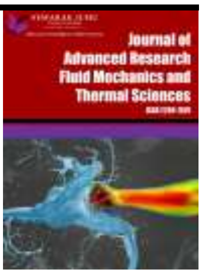
Journal of Advanced Research in Fluid
Mechanics and Thermal Sciences

Journal homepage:
https://semarakilmu.com.my/journals/index.php/fluid_mechanics_thermal_sciences/index
ISSN: 2289-7879



A Comprehensive Survey of Open-Source Tools for Computational Fluid Dynamics Analyses

Jyotsna Balakrishna Kodman[1], Balbir Singh[1,*], Manikandan Murugaiah[1,♣]

[1] Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India

ARTICLE INFO

Article history:
Received 17 March 2024
Received in revised form 29 June 2024
Accepted 11 July 2024
Available online 30 July 2024

Keywords:
CFD; open-source software; geometry modelling; mesh generation; numerical methods; post-processing

ABSTRACT

Computational Fluid Dynamics (CFD), blending disciplines like fluid mechanics and computer science, plays a pivotal role in various engineering and scientific endeavors. Despite its importance, the prohibitive costs and restricted access to commercial CFD tools pose significant barriers. This study addresses the need for accessible CFD solutions by conducting a comprehensive review of open-source CFD tools, highlighting their role in promoting open science. Through methodical analysis, the present study explores the capabilities, performance, and applicability of these tools in various contexts. The findings reveal that open-source CFD tools not only offer a cost-effective alternative to proprietary software but also foster collaboration and transparency in the scientific community. This study concludes that these tools are not only viable but essential for the advancement of CFD applications, encouraging wider adoption and development. This review serves as a bridge in the literature, enhancing understanding and accessibility of open-source tools in CFD, and supporting the paradigm shift towards open science.

1. Introduction

Computational Fluid Dynamics (CFD), an interdisciplinary field combining fluid mechanics, heat transfer, computational methods, and computer science, has revolutionized fluid flow and heat exchange analysis [1-4]. Its applications range from aerospace engineering to environmental modeling, establishing it as a cornerstone of modern scientific and engineering practices [5-11]. However, the high costs and proprietary nature of commercial CFD tools have limited their accessibility. This situation highlights the growing importance of open-source tools in CFD, which offer cost-effective solutions and embrace open science principles, fostering collaboration and transparency in research.

Despite their growing popularity, there is a significant gap in the literature regarding a comprehensive understanding of these tools. Many in the scientific community are not fully aware

---

[*] *Corresponding author.*
E-mail address: balbir.s@manipal.edu
[♣] *Corresponding author.*
E-mail address: manikandan.m@manipal.edu

https://doi.org/10.37934/arfmts.119.2.123148

of the capabilities of open-source CFD tools. This study aims to fill this gap by providing a thorough review of these tools, examining their features, applications, and impact. This paper explores various open-source CFD tools and discusses their practical applications in areas like geometry modeling, mesh generation, simulation, and post-processing. The main goal is to guide the CFD community in leveraging these open-source resources, paving the way for a more open, collaborative scientific future. This study provides an exhaustive review of open-source tools in CFD underscoring their unique contributions and potential to reshape the field.

2. Open-source Tools

The rising popularity of open-source CFD tools shown in Figure 1 can be attributed largely to their cost-effectiveness and can be customized as per user requirements, offering a clear benefit over commercial options. These tools are especially beneficial for small companies and individual researchers with limited budgets, as they eliminate the need for expensive licensing fees.

Fig. 1. CAE open-source simulation solvers [12]

Moreover, their customizable nature allows users to tailor the tools to their specific needs, flexibility often restricted in commercial software. One of the key advantages of open-source tools is their transparency. Users can inspect, verify, and test the code for accuracy and reliability, an essential aspect in scientific fields like CFD as shown in Figure 2(a) and Figure 2(b) both of which shows the comparison of the results obtained from commercial Ansys Fluent and open source software called OpenFOAM. This transparency not only fosters a deeper understanding of algorithms and numerical methods among students and researchers but also stimulates innovation in fluid dynamics research. Additionally, the robust support community and continuous development associated with open-source tools enhance their reliability and offer assistance, simplifying both learning and problem-solving.

These tools are also versatile, capable of being used in diverse areas and easily integrated into various workflows. In essence, open-source tools are transforming CFD by making sophisticated, adaptable tools widely accessible, fostering a collaborative environment conducive to learning and innovation. This is well exemplified by the use of OpenFOAM in small-scale aerospace projects, which has shown significant cost savings and flexibility, demonstrating the practical advantages of these tools in real-world applications. To enhance the user-friendliness of open-source software, developers have integrated tools like OpenFOAM into more accessible GUI environments, complemented by additional software for pre- and post-processing, such as Visual-CFD, HELYX, and simFlow. While these wrappers offer the convenience of a unified interface, they introduce an extra layer between the user and the execution code. The main advantage of using wrappers is their ability to provide some benefits of comprehensive commercial platforms at a lower cost. However, they also have downsides, including not addressing some core limitations of open-source software, such as limited user support and specialized features. Additionally, these wrappers bring their own potential issues, including bugs and possibly inadequate support and development.

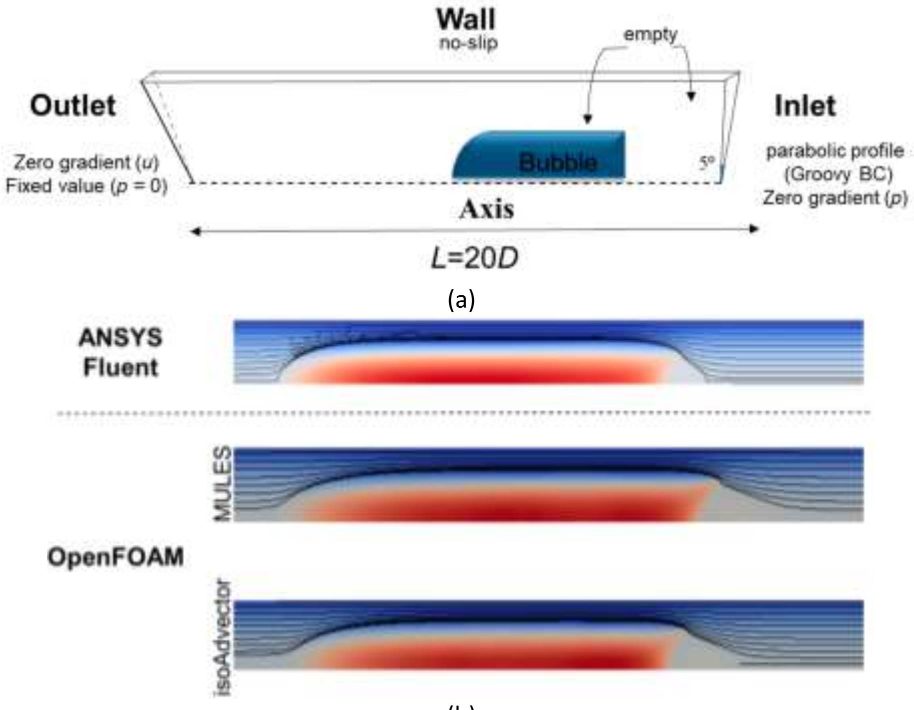


Fig. 2. (a) Domain and boundary conditions of bubble with OpenFOAM, (b) Comparison between ANSYS Fluent and OpenFOAM analysis with velocity streamlines at Re = 10 [13]

3. Comparison between Commercial and Open-source Tools

An open-source tool, is free and allows extensive customization, making it ideal for specialized research or unique simulation demands. Its flexibility is enhanced by the ability to modify and extend the code, especially useful for those proficient in programming skills. However, its less intuitive interface and steep learning curve, coupled with community-driven support, make it more challenging to master. Open-source tools are particularly accessible for individuals, small companies, or educational institutions with limited budgets. On the other hand, a commercial tool, is known for its user-friendly interface, making it easier for newcomers. It offers regular updates, dedicated customer support, and technical training as part of its licensing. Commercial tools seamlessly integrate with other tools for multi-physics simulations and provides advanced models and features right out of the box. However, its significant cost can be a barrier for smaller entities, and it offers less flexibility for customization than open-source. The license varies in cost from thousands to lakhs depending on usage, and there is a limitation on cell/nodes for student licenses. Open-source is more suited for users seeking deep customization, possessing programming skills, and engaged in unique or research-focused projects. Commercial tools cater more to commercial applications and users favoring an intuitive interface with less emphasis on customization. The choice between them hinges on the specific needs, budget, and expertise of the user or organization.

