, were nonetheless very similar. It should be noted, however, that the choice of mapping method remains problem-dependent, varying with factors such as simulation complexity, required accuracy, and mesh dimension. Moreover, performance metrics such as execution time, although potentially very different, may have only a marginal impact when the interpolation time has an order of magnitude lower with respect to that of mesh life span.
The final aspect that can be drawn from Table 3 concerns the comparison of remapping times with the original configuration introduced at the beginning of this chapter (Sec. 3), which forms the basis of the present work and presented in the *Baseline* column. All timings reported here were obtained from simulations executed on a single core, in order to ensure a direct comparison with the original setup.
Under these conditions, a clear performance difference emerges. The nearest approach consistently outperforms the baseline configuration, with remapping times reduced to between 54% and 68% of the original computational cost. Since the nearest algorithm itself was already employed in the previous setup, this improvement can be primarily attributed to the adoption of the *meshToMesh* utility as it offers significant advantages in terms of memory management. Once a topology change is initiated, the previous mesh is retained in memory and only the new mesh is loaded. By contrast, in the original approach both meshes had to be loaded, since the current mesh was cleared from memory prior to the remapping operation.
In the initial methodology, although the flow solver itself was executed in parallel, the topology-change and remapping stage was inherently serial. Specifically, whenever a topology change was required, the solution fields were reconstructed and mapped on a single processor, after which the domain was decomposed again to allow the simulation to resume in parallel.
An additional advantage provided by the *meshToMesh* utility is the ability to perform the remapping operation fully in parallel. As a consequence, the entire topology-change procedure benefits from domain decomposition. This parallel execution removes the serial bottleneck inherent in the original methodology and enables computationally expensive mapping schemes, such as the intersection method, to achieve execution times that can be lower than those of the baseline approach, provided that a sufficient degree of parallelization is employed.
Since the standard *meshToMesh* class is hard-coded to use the intersection method, the library has been extended to allow the user to arbitrarily select the mapping scheme through the newly introduced `method` entry in the *topoChanger* dictionary. In addition to the *intersections* and *nearest* schemes, the *matching* scheme has also been implemented as an option. Although functionally equivalent to the *nearest* algorithm, this scheme requires a strict one-to-one correspondence between cells, a condition that is not enforced in the present work. Consequently, the *matching* scheme is not applicable in this context and has not been further investigated.
In Alg. 3 is summarized the topology change procedure explained in this paragraph, where `repeat` denotes the rotation period expressed in crank angle degrees (CAD). This parameter is introduced to ensure that the meshes generated represent only the engagement of a single pair of teeth, subsequently, as the geometry is periodically repeated, the correct target topology can be determined by dividing the current time, expressed in CAD, by `repeat`.

Algorithm 3 Topology change

1: Topology change
Input: `srcMesh`, `tgtMesh`, `repeat`, `meshTimes` , Time, ϕ_i
Output: ϕ_j
2: **if** $fmod(Time, \texttt{repeat}) \in \texttt{meshTimeIndices}$ **then**
3: Find `tgtCells` & compute weights $(w_{ij}) \leftarrow$ `meshToMesh`(`srcMesh`, `tgtMesh`)
4: Fields projection: $\phi_j \leftarrow$ `meshToMesh`(`srcCells`, `tgtCells`, w_{ij}, ϕ_i)
5: **Return** ϕ_j
6: **else**
7: Continue
8: **end if**

4. Validation

The complete numerical workflow developed in the present study has been validated against the experimental and numerical results reported by Concli et al. [5]. The reference work aims at the validation of a CFD-based methodology for the prediction of load-independent power losses in geared transmissions. Such losses arise from the interaction between rotating mechanical components and the surrounding lubricant and represent a non-negligible contribution to the overall efficiency of gearboxes, especially under high-speed operating conditions. The total power loss in a gearbox can be decomposed into several contributions associated with different mechanical components, and it can be expressed as:

$$P_L = P_{LG} + P_{LG0} + P_{LB} + P_{LB0} + P_{LS} + P_{LS0}, \tag{27}$$

where the subscripts G, B, and S denote respectively the power losses related to gears, bearings, and seals, respectively. The subscript 0 identifies the load-independent contributions, which are primarily governed by fluid-dynamic phenomena such as windage, churning, and squeezing effects.
The focus of the experimental campaign presented in [5], and consequently of the present validation activity, is the assessment of the load-independent gear losses P_{LG0}. These losses originate from the interaction between the rotating gears and the lubricant and are independent of the transmitted torque. Accurately capturing this contribution represents a fundamental prerequisite for the validation of CFD-based approaches aimed at predicting gearbox efficiency and thermal behavior.

4.1. Experimental setup

The experimental data used for validation were obtained from the test campaign described in [5]. The experimental setup consists of a pair of spur gears, whose main geometrical parameters are summarized in Table 4. The driving gear is mechanically coupled to an electric motor, while the driven gear is free to rotate about its own shaft without transmitting load.
Both gears are housed inside a rigid casing with overall dimensions of $290 \times 145 \times 185$ mm, completely filled with FVA2 lubricant. The lubricant is pressurized at 6 bar in order to ensure fully flooded operating conditions and to avoid the presence of air pockets and the rising of phenomena such as cavitation. The physical properties of the lubricant at the reference temperature are density of $\rho = 871$ kg/m^3 and a kinematic viscosity of $\nu = 29.8$ mm^2/s.
The driving shaft is supported by rolling bearings and connected to the electric motor, whereas the driven shaft is mounted on rolling bearings and rotates freely. The rotational speed and the resistant torque acting on the system are measured by dedicated sensors, allowing the evaluation of the total resistant torque as a function of the angular velocity.
Since the numerical model described in Section 4.2 is specifically designed to capture the fluid-dynamic interaction between the gears and the lubricant, it does not explicitly account for the power losses associated with bearings and seals. For this reason, additional experimental tests were carried out without the gears installed. These measurements allow the quantification of the losses related to bearings, shafts, and seals, which can be subsequently subtracted from the total measured losses in order to isolate the pure gear-related load-independent contribution.