Computational Fluid Dynamics (CFD) is a specialized field in engineering and physics that involves the simulation of fluid flow and heat transfer using numerical methods. CFD software can be broadly classified into two categories: commercial and open-source. In this comparison in Table 1, various aspects of both types to help users make informed decisions based on their specific needs.

Table 1
Comparison between commercial and open-source entities based on various factors

Comparative Factor	Commercial	Open-Source
Cost	• Typically involves licensing fees, which can be substantial. • Additional costs for maintenance, updates, and support.	• Free to download and use, reducing financial barriers. • However, costs may still be incurred for training, support and hardware.
Code Accessibility and Transparency	• Closed-source, limiting the ability to modify or customize the code. • Users rely on the software provider for updates and bug fixes.	• Source code is accessible, allowing users to modify and customize. • Community-driven development often leads to frequent updates and bug fixes.
Community Support	• Typically comes with professional support services. • User forums and community support may be available but might not be as extensive.	• Relies on community support, which can be strong and active. • Extensive user forums and collaborative development.
Solver Capabilities	• Often provides a wide range of pre-built, validated solvers for various applications. • Proprietary solvers may be more user-friendly for non-experts.	• Variety of solvers available, and users can modify or develop their own. • May require more expertise to set up and use effectively.
User Interface and Ease of Use	• Generally comes with a polished, user-friendly graphical interface. • Aimed at a broader audience, including engineers with limited programming knowledge.	• User interfaces can vary in quality; some may be less intuitive. • Greater flexibility but may require more technical expertise.
Application Range	• Often covers a broad spectrum of applications with specialized modules. • Well-suited for industries with specific requirements and regulations.	• Flexibility to adapt to a wide range of applications but may require more user effort. • Well-suited for research, academic, and non-commercial purposes.
Integration with their Software	• May have better integration with other proprietary engineering software. • Plug-and-play solutions for specific industries.	• Open standards facilitate interoperability but may require more user effort. • Integration with other open-source tools may be straightforward.
Documentation and Training	• Generally comes with comprehensive documentation and user manuals. • Professional training programs may be available.	• Documentation quality can vary, but community-driven projects often have extensive guides. • Learning resources may include tutorials, online courses, and community forums.

The choice between commercial and open-source CFD software depends on various factors, including budget, application requirements, user expertise, and the level of customization needed. Commercial software may be preferable for industries with specific needs and a larger budget, while open-source options provide flexibility, accessibility, and community-driven support for research and educational purposes.

4. Foundations of Computational Fluid Dynamics

In a typical CFD analysis shown in Figure 3, defining the problem and the objectives of the study is the pivotal first step, requiring a thorough understanding of the physical processes involved, such as flow, heat transfer, and chemical reactions. This is followed by the selection of appropriate physical models, geometric modelling, mesh generation, and the meticulous assignment of boundary conditions. With the problem thus defined and contextualized, suitable numerical algorithms and solvers are employed to compute the flow and heat transfer. Following simulations, results are analysed, validated, and refined if necessary, culminating in comprehensive documentation and recommendations for design improvements or further analyses. Mastery of fluid mechanics, numerical methods, and specific software is crucial throughout the process to ensure the reliability and accuracy of the CFD analyses. Open-source tools provide a versatile and accessible framework for these foundational steps, offering extensive libraries and community-driven support for model selection, geometric modelling, and boundary condition assignment.

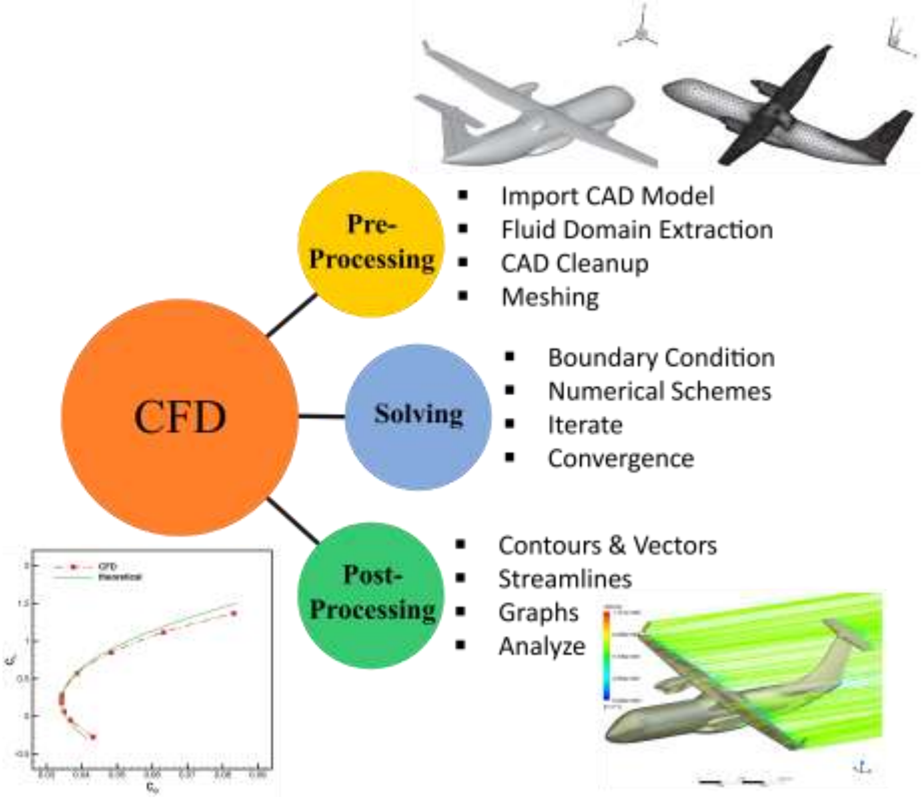


Fig. 3. Primary steps involved in CFD analysis [14,15]

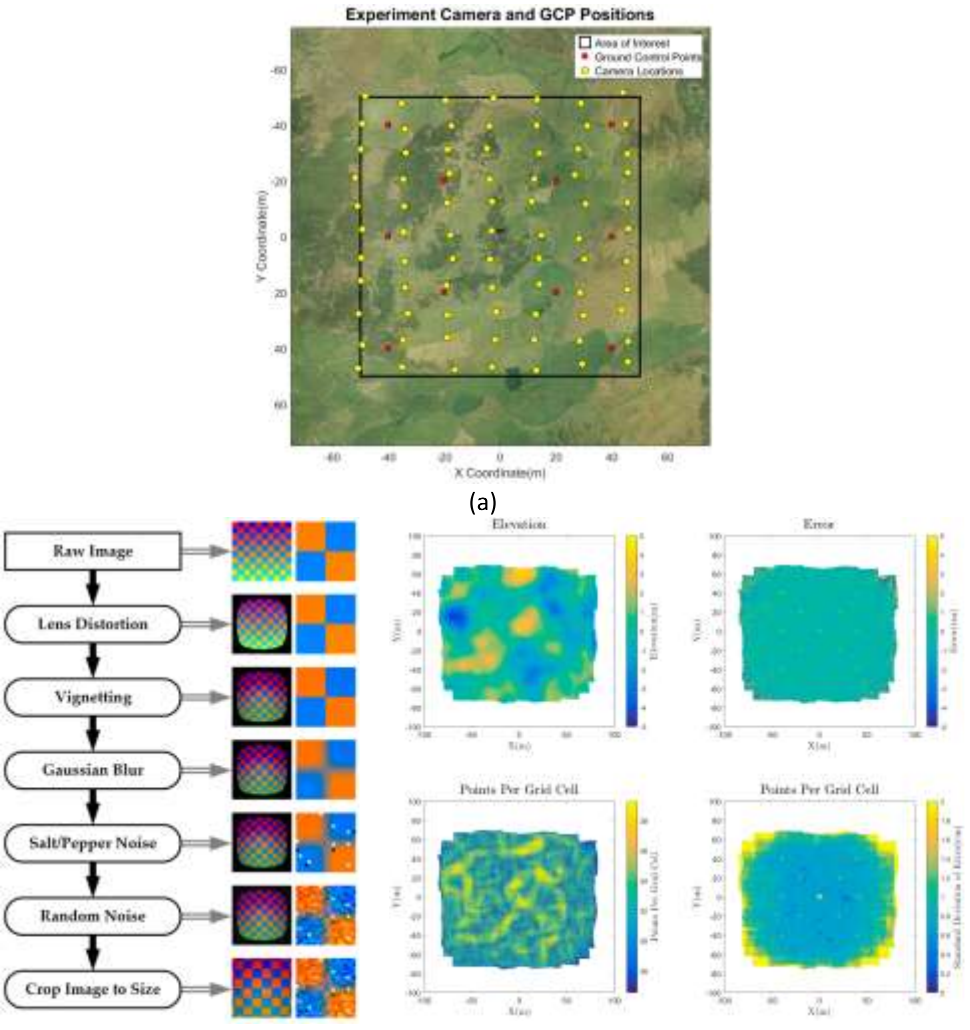
4.1 Utilizing Open-Source Tools for Precision Geometry Modeling in CFD

Constructing precise and refined model geometry is fundamental in CFD simulations, serving as the foundation for meshing and subsequent analytical tasks. The efficacy of these simulations heavily relies on acquiring meticulously prepared and refined CAD files, which detail essential geometry descriptions for prescribing accurate physical conditions and facilitating controlled meshing. Simplification and error removal are integral in focusing on the fluid flow in the spaces between solid objects and mitigating unnecessary complexities, especially in intricate components. Creating geometrical models that are compatible with CFD meshing frequently becomes a lengthy and challenging part of the CFD analysis process. It is essential to accurately depict the actual system to efficiently produce a computational grid. This grid is a key element in CFD simulations, as it needs to be sufficiently detailed to capture all relevant scales for the specific issue being addressed, yet not overly detailed to avoid excessively increasing the computation time [16].