4.2. Numerical setup

As discussed in the previous chapters, the numerical simulations were carried out using the open-source CFD software OpenFOAM. The Navier–Stokes equations were solved, without any turbulence model closure, by means of a transient pressure-velocity coupled solver based on the PIMPLE algorithm, employing two correctors and three outer correctors.
Due to the small clearances between the gears and the relatively low characteristic velocities involved, the flow is characterized by low Reynolds number Eq.28. Under these conditions, the flow can be reasonably assumed to be laminar; therefore, no turbulence model was adopted and the governing equations were solved in their laminar form.

$$Re = \frac{U_{max} D}{\nu} \Rightarrow O(Re) \approx 1.9 \cdot 10^4 \tag{28}$$

Where D is the gear diameter and $U_{max} = \omega_{max} \cdot D/2$ with $\omega_{max} = 4000$ rpm. The lubricant was modeled as a Newtonian fluid. The simulations are initialized with a potential flow solution to improve numerical robustness and convergence. The potential flow provides a physically consistent initial velocity and pressure field that already satisfies the boundary conditions and captures the main features of the outer flow. Starting from such a field reduces strong initial transients, limits non-physical pressure and velocity gradients. As a result, the solver typically converges faster and more reliably than when initialized with a uniform or quiescent flow field.
Although the experimental setup is not geometrically symmetric because of the presence of the shafts, the numerical domain was treated as symmetric by imposing a symmetry boundary condition. This assumption allows a significant reduction in the computational domain size and, consequently, in the total number of mesh elements, without substantially affecting the accuracy of the results.
The fluid was assumed to be incompressible and isothermal. The lubricant temperature was considered uniform, constant, and known *a priori*; for this reason, the energy equation was not solved. The physical properties of the working fluid were therefore assumed constant and were derived from the experimental conditions described in Section 4.1.
The mechanical power losses associated with the gears were computed according to the following expression:

$$P_L = |M_{p_z}||\omega_p| + |M_{g_z}||\omega_g|, \tag{29}$$

where the subscripts p and g refer to the pinion and the gear, respectively, ω_i denotes the imposed angular velocity of each wheel, and M_{i_z} is the component along the z-direction of the resultant torque acting on the corresponding gear.
The torque was evaluated as the sum of the contributions due to pressure and viscous forces acting on the gear surfaces. For each gear, the resultant moment was computed using a dedicated OpenFOAM function object according to:

$$\vec{M} = \sum_i \left(\vec{r}_i \times (P_i \vec{n}_i A_i) + \vec{r}_i \times (\vec{\tau}_i A_i)\right), \tag{30}$$

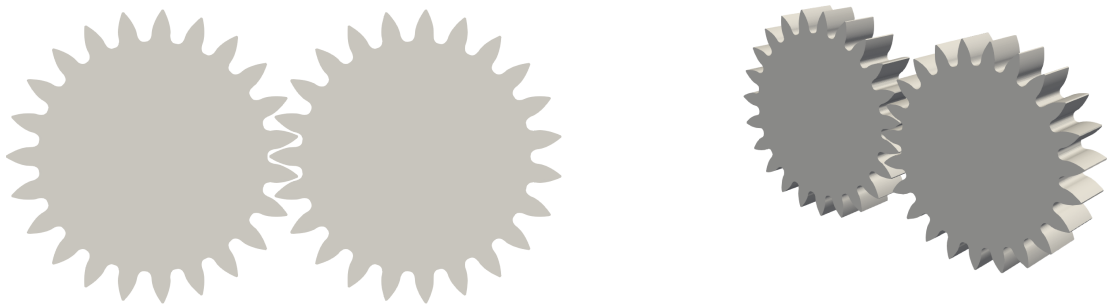
where the index i refers to the i-th face of the gear surface, $\vec{r}_i$ is the position vector of the face center with respect to the gear center of rotation, P_i is the local pressure, $\vec{\tau}_i$ represents the viscous shear stress vector, $\vec{n}_i$ is the outward unit normal vector, and A_i is the face area. Since the simulations were conducted under the assumption of incompressible flow, the reference pressure was set to zero, as only pressure differences are physically relevant and the absolute pressure level does not influence the flow phenomenology. Therefore, in the Results section (Sec. 4.3), the pressure field is presented in terms of relative pressure.
Under the Newtonian fluid assumption, the viscous stress was computed as:

$$\vec{\tau} = \mu \nabla \vec{u}, \tag{31}$$

	Symbol	Unit	Value		Symbol	Unit	Value
Width	b	mm	40	**Tip diameter**	d_a	mm	102.5
Helix angle	β	°	0	**Pressure angle**	α	°	20
Normal module	m_n	mm	4	**Center distance**	a	mm	91.5
Number of teeth	z	–	23	**Chordal thickness**	s_j	mm	5.35

Table 4: Gear parameters.

where μ is the dynamic viscosity of the lubricant.
Concerning the spatial discretization, the gear geometry and the mesh generation were obtained as described respectively in Sections 3.1 and 3.2, using the geometrical parameters listed in Table 4. Owing to the symmetry assumption, the gear thickness was halved in the numerical model. The resulting two-dimensional and three-dimensional geometries are shown in Figure 7.



(a) (b)

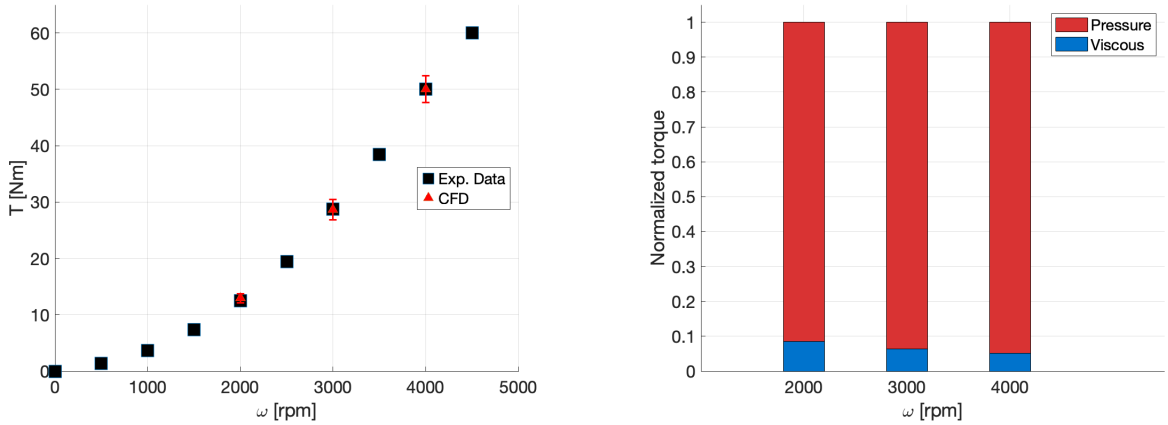
Figure 7: a) 2D geometry; b) 3D geometry.

Algorithm 4 Name of the Algorithm

1: Initial instructions
2: Mesh generation block (Alg. 1)
3: **for** n: `startTime` $\rightarrow$ `endTime` **do**
4: Topology change block (Alg. 3)
5: Mesh motion block (Alg. 2)
6: PIMPLE algorithm block
7: **Return:** U^{n+1}, p^{n+1}
8: **end for**

4.3. Results

In this final part of the validation section, the results are presented and discussed. Three simulations were performed, each at a constant angular speed (2000 rpm, 3000 rpm, and 4000 rpm). Convergence was monitored through the resistant torque and was considered achieved after two engagements; consequently, a total of three engagements were simulated for each case, corresponding to a rotation of 63 CAD for each gear. Fig. 8a shows the resistant torque as a function of angular velocity for both the experimental campaign and the CFD results. The numerical torque was computed, according to Eq. 30, by accounting for the contributions of both gears and subsequently multiplied by two to enforce the geometrical symmetry. In addition, the individual inertial and pressure relative contributions are also reported in Fig. 8b. The numerical values displayed in the figure were obtained by time-averaging the torque over the last simulated engagement, while the associated error bands were determined from the discretized standard deviation.
In Fig. 9 are displayed the relative pressure fields on the pinion side, far from the engagement zone, for the different rotational speed. The pressure distribution on the pinion surface exhibits higher values on the leading flanks and lower values on the trailing flanks. Moreover, the maximum and minimum pressure values are observed near the axial ends of the leading and trailing flanks, respectively, showing increasingly pronounced intensity extremes as velocity increases. This pressure gradient gives rise to a secondary flow directed from the high-pressure regions toward the low-pressure regions, as also evidenced by the streamlines reported in Fig. 10.



(a) (b)

Figure 8: a)Experimental vs. numerical data b)Torque relative contributions.

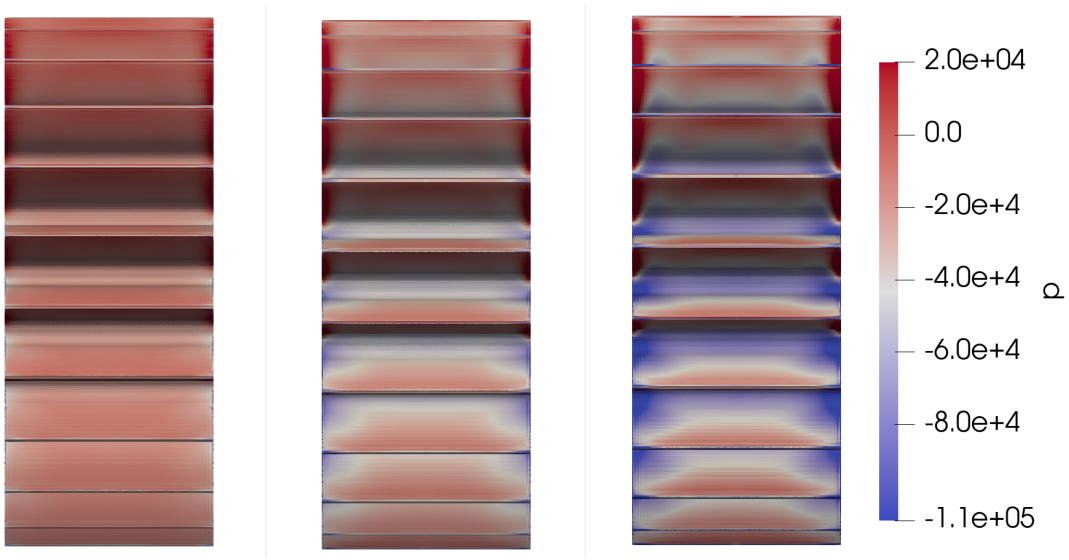


Figure 9: Relative pressure field: L) 2000rpm, C)3000rpm, R)4000rpm

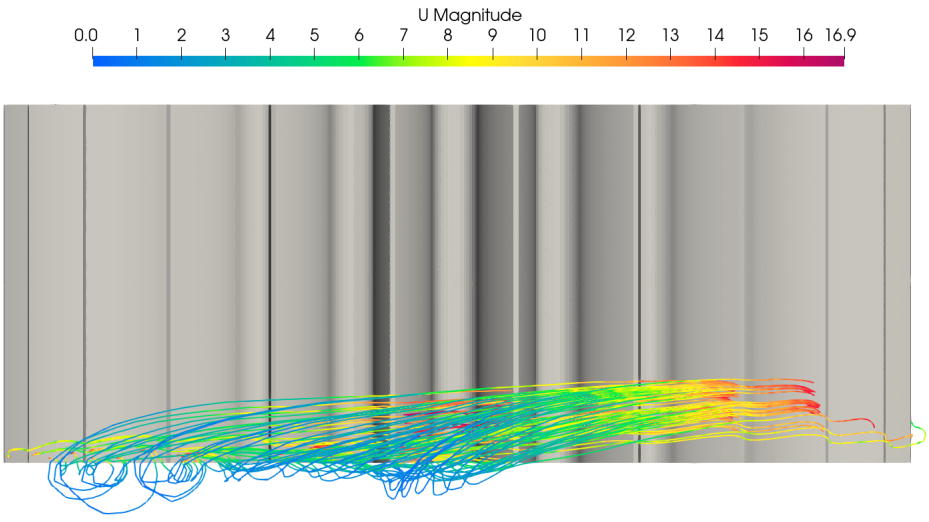


Figure 10: Streamlines, 3000 rpm

In the remainder of this section, some characteristic flow phenomena are presented. Since these features are