In the spectrum of tools facilitating advanced modeling and meshing, OpenVSP (Open Vehicle Sketch Pad), distinguishes itself as an open-source parametric aircraft geometry tool, initially developed by NASA [17-20]. This tool is designed to create 3D models of aircraft and to support the engineering analysis of these models. It permits users to quickly translate ideas into computer models that can be further analyzed, proving invaluable for generating and evaluating unconventional design concepts. OpenVSP offers a multitude of basic and advanced geometries common to aircraft modeling, which users modify and assemble to create models. Alongside geometry modeling, OpenVSP encompasses a variety of tools, including CompGeom and VSPAERO, aiding in aerodynamic or structural analysis of models [18]. It allows importing and exporting of various geometry formats like STL, CART3D (.tri), and PLOT3D, enhancing its utility in mesh generation and in CFD or FEA software.

Among other notable tools are FreeCAD, a considerable alternative to commercial CAD packages; OpenSCAD, known for enabling the creation of accurate 3D models and parametric designs through script-based modeling; and Blender, recognized for its robust 3D modeling engine [21-30]. Wings 3D is pivotal for learning 3D modeling basics with its advanced subdivision modeling techniques, while SketchUp serves as an ideal starting point for architectural modelling [31-33]. Onshape offers collaborative, cloud-based 3D modeling solutions, and MeshLab provides extensive features for processing and editing 3D triangular meshes [34-38]. Lastly, SALOME stands out as a versatile platform, encompassing a broad spectrum of applications from 3D modeling to post-processing in various industrial sectors [39-41]. Figure 4 shows one such example of usage of Blender for UAS-Based Photogrammetry [42]. Additionally, BRL-CAD is an open-source, cross-platform solid modelling system that includes a suite of tools for geometry editing, ray-tracing, image and signal processing, among others [43,44]. Finally, SolveSpace is identified as a user-friendly parametric 3D CAD program, offering both 2D sketching and 3D modelling with a focus on constraint-based modelling [45,46].

These tools, each with its own unique capabilities and distinctive functionalities, play an instrumental role in streamlining the intricate tasks of geometry construction, modelling, and meshing in CFD simulations. They ensure the simplification and optimization of intricate geometries, efficient management of computational resources, and accuracy in the representations of physical systems. By integrating accurate geometric modelling with advanced meshing solutions provided by these tools, professionals can achieve highly refined and precise simulations, marking them as indispensable entities in contemporary engineering landscapes.



(a)

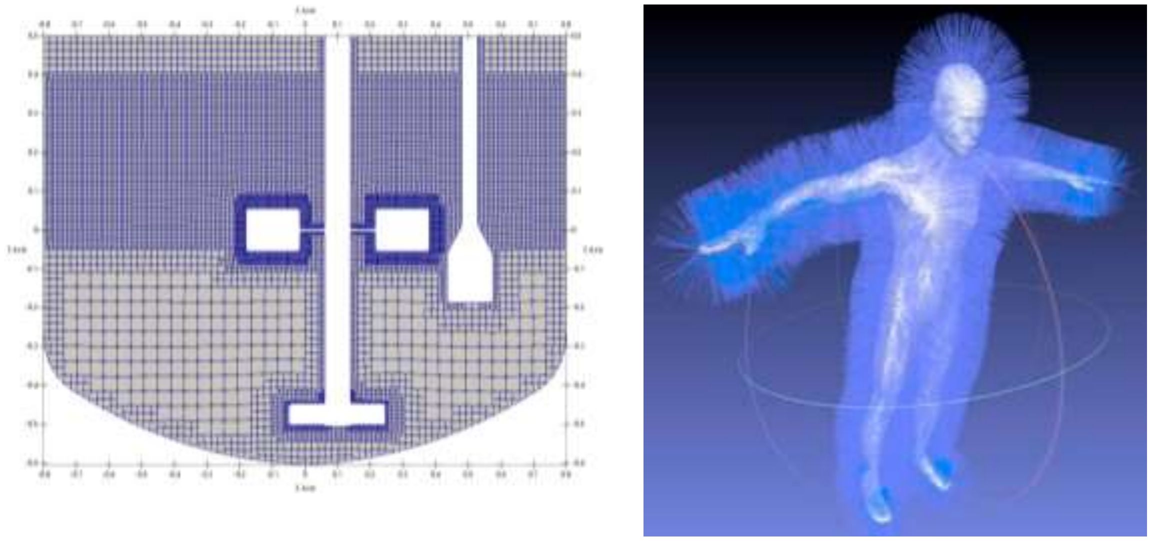
(b) (c)

Fig. 4. (a) Experimental Photogrammetry (b) Blender imagery rendered using pin hole camera model (c) Binning gridding algorithm and its usage in visualizing errors and PPGC [42]

4.2 Meshing Techniques and Open-Source Tools in Computational Simulations

Meshing is crucial in software-based simulations like Finite Element Analysis (FEA) and, more prominently, in CFD as it influences the simulations' accuracy and resource demands [47,48]. In CFD, the mesh quality often dictates the model's efficacy, impacting convergence, memory needs, and solution precision. A robust mesh must avoid void regions and overlapping elements while maintaining high quality, sufficient resolution, and minimal computational cost. The structured and unstructured meshing techniques allow for precise transformations of continuous geometric entities into definable shapes, adapting to intricate designs with varying regularity. Modern meshing tools, many of which are open-source, have provisions for automatic checks or offer solutions to detect and amend transgressions in mesh creation, ensuring the absence of elements with zero or negative volume that render the equations extremely challenging to solve. They focus on optimizing the aspect ratio, mitigating skewness, and controlling growth rate, which is critical for maintaining local accuracy and manageable equation conditions, particularly when transitioning from high-aspect-ratio elements to isotropic elements in areas like boundary layer meshes. These optimizations are vital for maintaining solid and watertight mesh geometries in fluid flow simulations, enabling solvers to identify the correct flow domain and facilitating precise assignments of governing equations to distinct cells. Such meticulous approaches to meshing are imperative for achieving a balanced, efficient, and accurate representation in numerical analyses, particularly in areas subjected to significant stress or located in the load path, ultimately advancing the comprehensive capabilities of computational simulations in varied scientific domains.

It is here that software tools like Netgen/NGSolve and Gmsh become pivotal, providing high-performance solutions and seamless integrations for analyzing models from solid mechanics to electromagnetics [41,49-54]. Netgen/NGSolve is renowned for its flexible Python interface and seamless integration from geometric modeling to visualization [55]. In contrast, Gmsh serves as a 3D finite element mesh generator, emphasizing a user-friendly and modular approach with efficient interactions with Netgen for mesh adaptations [56,57]. Additionally, the CFD General Notation System (CGNS) focuses predominantly on compressible viscous flow data, serving as a standard for data storage and retrieval in CFD analysis, enabling data exchange and archiving of aerodynamic data [58-61]. SALOME, a comprehensive open-source scientific computing environment, integrates physics solvers and offers modules accessible through GUI and Python scripts. CalculiX, another notable software, provides extensive support and integration options, hosting implicit and explicit solvers and offering functionalities analogous to commercial FEM programs like Abaqus [62]. Solutions like Overture, specializing in solving partial differential equations (PDEs), and OpenFOAM, providing premier open-source mesh tool suitable for varied complexities, contribute to a diversified and flexible software development environment for simulating physical processes in intricate moving geometries [41,63-68]. Lastly, enGrid, designed specifically for CFD applications, emphasizes mesh generation and supports automatic prismatic boundary layer grids for Navier-Stokes simulations [69-71]. Each tool, with its unique characteristics, contributes to establishing a comprehensive ecosystem for CFD and FEA, advancing multiple scientific domains with their open-source nature, versatility, and extensive capabilities, thus underlining the importance of the meshing process in the realm of computational simulations and mechanics. Figure 5(a) and Figure 5(b) show the OpenFOAM-based BlockMesh and MeshLab as open-source grid tools, respectively [72].