qualitatively similar for all the considered velocities, given that the flow is incompressible and no submodels (e.g., cavitation or aeration) are employed, they are discussed with reference to the angular velocity of 3000 rpm. In Fig.11, the velocity magnitude field on the symmetry plane is shown, a backflow can be clearly observed in the region where gears engagement begins.

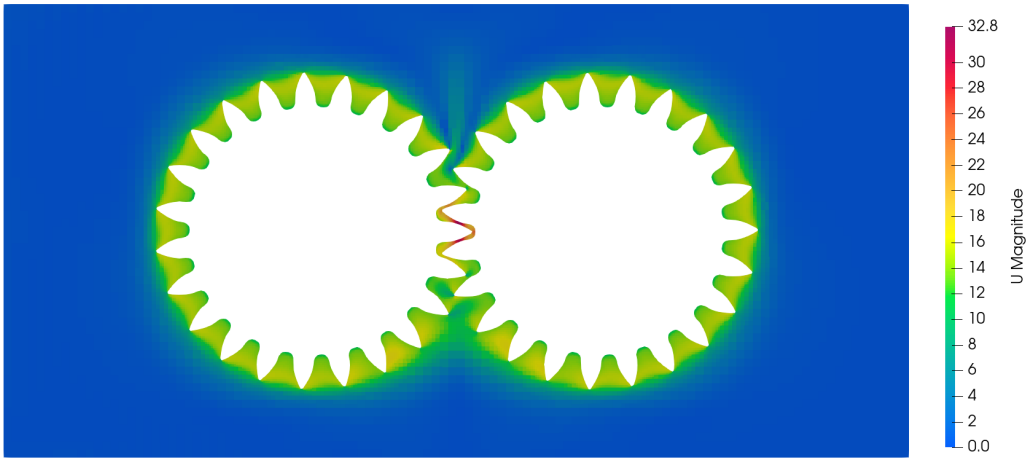
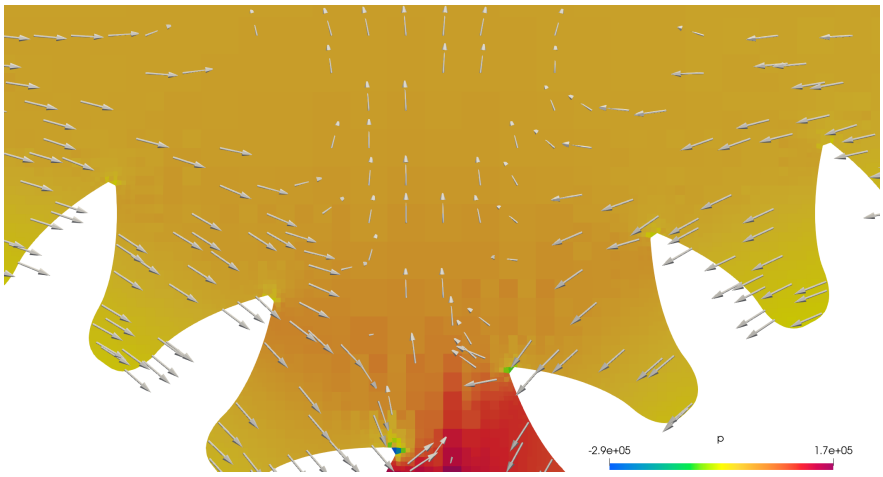


Figure 11: Velocity magnitude field, 3000rpm

This phenomenon is characteristic of mating gears flow and originates from the formation of a high-pressure front at the onset of the mating region, which causes part of the flow to be diverted in a direction opposite to the gear rotation. This behavior is more clearly illustrated in Fig. 12, where the pressure field on the symmetry plane is displayed together with the velocity vector field.



(a)

Figure 12: Relative pressure field, 3000rpm

Another secondary flow is generated by the squeezing (or pocketing) effect, which is associated with the pressure increase occurring as the gap between the teeth reduces at the onset of tooth pairing, followed by a pressure decrease due to the subsequent volume expansion downstream of the contact point. In the present work, the gears are slightly scaled down and the actual contact point is therefore not reproduced. As a result, the pressure values obtained in this region are not quantitatively reliable; nevertheless, the qualitative behavior of the phenomenon can be observed in Fig. 13, where it manifests as a flow exiting the gear cavities in the axial direction and recirculating externally along the lateral sides of the gears in the disengagement region.