(a) (b)

Fig. 5. (a) Meshing of an industrial reactor model in OpenFOAM BlockMesh, (b) Surface normal of a model in MeshLab [73,74]

4.3 Open-source CFD Solvers

In the realm of CFD modeling, the preliminary steps of preparing the geometry and creating the mesh are fundamental. It is paramount to choose the appropriate models to accurately represent the problem's physics, be it through transient or steady-state simulations, based on the flow's spatial or temporal variations. CFD solvers are software tools designed for simulating fluid flow and related phenomena using numerical methods and algorithms, mainly solving the Navier-Stokes equations. They are capable of addressing a variety of issues like airflow around objects, heat transfer, turbulent flows, and fluid-based chemical reactions. The process involves three stages: pre-processing (defining the problem and preparing the computational domain), solving (numerical solution of the equations), and post-processing (analysis and visualization of results). The effectiveness of a CFD solver is influenced by the model's complexity, the discretization methods used (like finite volume or finite element), and the available computational resources. Different solvers are tailored for specific types of problems, with some being better suited for incompressible flows and others for compressible or multiphase flows. The choice of solver depends on the specific needs of the problem, including flow regime, geometric complexity, and the required detail in the outcomes.

CFD software packages are generally categorized into CAD Embedded (SolidWorks Flow Simulation, Autodesk CFD, and ANSYS Discovery Live), Open-Source (OpenFOAM, SU2, MFIX and SimScale), Semi-Comprehensive (COMSOL CFD, CONVERGE CFD and NUMECA OMNIS), and Comprehensive (Simcenter STAR-CCM+ and ANSYS Fluent) shown in Figure 6 [75].

OpenFOAM is particularly noteworthy as a C++ toolbox, acclaimed for its versatility in solving a plethora of continuum mechanics problems, including complex fluid flows involving chemical reactions, turbulence, heat transfer, acoustics, solid mechanics, and electromagnetics. This software finds extensive application in numerous industries and academic institutions, endorsed by its vast user base. Similarly, other open-source software like SU2, Code_Saturne, Gerris, COOLFluiD, FreeFEM, OpenFVM, ReFRESCO, and TrioCFD, each with its unique features and capabilities, caters to diverse needs in the computational fluid dynamics landscape [76-94]. They range from solving the Navier-Stokes equations for various flows in Code_Saturne to providing a powerful component-based framework for high-performance computing in COOLFluiD [95].

Fig. 6. CFD software packages

Each piece of software serves as a cog in the extensive machinery of computational fluid dynamics, contributing to the sophisticated simulation of fluid flows in varied domains. In the realm of CFD, over a thousand solvers have been developed by global research entities and laboratories. These solvers are predominantly available for non-commercial utilization under the General Public

License. For instance, FluidX3D emerges as a distinct CFD software, excelling in lattice Boltzmann methods [96-100]. Renowned for its speed and efficient memory usage, it operates on all GPU platforms via OpenCL [101]. Created by Moritz Lehmann, FluidX3D is free for non-commercial purposes, aligning with educational, research, and amateur applications. Additionally, FLOWUnsteady presents itself as an open-source, variable-fidelity framework for unsteady aerodynamics and aeroacoustics [102]. It is based on the reformulated vortex particle method (rVPM), a brainchild of the FLOW Lab at Brigham Young University [103]. PALABOS, focusing on lattice Boltzmann methods, is adept in simulating fluid flows in intricate geometries and multiphase conditions [104,105]. Nektar++, a spectral/hp element framework, addresses a broad spectrum of scientific and engineering challenges, encompassing fluid dynamics and wave propagation shown in Figure 7 [106-108]. FEniCSx, though not solely a CFD solver, offers a comprehensive suite of free software for solving partial differential equations (PDEs), inclusive of fluid dynamics challenges [109-114]. CFDTool, a MATLAB-based toolbox, simplifies the learning of fluid dynamics basics, targeting educational and basic commercial applications [115-118]. Elmer, a multi-physics simulation software, integrates fluid dynamics with structural mechanics, electromagnetism, and heat transfer, making it suitable for coupled multi-physical problems [119-121]. BARAM, an open-source CFD software, is designed to streamline the learning process for text-based solvers, featuring a user-friendly graphical interface and incorporating OpenFOAM® solvers modified by NEXTFOAM under GPL [122-124]. The University of Liverpool's Solver, also under GPL, serves an educational purpose in CFD, encompassing Euler equations, Roe's solver, Harten's entropy correction, and other features for both steady and unsteady flows [125]. UCNS3D, an open-source solver for compressible flows on unstructured meshes, employs high-order methods apt for industrial-scale CFD challenges [126]. The HOS Solvers, released under GPLv3, play a pivotal role in naval engineering by facilitating nonlinear irregular wave generation in CFD, crucial for assessing loads on offshore structures [127,128].

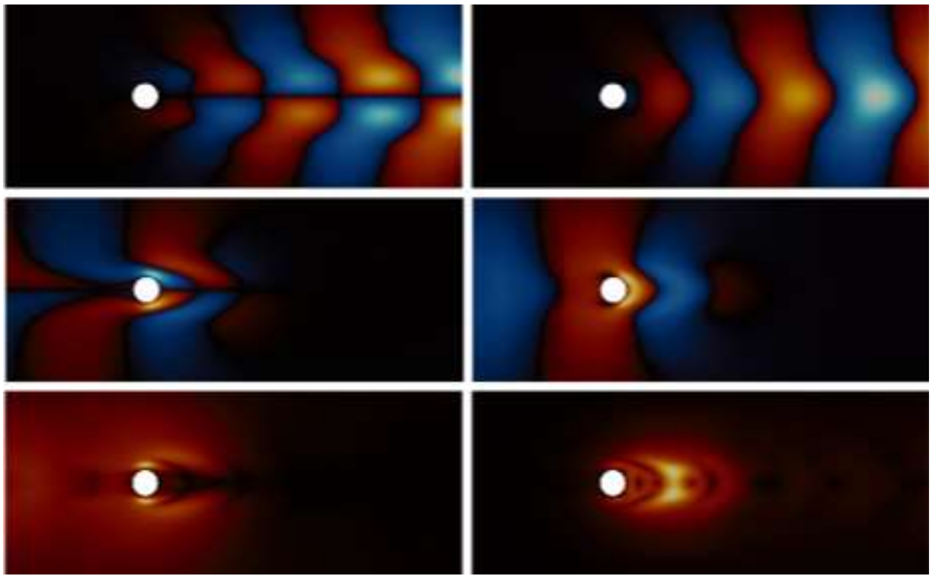


Fig. 7. Linear stability analyses of two-dimensional flow past a circular cylinder at $Re = 4$ in Nectar++ [106]

The compilation from Simon Wenkel's website adds other notable solvers like deal.II, Gerris, and Kratos Multiphysics [129-134]. deal.II, an open-source library, is instrumental in solving PDEs using finite element methods, primarily in computational science and engineering. Gerris, a flow solver, is recognized for its proficiency in managing complex geometric domains with adaptive mesh refinement. Kratos Multiphysics, an opensource framework, is acclaimed for its adaptability in handling diverse physical processes in computational simulations.

Flowsquare is a two-dimensional CFD software ideal for analyzing both unsteady and reactive/nonreactive fluid flows, including subsonic and supersonic flows [135,136]. Flowsquare is designed for ease of use and aims to make CFD technology more accessible for academic and educational purposes. It does not require expertise in meshing, programming, CAD, or pre/post-processing, allowing for straightforward simulation setups using bitmap images and text files. Flowsquare is used globally by individuals, companies, and educational institutions for various purposes.

APHROS is a finite volume solver, excels in simulating incompressible multiphase flows with surface tension, making it ideal for complex geometries and interactions between immiscible fluids [137,138]. hyStrath, another C++ code, is tailored for hypersonic and rarefied gas dynamics and is GPL-3.0 licensed [139,140]. FourierFlows.jl utilizes Julia to create adaptable pseudospectral solvers for partial differential equations [141,142]. Flow, by NVIDIAGameWorks, is a C-based library for real-time fluid simulation in sparse grid setups [143,144]. PteraSoftware, a Python package, is designed for the analysis of flapping-wing flight, offering speed and ease of use [145,146]. Nalu, leveraging the Sierra Toolkit and Trilinos solver stack, is a versatile, unstructured low Mach flow code suitable for a range of applications [147-149]. FluidFoam enhances OpenFOAM postprocessing with its Python-based tools. Incompact3d, developed in Fortran, optimizes Navier-Stokes equation solutions for current CPU architectures [150-152]. The 2d-fluidsimulator, implemented in Taichi and Python, handles 2D incompressible fluid dynamics. Lastly, MagIC, a high-performance Fortran code, adeptly solves magneto-hydrodynamics equations in rotating spherical shells [153]. Each solver has its unique strengths and suitability for particular CFD problems, influenced by factors like geometric complexity, flow nature, the necessity for multi-physics coupling, and computational resource availability.

4.4 Open-source Post-processors in Computational Simulations

In the process of CFD analyses, post-processors serve as software tools that scrutinize, interpret, exhibit, and make sense of simulation outcomes. They transform the extensive data produced by CFD simulations into visual formats such as graphs, charts, or thermal mappings, varying with the software used. Engineers and scientists can uncover trends, understand fluid flow dynamics, and make informed decisions with these post-processing tools. Certain CFD solvers, including OpenFOAM, SU2, and Gmsh, are equipped with integral post-processors [75,76,78]. The post-processor in Gmsh can be augmented with custom plug-ins that modify existing visualizations or generate new perspectives based on the existing data.

ParaView is an open-source application extensively utilized for CFD simulations, founded on the Visualization Toolkit (VTK) [39,154-156]. It facilitates Python scripting scripting and batch processing for efficient operations as shown in Figure 8 [157]. VisIt offers various visualization techniques, including contour plots, volume rendering, 2D and 3D visualization as shown in Figure 9, for presenting complex data [158-162]. It handles both time-varying and structured data, allowing users to dissect specific areas or time intervals within datasets. VisIt supports numerous data formats used in scientific and technical domains, such as VTK, HDF5, NetCDF, and others [163-165]. It can immediately read data from experiments or simulations. OpenDX, MayaVi, and GNUplot represent a selection of the accessible open-source post-processing software solutions [166-168].

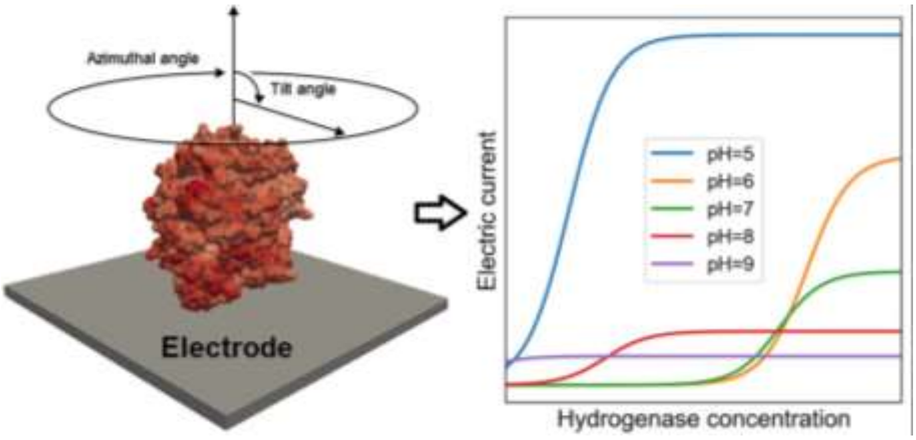
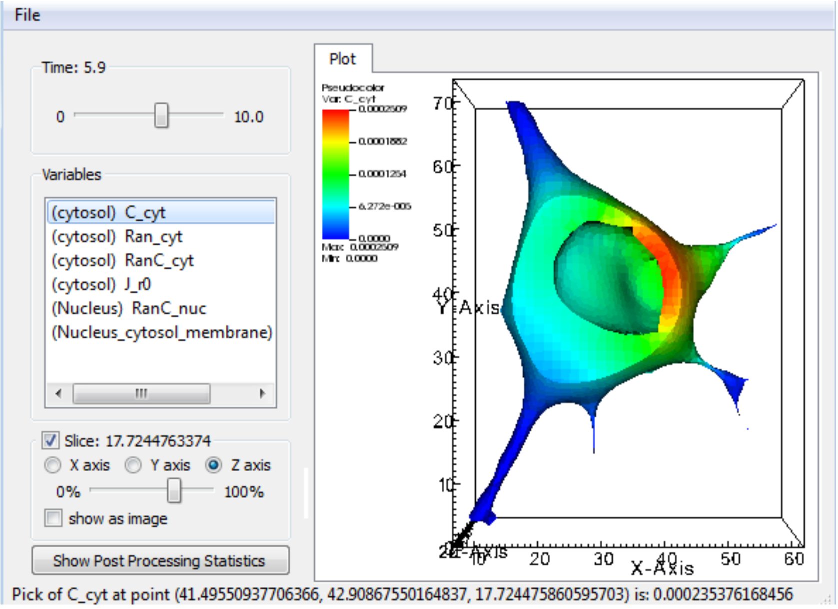


Fig. 8. ParaView visualization of electrostatic interaction and catalytic activity of [NiFe] hydrogenases on a planar electrode [169]



Time: 5.9

☑ Slice: 17.7244763374

○ X axis ○ Y axis ● Z axis

☐ show as image

Pick of C_cyt at point (41.49550937706366, 42.90867550164837, 17.724475860595703) is: 0.000235376168456

Fig. 9. A glimpse of the VisIt post-processing tool GUI [170]

Mayavi2 which is part of the Mayavi Project, utilizes Python to display scientific data, offering a range of capabilities for 3D scientific data visualization and plotting [171]. On the other hand, the highly scriptable GNUplot is a portable, command-line-driven, open-source tool for creating various types of charts and graphs, making it versatile for data visualization [172,173]. Open-source tool ENigMA implements several capabilities such as Mesh generation, post-processing, STL file processing, Smoothed Particle Hydrodynamics (SPH), PDEs, FEM, FVM, FDM, etc [174,175].

SimWorks, a complimentary CFD software, incorporates a unified GUI for OpenFOAM, the renowned open-source software pivotal for meshing and solving tasks in CFD [176,177]. Its capabilities span across multiple CFD analytical stages, encompassing geometry inspection, pre-processing, meshing, case setup, solving, and post-processing. Concurrently, Cassiopee, an innovation by ONERA, emerges as another open-source software that consolidates pre- and post-processing tools within a singular platform [178]. This integration aims to streamline various CFD analytical components, thereby offering a holistic solution that addresses both initial and final stages of CFD investigations. Additionally, Octave, as an open-source scientific programming language, positions itself as a MATLAB alternative, adept at executing a range of scientific computations, including those related to CFD data analysis and visualization [179,180].

4.5 Open-source Post Data Generation and Availability

Computational Fluid Dynamics (CFD) simulations generate vast amounts of data, and making this data available to the scientific community is crucial for collaboration, validation, and further research. Commercial software such as ANSYS Fluent, COMSOL Multiphysics, and Siemens STAR-CCM+ are widely used for CFD simulations in various industries. These tools offer advanced solvers, pre-processing, and post-processing capabilities. Open-Source CFD Software like OpenFOAM, SU2, and FEniCS are popular open-source CFD software. They provide flexibility and customization, making them widely adopted in academic and research settings. Many CFD simulations require significant computational resources. HPC clusters and supercomputers are often used to perform large-scale simulations.

The platforms for hosting this massive amount of data are 1. Institutional Repositories: Universities and research institutions often have their own data repositories. Institutional repositories may use platforms like DSpace, Figshare, or Dataverse to store and share CFD data 2. Public Data Repositories: Public repositories such as Zenodo, Dryad, and DataCite accept CFD data submissions. These platforms provide Digital Object Identifiers (DOIs) for datasets, ensuring proper attribution and citation 3. Collaborative Platforms: Collaborative platforms like GitHub and GitLab are commonly used for sharing not only code but also input files, simulation setups, and post-processing scripts. These platforms facilitate version control and collaborative development 4. Domain-Specific Repositories: Some disciplines have specific repositories for CFD data. For example, all the subscription and Open-Source CFD International Journal encourages authors to provide data along with their publications 5. Journals and Conference Proceedings: CFD data can be published as supplementary material in scientific journals or conference proceeding. This allows researchers to share the data alongside their findings.

However, this comes with certain challenges and considerations. For instance, data privacy and sensitivity. Depending on the nature of the simulations, there may be privacy or proprietary concerns. Care must be taken to anonymize or exclude sensitive information. Another issue is about data format and metadata. Standardizing data formats (e.g., HDF5, NetCDF) and providing detailed metadata enhance the usability of the shared CFD data. Researchers should specify the licensing terms for the shared data, ensuring proper attribution and adherence to copyright policies. Large datasets may pose challenges for storage, transfer, and download. Consideration should be given to the scalability of hosting platforms. Ensuring long-term accessibility and preservation of CFD data is essential. Repositories with sustainable funding and archiving practices are preferable. So, the availability of CFD data relies on a combination of suitable tools for simulation and appropriate platforms for data hosting. Open-access repositories, collaborative platforms, and domain-specific repositories contribute to the dissemination of CFD data, fostering collaboration and advancing scientific knowledge in fluid dynamics [181].