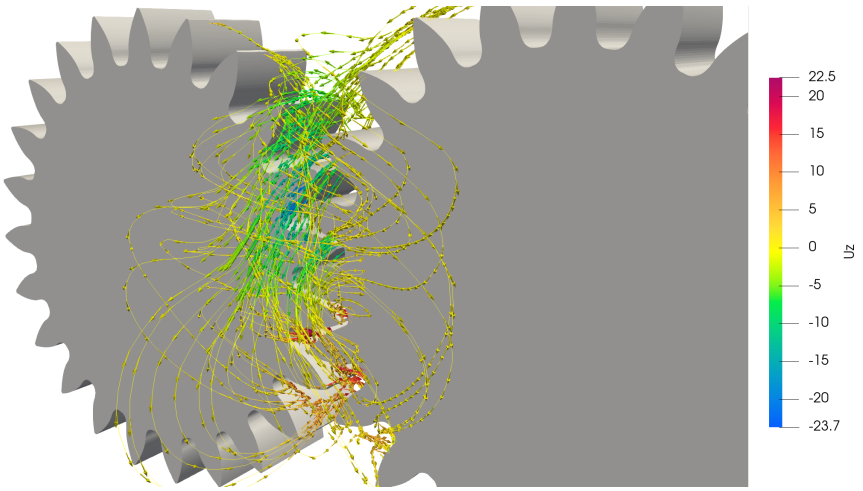
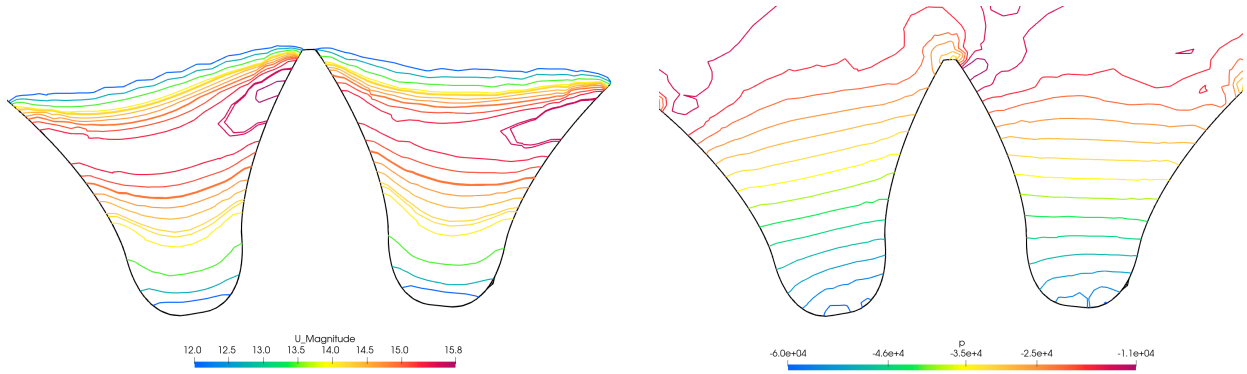


Figure 13: Pocketing, 3000rpm

In Fig. 14, the contour plots of the velocity magnitude and relative pressure variation are reported for two tooth cavities on the symmetry plane of the gears. Figure 14b reveals a pressure gradient predominantly oriented along the radial direction. This distribution is mainly governed by centrifugal effects, as the fluid is confined within the cavity formed between a pair of meshing teeth and rotates quasi-steadily with the gears.
Under these conditions, convective accelerations are negligible and the centrifugal force can be absorbed into the pressure term, leading to a quasi-hydrostatic pressure distribution in the rotating reference frame.
This pressure pattern is locally disrupted near the tooth tip radius, where the oil is no longer fully confined, and in the vicinity of the trailing flank close to the tooth tip, where flow separation and recirculation occur. These features are also evident in the velocity magnitude contours shown in Fig. 14a.



(a) (b)

Figure 14: a)Velocity magnitude contours b)relative pressure variation contours: same trend as the centrifugal force

5. Multi-phase

To further extend the applicability and the relevance of the dynamic mesh handling methodology presented in this work, a multiphase test case has been developed. Two fluids are employed, oil and air each filling half of the casing, to better represent the real operating conditions, such as in external gear pumps and gearboxes.
In this case is employed a Volume of Fluid (VoF) incompressible solver. Once again the simulations are considered isothermal, so the energy equation is not solved, this time Reynolds averaged Navier-Stokes (RANS) equations were solved, adopting the $k-\epsilon$ turbulence model as closure of the system. From the standard incompressible, single phase equations, an additional scalar field, the void fraction (α) is introduced, a dimensionless ratio representing the empty space (voids) within a two-phase flow, with alpha bounded between 1, pure oil, and 0, pure air (Eq. 32). The solver is single fluid type, therefore it solves the equations for a single fluid which has the physical properties weighted, between those of the phases, on the void fraction (Eq. 33). An additional transport equation is solved, in order to have the value of α within each cell (Eq. 34). The incompressilbeVoF is an interface-capturing method, therefore the free surface is not represented as a sharp boundary but rather

as a diffused interface.

$$\alpha_{\text{oil}} = \frac{V_{\text{oil}}}{V} \qquad \alpha_{\text{oil}}(\vec{x}, t) = \begin{cases} 1 & \text{oil} \\ (0,1) & \text{interface} \\ 0 & \text{air} \end{cases} \tag{32}$$

$$\Theta = \Theta_{oil}\,\alpha_{oil} + \Theta_{air}\,(1 - \alpha_{oil}) \tag{33}$$

$$\frac{\partial(\rho\,\alpha)}{\partial t} + \nabla \cdot (\rho \vec{u} \alpha) = S \tag{34}$$

Where Θ, represent a generic physical property and S is a source term to account for sub-models.
The aim of these simulations is to asses the aeration phenomenon and the impact on the performance of mating gears. Aeration develops in turbulent flows, specifically when the turbulence is high enough in the free-surface flow, air bubbles are entrained into the liquid phase and transported along with it. The length scale involved in this phenomenon are in the order of millimeters, therefore without a very fine grid, air entrainment is not captured. Another possible way is to model aeration through subgird models.
The simulations were conducted implementing the Hirt's aeration model [10]. Such model introduces an explicit source term S defined as:

$$S = C_{air}\,A_s \sqrt{2\frac{P_t - P_d}{\rho}} \tag{35}$$

with C_{air} being a calibration constant, A_s the free surface area at each cell, ρ the fluid density computed according to Eq. 33, P_t and P_d respectively a destabilizing contribution, given by turbulence, and a stabilizing one, given by the action of gravity and surface tension, defined as:

$$P_t = \rho\,k \qquad P_d = \rho g_n L_t + \frac{\sigma}{L_t} \tag{36}$$

where, k is the turbulent kinetic energy, g_n is the normal component to the free surface of the gravitational acceleration, σ the surface tension and L_t the turbulent characteristic length scale defined as:

$$L_t = C_\mu \sqrt{\frac{3}{2}} \frac{k^{\frac{3}{2}}}{\epsilon} \tag{37}$$

where, $C_\mu = 0.09$ for the $k-\epsilon$ turbulence model, and ϵ the turbulence diffusivity. The aeration model is computed just in proximity of the interface; an air source term appear in the lubricant phase when the condition $P_t > P_d$ is verified.
For these simulations the meshes were provided externally, however the dynamic mesh handling a topology change methodologies are kept the same as explained in the previous chapters, demonstrating the flexibility and robustness of the methodology proposed.
The results have been paired against experimental tests. Two lubricants have been tested, their properties are listed in the Tab. 5. Experimental tests were conducted at 26.5°C, therefore the correct properties are computed as follows: for the density a linear volume expansion is assumed, giving Eq. 38.