4.6 Reproducibility Problem in CAE Applications

Reproducibility issues in Computer-Aided Engineering (CAE) applications can pose significant challenges for researchers and the broader scientific community. These problems often arise due to the lack of sharing source code, difficulties in submitting changes to mainstream repositories, and the resulting need for researchers to reinvent solutions. Here are some key aspects contributing to these challenges:

(i) Closed-Source Nature of CAE Software

Many CAE applications are proprietary and closed-source, restricting access to the underlying code. This makes it difficult for researchers to understand, modify, or extend the software to address specific needs or improve its functionality

(ii) Difficulty in Submissions to Mainstream Repositories

Submitting changes or improvements to mainstream software repositories can be challenging for several reasons. The development processes of established software projects may be rigid, making it difficult for external contributions to be accepted. Additionally, legal and licensing issues may further complicate the integration of external code into existing projects.

(iii) Lack of Documentation

Even when researchers are willing to share their code, the lack of comprehensive documentation can hinder reproducibility. Incomplete or unclear documentation makes it difficult for others to understand and use the code effectively.

(iv) Funding and Time Constraints

Researchers may face constraints in terms of funding and time, which can limit their ability to share code, document their work thoroughly, or contribute to mainstream repositories. These constraints contribute to the challenges of reproducibility in the CAE domain.

(v) Limited Code Sharing

In the CAE community, there is often a lack of culture surrounding the sharing of source code. Researchers may be hesitant to share their code due to concerns about intellectual property, competition, or simply because they have not prioritized making their code publicly available.

(vi) Versioning and Compatibility Issues

CAE applications often involve complex dependencies on hardware, libraries, and other software components. This can lead to versioning and compatibility issues, making it challenging to reproduce results when different versions of the same software or its dependencies are used.

(vii) High Computational Requirements

CAE simulations often require significant computational resources, including specialized hardware and software configurations. This makes it difficult for researchers with limited resources to replicate and validate the results obtained by others.

To address these issues, fostering a culture of openness, collaboration, and code sharing within the CAE community is crucial. Encouraging researchers to provide clear documentation, use open-source licensing, and actively contribute to relevant repositories can significantly enhance the reproducibility and transparency of CAE research. Additionally, efforts to develop community-driven standards for code sharing and integration could help streamline the process of submitting changes to mainstream repositories [182].

4.7 Advantages and Disadvantages of OS Software

Open-source CFD software has gained popularity due to its accessibility, flexibility, and collaborative nature. However, like any technology, it comes with its own set of advantages and disadvantages from a user perspective. Table 2 below shows the advantages and disadvantages of open source software.

Table 2
Pros and cons of open-source entities and their details

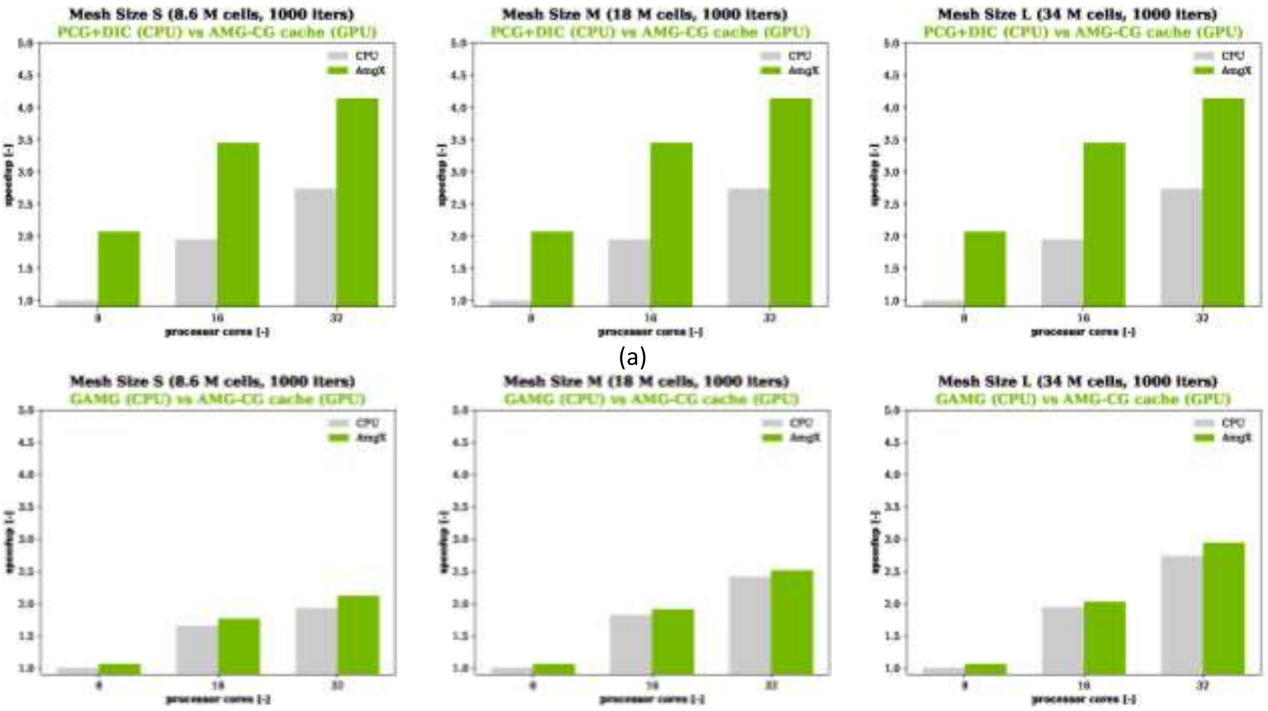
Advantages	Disadvantages
Cost: One of the most significant advantages is cost savings. Open-source software is typically free to download and use, making it an attractive option for individuals, academic institutions, and small businesses with limited budgets.	Limited User Interface and Documentation: Open-source CFD software may lack a user-friendly interface compared to commercial counterparts. Additionally, documentation may not be as comprehensive or accessible, which can steepen the learning curve for new users.
Flexibility and Customization: Users have the freedom to modify and customize the source code according to their specific needs. This level of flexibility is particularly beneficial for researchers and developers who want to tailor the software to their unique requirements.	Support and Training: While there is a community for support, it may not be as responsive or comprehensive as dedicated customer support from a commercial software vendor. Users may find it challenging to get timely assistance for specific issues.
Community Support: Open-source CFD software often has a large and active community of users and developers. This community support can be invaluable when seeking help, troubleshooting issues, or collaborating on the improvement of the software.	Integration Challenges: Integrating open-source CFD software into existing workflows or with other software tools may pose challenges. Compatibility issues and the need for additional customization can require extra effort.
Continuous Development and Updates: With a large community contributing to the software, updates and improvements are frequent. Users can benefit from the latest features, bug fixes, and enhancements without relying solely on the development roadmap of a commercial vendor.	Commercial Code Features: Some advanced features found in commercial CFD software may not be available in open-source alternatives. Users with specific requirements may find that certain capabilities are only provided by proprietary solutions.
Transparency: The open nature of the source code provides transparency, allowing users to understand how the algorithms work. This transparency is essential for research purposes and for building trust in the accuracy and reliability of the simulations.	Stability and Validation: Open-source software may undergo rapid development, leading to potential instability in certain releases. Additionally, the lack of strict validation processes compared to commercial software may raise concerns about the accuracy of results in critical applications.

5. Role of AI-ML-DL and High-performance Computing in CFD using Open-source Tools

Traditional computational fluid dynamics (CFD) methods involve solving partial differential equations governing fluid flow behavior using numerical techniques. However, these methods encounter difficulties in handling complexities such as intricate geometries, turbulent flows, and multi-physics interactions. The integration of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) techniques into CFD workflows offers promising solutions to these challenges, unlocking new capabilities in fluid dynamics simulations. Open-source tools and software platforms provide accessible and customizable environments for implementing and experimenting with AI-driven approaches in CFD simulations.

In addition to software like SimScale based on cloud computing (with free access for limited hours nearly one time 3000 computational hours and 500 GB storage) compared to commercial options like Ansys, open-source high-fidelity codes have been utilized for various applications ranging from reconfigurable systems to intricate problems like insect aerodynamics, flow topology and bird flapping kinematics and aerodynamics [183-188]. AI and ML algorithms play a crucial role in enhancing turbulence modelling by leveraging large datasets of experimental or high-fidelity simulation results. Techniques such as neural networks excel in capturing complex flow features, leading to more accurate turbulence closure models [189].