$$\rho(T) = \frac{\rho(T_0)}{1 + \beta\,(T - T_0)} \tag{38}$$

while the kinematic viscosity is computed according to the ASTM D 3341 standard (Eq. 39).

$$log(log(\nu + 0.7)) = A - B\,log(T) \tag{39}$$

Where β, A and B are constants computed through the data given in the table. Air is considered at standard conditions.
The two lubricants exhibit the same density and kinematic viscosity; however, different aeration behavior is expected.
The experimental tests were conducted following a similar procedure described in Section 4.1. In this case, a sample of oil was extracted from the bottom of the casing in order to determine the aeration level of the lubricant. The numerical simulations were performed assuming a constant angular velocity of the pinion equal to 1000 rpm, with a gear ratio of 25:30. In both cases, a value of $C_{air} = 0.5$ was adopted, while different surface tension values were prescribed according to the experimental data. As in the previous analyses, only half of the computational domain was discretized, and a symmetry boundary condition was applied. A total of eight

	Unit	VP.FLG.4.2025/207	VP.FLG.4.2025/208
Kinematic viscosity at 40 °C	mm^2/s	105	105
Kinematic viscosity at 100 °C	mm^2/s	12	12
Surface Tension	N/m	0.032	0.028
Viscosity index	–	103	103
Density at 15 °C	kg/m^3	870	870
Density at 50 °C	kg/m^3	848	848
Air release at 50 °C	min	5	13
Foaming characteristics – Seq. I	ml	10 / 0	650 / 300
Foaming characteristics – Seq. II	ml	10 / 0	40 / 0
Foaming characteristics – Seq. III	ml	15 / 0	670 / 350

Table 5: Lubricant properties

Oil	Aer_{exp}	$T_{exp}[Nm]$	$T_{CFD}[Nm]$	Δ_T
207	12.5%	0.12	0.1221	-1.8%
208	25%	0.125	0.1223	2.2%

Table 6: Aeration results

complete pinion revolutions were simulated. Since the computations were carried out using a RANS approach, only the final time step was considered for post-processing and analysis of the results.

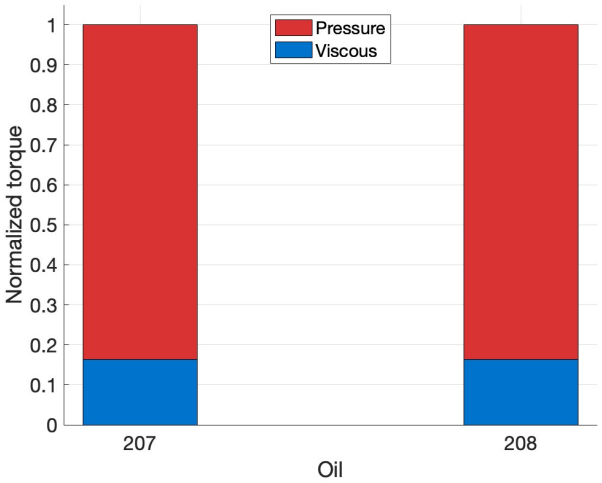
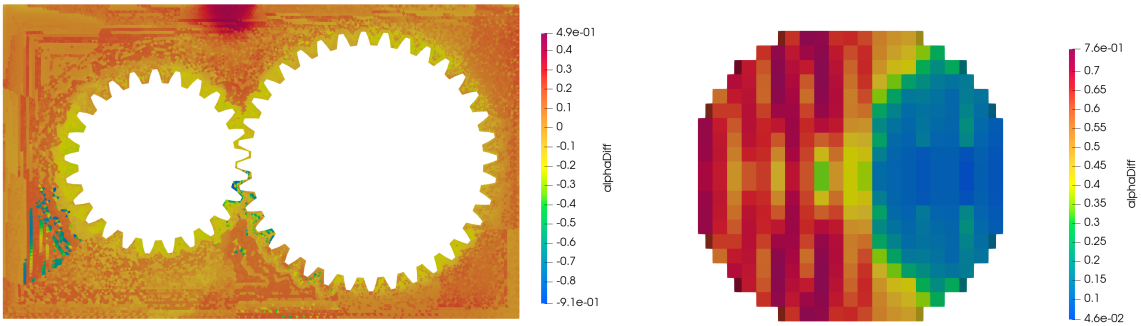


Figure 15: Torque relative contributions

Table 6 reports the results related to the resistant torque obtained from both the experimental and numerical tests, whereas Fig. 15 shows the relative torque contributions extracted from the CFD simulations. Figure 16a illustrates, on the symmetry plane, the difference in the oil void fraction (Eq. 40) with respect to the 208 lubricant between the two simulations, highlighting the different air distributions resulting from the change in the working fluid.
The relative aeration difference in the numerical results was quantified by averaging the oil volume fraction according to Eq. 41 over a circular region corresponding to the oil extraction port located on the bottom of the casing, as shown in Fig. 16b.



(a) (b)

Figure 16: a)VoF difference on the symmetry plane b) VoF difference oil port.

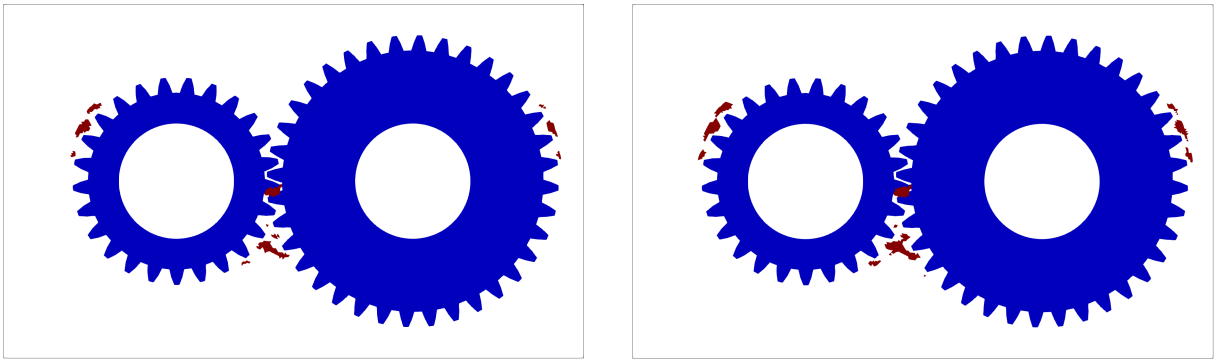
$$\Delta_{\alpha_{oil}} = \frac{\alpha_{oil}^{208} - \alpha_{oil}^{207}}{\alpha_{oil}^{208}} \tag{40}$$

$$\bar{\Delta}_{\alpha_{oil}} = \frac{1}{V} \int \Delta_{\alpha_{oil}} \, dV = 0.438 \tag{41}$$

Given the experimental data on aeration the relative relative variation can be computed as:

$$V = \frac{Aer_{exp}}{V_{oil}(1 + Aer_{exp})} \qquad \Delta = \frac{V_{208} - V_{207}}{V_{208}} = 0.464 \tag{42}$$

Figure 17 shows the cells in which an air source term is introduced, highlighted in red, according to Hirt's model during the simulations for the two lubricants. As expected, oil 208 exhibits a larger number of air source cells, indicating a higher level of aeration.



(a) (b)

Figure 17: a) Air sources Oil 207 b) Air sources 208; 100CAD

The results show good agreement between the experimental measurements and the CFD predictions. Furthermore, in agreement with the literature [15], they indicate that increasing lubricant aeration leads to a corresponding increase in power losses.

6. Conclusions

A computational workflow for the simulation of the fluid-dynamic behavior of mating gears has been developed and assessed. Following a review of the relevant literature, a dynamic-mesh strategy based on a combination of mesh morphing and mesh replacement was selected. This approach, while effective, entails a complex pre-processing phase due to the requirement for multiple meshes representing successive gear-engagement positions. To address this challenge, a dedicated utility was adopted to reconstruct the gear geometry starting from its geometrical parameters. The geometry is initially generated in the form of a point cloud, subsequently converted into a three-dimensional solid model, and automatically meshed using the *snappyHexMesh* utility. The resulting workflow enables the automated generation of a sequence of meshes describing the gear engagement through user-defined rotational increments, while consistently enforcing the correct relative rotation between the gears according to the gear ratio.

The dynamic-mesh handling was then addressed through the implementation of a boundary condition enforcing the kinematic law governing the gear motion, with particular attention devoted to the management of topology changes. In this context, the adoption of the *meshToMesh* utility proved highly effective, allowing the automatic mapping of solution fields based on the simulation time, the cyclic reuse of meshes to achieve continuous gear rotation as a succession of engagements, and the parallelisation of the mapping procedure. The associated library was further extended to include multiple interpolation schemes, increasing the flexibility and applicability of the approach.

The complete workflow was validated against experimental measurements, with particular emphasis on the comparison of the resistant torque acting on the gears. The numerical results showed close agreement with the experimental data, demonstrating the accuracy and robustness of the proposed methodology. Moreover, the characteristic physical phenomena associated with the operation of geared systems were correctly reproduced, within the limitations inherent to the modeling assumptions adopted.

The methodology proved to be effective even in the multi-phase test case, demonstrating flexibility, allowing to handle third party meshes, and to endure more complex cases, being able to represent successfully the phenomenology of two phases and their interaction through an aeration sub-model.

Further developments may focus on improving the robustness and generality of the workflow. In particular, the extension to fully support helical gears represents a relevant improvement. At present, such geometries can only be partially handled, as the automated generation of a stereolithographic representation of helical gears is not available, although externally provided geometries can already be successfully meshed. Additional enhancements may include reducing the overall computational cost associated with mesh generation, for instance by exploiting structured or semi-structured approaches based on the *blockMesh* utility. Finally, a more accurate representation of the zero-clearance contact region between mating gears could be achieved, either through dedicated local mesh refinement strategies or by introducing appropriate artificial-viscosity models to replicate the contact region behavior.

References

[1] P. Casoli, A. Vacca, and G. L. Berta. A numerical model for the simulation of flow in hydraulic external gear machines. *Power Transmission and Motion Control (PTMC)*, 80:147–168, 2006.

[2] Ismail B. Celik, Urmila Ghia, Patrick J. Roache, Christopher J. Freitas, Hugh Coleman, and Peter E. Raad. Procedure for estimation and reporting of uncertainty due to discretization in cfd applications. *Journal of Fluids Engineering*, 130(7):078001, 07 2008.

[3] F. Concli and C. Gorla. Influence of lubricant temperature, lubricant level and rotational speed on the churning power loss in an industrial planetary speed reducer: Computational and experimental study. *Int. J. Comput. Methods Exp. Meas.*, 1:353–366, 2013.

[4] F. Concli and C. Gorla. Windage, churning and pocketing power losses of gears: different modeling approaches for different goals. *Forsch Ingenieurwes*, 80:85–99, 2016.

[5] F. Concli, A. Della Torre, C. Gorla, and G. Montenegro. A new integrated approach for the prediction of the load independent power losses of gears: Development of a mesh-handling algorithm to reduce the cfd simulation time. *Advances in Tribology*, page 8, 2016.