DL-based surrogate models offer an alternative to computationally expensive CFD simulations for tasks such as parameter optimization, sensitivity analysis, and uncertainty quantification. These surrogate models provide faster evaluations while maintaining acceptable accuracy levels. AI algorithms automate complex geometries and meshing processes, reducing manual intervention and enhancing efficiency. DL methods, like generative adversarial networks (GANs), can generate realistic geometries based on specified design criteria [189]. ML techniques enable real-time flow control and optimization by learning from simulation data and adjusting control strategies accordingly. Reinforcement learning algorithms optimize parameters to achieve desired flow characteristics or performance metrics. AI-driven approaches detect anomalies in CFD simulations and rectify errors to enhance reliability and robustness. ML models trained on historical simulation data can identify discrepancies and propose corrective actions during runtime [190]. Various open-source tools and software platforms facilitate the integration of AI, ML, and DL techniques into CFD simulations. For example, using dynamic mode decomposition which is a machine learning based reduced order modeling technique or using neural networks and sparse algorithms for accelerated CFD [190]. OpenFOAM, for instance, supports customization and extension for implementing AI-driven algorithms. TensorFlow, PyTorch, scikit-learn, and Keras are popular frameworks for building and training neural networks and machine learning models in Python. Additionally, SU2 and other high-fidelity open-source CFD codes offer capabilities for adjoint-based optimization and uncertainty quantification, compatible with ML integration. Some of these open source codes for complex problems need high computational facility like GPU platforms to get excellent results as shown in Figure 10 [191]. Thus, the integration of AI, ML, and DL techniques revolutionizes the field of computational fluid dynamics, addressing challenges and unlocking new possibilities for simulation accuracy, efficiency, and automation. Open-source tools and software platforms democratize access to these advanced techniques, fostering collaboration and innovation in the CFD community. The era of relying on conventional methods for tackling straightforward tasks, for example like MHD stream based analysis or modified shape FSI is behind us [192,193]. Now, with open-source tools readily available, even the most intricate problems can be effortlessly addressed, devoid of concerns about commercial constraints.



(a)

(b)

Fig. 10. Scalability calculated across three tested grids (S, M, L). (a) PCG-DIC (CPU) vs. AMG-PCG (GPU) and (b) GAMG (CPU) vs. AMG-PCG (GPU) [191]

6. Conclusions

This article has conducted a thorough review of the myriads of open-source tools available for Computational Fluid Dynamics (CFD) and their significant role in propelling research in fluid dynamics. These tools, which range from advanced geometry modeling to sophisticated post-processing software, are not just alternatives to their commercial counterparts but are pivotal in democratizing advanced computational capabilities. By providing cost-effective, customizable, and transparent solutions, open-source tools are reshaping the CFD landscape, making it more accessible to a broader audience. They facilitate a culture of openness and collaboration, encouraging innovation and knowledge-sharing among researchers and practitioners. This is in line with the growing global movement towards open science, where transparency, reproducibility, and communal progress are at the forefront. As CFD continues to evolve, the adoption of open-source tools will undoubtedly play a crucial role in the sustainability and expansion of this field, ensuring that the scientific community can collectively tackle more complex and nuanced fluid dynamics challenges. This paper is exclusively focused on the analysis of incompressible open-source software, excluding considerations for compressible counterparts. It is important to note that there exist open-source codes designed to handle compressible flow scenarios as well. However, we intend to explicitly highlight the inherent incompressibility of the software under examination, providing a clearer understanding of their capabilities.

Acknowledgement

We authors would like to express our sincere gratitude to all those who contributed to the development and completion of this comprehensive review paper including the peer guidance. We express our gratitude the Manipal Institute of Technology, a constituent engineering institute of Manipal Academy of Higher Education, an Institution of Eminence that supported and facilitated the authors in their pursuit of knowledge and the dissemination of scientific insights.

Funding

This research receives no funding.

Conflict of Interest

Authors declare no conflict of interest.

References

[1] Bhaskaran, Rajesh, and Lance Collins. "Introduction to CFD basics." *Cornell University-Sibley School of Mechanical and Aerospace Engineering* (2002): 1-21.

[2] Ashgriz, Nasser, and Javad Mostaghimi. "An introduction to computational fluid dynamics." *Fluid Flow Handbook* 1 (2002): 1-49.

[3] van Leer, Bram, and Kenneth G. Powell. "Introduction to computational fluid dynamics." *Encyclopedia of Aerospace Engineering* (2010). https://doi.org/10.1002/9780470686652.eae048

[4] Roelofs, F., and A. Shams. "CFD-introduction." In *Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors*, pp. 213-218. Woodhead Publishing, 2019. https://doi.org/10.1016/B978-0-08-101980-1.00006-5

[5] Spalart, Philippe R., and V. Venkatakrishnan. "On the role and challenges of CFD in the aerospace industry." *The Aeronautical Journal* 120, no. 1223 (2016): 209-232. https://doi.org/10.1017/aer.2015.10

[6] Viviani, Antonio, Andrea Aprovitola, Giuseppe Pezzella, and Cinzia Rainone. "CFD design capabilities for next generation high-speed aircraft." *Acta Astronautica* 178 (2021): 143-158. https://doi.org/10.1016/j.actaastro.2020.09.006

[7] Loureiro, Eric Vargas, Nicolas Lima Oliveira, Patricia Habib Hallak, Flávia de Souza Bastos, Lucas Machado Rocha, Rafael Grande Pancini Delmonte, and Afonso Celso de Castro Lemonge. "Evaluation of low fidelity and CFD methods for the aerodynamic performance of a small propeller." *Aerospace Science and Technology* 108 (2021): 106402. https://doi.org/10.1016/j.ast.2020.106402

[8] Mani, Mori, and Andrew J. Dorgan. "A perspective on the state of aerospace computational fluid dynamics technology." *Annual Review of Fluid Mechanics* 55 (2023): 431-457. https://doi.org/10.1146/annurev-fluid-120720-124800

[9] Pantusheva, Mariya, Radostin Mitkov, Petar O. Hristov, and Dessislava Petrova-Antonova. "Air pollution dispersion modelling in urban environment using CFD: a systematic review." *Atmosphere* 13, no. 10 (2022): 1640. https://doi.org/10.3390/atmos13101640

[10] Bournet, Pierre-Emmanuel, and Fernando Rojano. "Advances of Computational Fluid Dynamics (CFD) applications in agricultural building modelling: Research, applications and challenges." *Computers and Electronics in Agriculture* 201 (2022): 107277. https://doi.org/10.1016/j.compag.2022.107277

[11] Li, Yangluxi, Lei Chen, and Li Yang. "CFD modelling and analysis for green environment of traditional buildings." *Energies* 16, no. 4 (2023): 1980. https://doi.org/10.3390/en16041980

[12] Welahettige, Prasanna, and Knut Vaagsaether. "Comparison of OpenFOAM and ANSYS FLUENT." In *Proceedings of 9th EUROSIM Congress on Modelling Simulation, EUROSIM*, pp. 1005-1012. 2016.

[13] Silva, Mónica F., João BLM Campos, João M. Miranda, and José DP Araújo. "Numerical study of single taylor bubble movement through a microchannel using different CFD packages." *Processes* 8, no. 11 (2020): 1418. https://doi.org/10.3390/pr8111418

[14] LearnCAx. "Video tutorial: Fluid Domain Extraction of a Manifold using ANSYS ICEM CFD." *LearnCAx*. Accessed October 21, 2023. https://www.learncax.com/knowledge-base/cfd-tutorials/video-tutorial-fluid-domain-extraction-of-a-manifold-using-ansys-icem-cfd.html.

[15] Natale, Nunzio, Teresa Salomone, Giuliano De Stefano, and Antonio Piccolo. "Computational evaluation of control surfaces aerodynamics for a mid-range commercial aircraft." *Aerospace* 7, no. 10 (2020): 139. https://doi.org/10.3390/aerospace7100139

[16] CFD Support. "CAE Open Source Software." *CFD Support*. Accessed November 1, 2023. https://www.cfdsupport.com/cae-open-source-software.html.