[6] R. S. Devendran and A. Vacca. Theoretical analysis for variable delivery flow external gear machines based on asymmetric gears. *Mechanism and Machine Theory*, 108:123–141, 2017.

[7] J. Donea, A. Huerta, J. Ponthot, and A. Rodriguez-Ferran. Arbitrary lagrangian–eulerian methods. *Encyclopedia of Computational Mechanics*, 1:413–437, 11 2004.

[8] F.Concli and M. N. Mastrone. Latest advancements in the lubricant simulations of geared systems: a technology ready for industrial applications. *Forschung im Ingenieurwesen*, 2023.

[9] Christopher Greenshields and Henry Weller. *Notes on Computational Fluid Dynamics: General Principles*. CFD Direct Ltd, Reading, UK, 2022.

[10] C. W. Hirt. Modeling turbulent entrainment of air at a free surface. *Flow science, Inc.*, 2003.

[11] S. Idelsohn and E. Oñate. To mesh or not to mesh. that is the question. . . *Comput. Methods Appl. Mech. Engrg.*, 195:4681, 2006.

[12] IEEE Global Fluid Power Society Symposium. *CFD Simulation of Helical External Gear Pump with Asymmetric Tooth Profile: Development of 3D Helical Mesh Generation and Motion Algorithm and Experimental Validation*, 10 2023.

[13] L. Maccioni and F. Concli. Computational fluid dynamics applied to lubricated mechanical components: Review of the approaches to simulate gears, bearings, and pumps. *Applied Sciences*, 10(24):8810, 12 2020.

[14] S. Mancò and N. Nervegna. Simulation of an external gear pump and experimental verification. In The Japan Fluid Power System Society, editor, *Proceedings of the JFPS International Symposium on Fluid Power*, volume 1, pages 147–160, 1989.

[15] Mastrone M. N. and Concli F. Numerical modeling of fluid's aeration: Analysis of the power losses and lubricant distribution in gearboxes. *Journal of Applied and Computational Mechanics*, 9, 2022.

[16] Mastrone M. N. and Concli F. Numerical modeling of fluid's aeration: Analysis of the power losses and lubricant distribution in gearboxes. *Journal of Applied and Computational Mechanics*, 9(1), 2023.

[17] F. Orlandi, L. Montorsi, and M. Milani. Cavitation analysis through cfd in industrial pumps: A review. *International Journal of Thermofluids*, 20:100506, 2023.

[18] F. Orlandi, G. Muzzioli, M. Milani, F. Paltrinieri, and L. Montorsi. Development of a numerical approach for the cfd simulation of a gear pump under actual operating conditions. *Fluids*, 8(7):244, 2023.

[19] P. J. Roache. Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, 116(3):405–413, 09 1994.

[20] Javier Martìnez Rubio. *Multidimensional simulations of external gear pumps*. PhD thesis, Politecnico di Milano, Milano, 12 2017.

[21] M. Rundo. Models for flow rate simulation in gear pumps: A review. *Energies*, 10(9):1261, 8 2017.

[22] Y. G. Shah, A. Vacca, S. Dabiri, and E. Frosina. A fast lumped parameter approach for the prediction of both aeration and cavitation in gerotor pumps. *Meccanica*, 53, 1018.

[23] G. Shoukat, K. Siddique, and M. Sajid. Modeling and cfd analyses of gear pump in openfoam. *ASME-JSME-KSME 2019 8th Joint Fluids Engineering Conference*, 2, 2019.

Abstract in lingua italiana

I sistemi a ingranaggi che interagiscono con un fluido di lavoro sono componenti essenziali in numerosi settori industriali, dall'aerospaziale al petrolchimico, con le pompe a ingranaggi esterni che ne rappresentano un esempio emblematico. Le tradizionali simulazioni numeriche hanno storicamente fatto affidamento su formulazioni analitiche o modelli a parametri concentrati per la loro efficienza computazionale; tuttavia, tali approcci spesso non riescono a catturare l'accoppiamento tridimensionale di complessi meccanismi fisici. Al contrario, la Fluidodinamica Computazionale (CFD) fornisce un contesto robusto per l'analisi di fenomeni localizzati quali le perdite interne, la cavitazione e l'aerazione, che diventano sempre più rilevanti man mano che i sistemi evolvono verso una maggiore compattezza e velocità di rotazione più elevate.
Questo lavoro presenta una procedura CFD completamente automatizzata, sviluppata specificamente per affrontare le sfide poste dalla deformazione continua della geometria interna. Per gestire il dominio dinamico, viene adottata una strategia ibrida che combina il movimento della mesh e la sostituzione della mesh, privilegiando l'accuratezza della simulazione pur mantenendo un costo computazionale equilibrato, e superando le limitazioni di approcci più semplici come il *Multiple Reference Frame* e le tecniche di *overlapping grids* non conservative.
La procedura sviluppata, automatizza la generazione della geometria degli ingranaggi a partire dai parametri di progetto e impiega l'applicazione *snappyHexMesh* per produrre griglie conformi alle superfici. Le variazioni topologiche vengono gestite tramite *meshToMesh*, che consente di mappare i campi in parallelo ed il riutilizzo ciclico di sequenze di griglie, riducendo significativamente gli sforzi di *pre-processing*.
La metodologia è validata mediante il confronto tra la coppia resistente calcolata numericamente e le misure sperimentali, mostrando un eccellente accordo e riproducendo accuratamente i principali fenomeni fisici, inclusi il riflusso e l'andamento della pressione con la forza centrifuga. Infine, per rappresentare in modo più realistico le condizioni operative e dimostrare la flessibilità del metodo proposto, la procedura è stata estesa a simulazioni multifase mediante un approccio *Volume of Fluid* che implementa il modello di aerazione di Hirt.

Parole chiave: Pompe a ingranaggi esterni, Ruote dentate, Fluidodinamica computazionale, Generazione griglia, Movimento griglia