[17] Hu, Beichao, and Dwayne McDaniel. "Applying Physics-Informed Neural Networks to Solve Navier-Stokes Equations for Laminar Flow around a Particle." *Mathematical and Computational Applications* 28, no. 5 (2023): 102. https://doi.org/10.3390/mca28050102

[18] McDonald, Robert A., and James R. Gloudemans. "Open vehicle sketch pad: An open source parametric geometry and analysis tool for conceptual aircraft design." In *AIAA SciTech 2022 Forum*, p. 0004. 2022. https://doi.org/10.2514/6.2022-0004

[19] Hahn, Andrew. "Open Vehicle Sketch Pad Aircraft Modeling Strategies." In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 331. 2012. https://doi.org/10.2514/6.2013-331

[20] Gloudemans, James, Paul Davis, and Paul Gelhausen. "A rapid geometry modeler for conceptual aircraft." In *34th Aerospace Sciences Meeting and Exhibit*, p. 52. 1996. https://doi.org/10.2514/6.1996-52

[21] Riegel, Jürgen. "FreeCAD - A free extensible CAx system." *FreeCAD*, 2002. https://wiki.opensourceecology.org/images/3/36/Freecadspec2002.pdf.

[22] Castro, Hélio, Goran Putnik, Alrenice Castro, and Rodrigo Dal Bosco Fontana. "Open design initiatives: an evaluation of CAD Open Source Software." *Procedia CIRP* 84 (2019): 1116-1119. https://doi.org/10.1016/j.procir.2019.08.001

[23] Gayer, D., C. O'Sullivan, S. Scully, D. Burke, J. Brossard, and C. Chapron. "FreeCAD visualization of realistic 3D physical optics beams within a CAD system-model." In *Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VIII*, vol. 9914, pp. 745-758. SPIE, 2016. https://doi.org/10.1117/12.2233268

[24] Nilsiam, Yuenyong, and Joshua M. Pearce. "Free and open source 3-D model customizer for websites to democratize design with OpenSCAD." *Designs* 1, no. 1 (2017): 5. https://doi.org/10.3390/designs1010005

[25] Bashkatov, Alexander Mayorovich. "Modeling in OpensCAD: examples." *Bachelor's Thesis, Pridnestrovian State University*, 2020. https://doi.org/10.12737/959073

[26] Machado, Felipe, Norberto Malpica, and Susana Borromeo. "Parametric CAD modeling for open source scientific hardware: Comparing OpenSCAD and FreeCAD Python scripts." *PloS One* 14, no. 12 (2019): e0225795. https://doi.org/10.1371/journal.pone.0225795

[27] Dow, Colin. *Simplifying 3D Printing with OpenSCAD: Design, build, and test OpenSCAD programs to bring your ideas to life using 3D printers*. Packt Publishing, 2022.

[28] Filippov, Sergey Valeryevich. "Blender Software Platform as an Environment for Modeling Objects and Processes of Natural Science Disciplines." *Keldysh Institute Preprints* 230 (2018): 1-42. https://doi.org/10.20948/prepr-2018-230

[29] Flavell, Lance. *Beginning blender: open source 3d modeling, animation, and game design*. Apress, 2011. https://doi.org/10.1007/978-1-4302-3127-1

[30] Patoli, Muhammad Zeeshan, Michael Gkion, Abdullah Al-Barakati, Wei Zhang, Paul Newbury, and Martin White. "An open source grid based render farm for blender 3d." In *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1-6. IEEE, 2009. https://doi.org/10.1109/PSCE.2009.4839978

[31] Chopra, Aidan. *Introduction to google sketchup*. John Wiley & Sons, 2012.

[32] Al-Zubaydi, Ahmed Y. "Building models design and energy simulation with google sketchup and openstudio." *Journal of Advanced Science and Engineering Research* 3, no. 4 (2013): 318-333.

[33] Hong, Tang, and Wang Chun-Xia. "Discussing sketchup software in the application of architectural design teaching." *Journal of Theoretical & Applied Information Technology* 48, no. 2 (2013).

[34] Moss, Elise. *Getting started with onshape*. SDC Publications, 2023.

[35] Leipold, Kate N. "CAD for College: Switching to Onshape for engineering design tools." In *2020 ASEE Virtual Annual Conference Content Access*. 2020.

[36] Cignoni, Paolo, Guido Ranzuglia, M. Callieri, M. Corsini, F. Ganovelli, N. Pietroni, and M. Tarini. *MeshLab*. Università degli Studi di Milano, 2011.

[37] Cignoni, Paolo, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. "Meshlab: an open-source mesh processing tool." In *Eurographics Italian Chapter Conference*, vol. 2008, pp. 129-136. 2008.

[38] Callieri, Marco, Guido Ranzuglia, Matteo Dellepiane, Paolo Cignoni, and Roberto Scopigno. "Meshlab as a complete open tool for the integration of photos and colour with high-resolution 3D geometry data." *Computer Applications and Quantitative Methods in Archaeology* (2012): 406-16.

[39] Ribés, Alejandro, and Adrien Bruneton. "Visualizing results in the SALOME platform for large numerical simulations: An integration of ParaView." In *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 119-120. IEEE, 2014. https://doi.org/10.1109/LDAV.2014.7013218

[40] Ribes, Andre, and Christian Caremoli. "Salome platform component model for numerical simulation." In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, vol. 2, pp. 553-564. IEEE, 2007. https://doi.org/10.1109/COMPSAC.2007.185

[41] Kortelainen, Juha. *Meshing Tools for Open Source CFD: A Practical Point of View*. Research Report VTT-R-02440-09, 2009.

[42] Slocum, Richard K., and Christopher E. Parrish. "Simulated imagery rendering workflow for UAS-based photogrammetric 3D reconstruction accuracy assessments." *Remote Sensing* 9, no. 4 (2017): 396. https://doi.org/10.3390/rs9040396

[43] Dykstra, Phillip C., and Michael John Muuss. "The BRL-CAD package an overview." In *USENIX, Proceedings of the Fourth Computer Graphics Workshop*. 1987.

[44] Muuss, Mike. "BRL-CAD: Solid Modeling and CAD Software." *BRL-CAD*. Accessed November 1, 2023. https://brlcad.org/.

[45] GitHub. "SolveSpace." *GitHub*. Accessed October 29, 2023. https://github.com/solvespace/solvespace.

[46] Cheng, Kathy, Phil Cuvin, Alison Olechowski, and Shurui Zhou. "User Perspectives on Branching in Computer-Aided Design." *Proceedings of the ACM on Human-Computer Interaction* 7, no. CSCW2 (2023): 1-30. https://doi.org/10.1145/3610220

[47] Bhavikatti, S. S. *Finite element analysis*. New Age International, 2005.

[48] Roylance, David. "Finite element analysis." *Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge* (2001).

[49] Arnold, Noam. "NETGEN/NGSolve Manual." *Master Thesis, University of Zurich*, 2013.

[50] Schöberl, Joachim. "C++ 11 implementation of finite elements in NGSolve." *Institute for Analysis and Scientific Computing, Vienna University of Technology* 30 (2014).

[51] Lehrenfeld, Christoph, Joachim Schöberl, and Arnold Reusken. "Hybrid discontinuous Galerkin methods for solving incompressible flow problems." *Rheinisch-Westfalischen Technischen Hochschule Aachen* 111 (2010).

[52] Remacle, J-F., Christophe Geuzaine, Gaëtan Compere, and Emilie Marchandise. "High-quality surface remeshing using harmonic maps." *International Journal for Numerical Methods in Engineering* 83, no. 4 (2010): 403-425. https://doi.org/10.1002/nme.2824

[53] Marchandise, Emilie, C. Carton de Wiart, W. G. Vos, Christophe Geuzaine, and J-F. Remacle. "High-quality surface remeshing using harmonic maps-Part II: Surfaces with high genus and of large aspect ratio." *International Journal for Numerical Methods in Engineering* 86, no. 11 (2011): 1303-1321. https://doi.org/10.1002/nme.3099

[54] Marchandise, Emilie, Jean-François Remacle, and Christophe Geuzaine. "Optimal parametrizations for surface remeshing." *Engineering with Computers* 30 (2014): 383-402. https://doi.org/10.1007/s00366-012-0309-3

[55] Hochsteger, Matthias. "High order discontinuous Galerkin methods on GPUs." *PhD diss., Vienna University of Technology*, 2014.

[56] Geuzaine, Christophe, and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities." *International Journal for Numerical Methods in Engineering* 79, no. 11 (2009): 1309-1331. https://doi.org/10.1002/nme.2579

[57] Gangl, Peter, Kevin Sturm, Michael Neunteufel, and Joachim Schöberl. "Fully and semi-automated shape differentiation in NGSolve." *Structural and Multidisciplinary Optimization* 63, no. 3 (2021): 1579-1607. https://doi.org/10.1007/s00158-020-02742-w

[58] Poirier, Diane, Steven Allmaras, Douglas McCarthy, Matthew Smith, and Francis Enomoto. "The CGNS system." In *29th AIAA, Fluid Dynamics Conference*, p. 3007. 1998. https://doi.org/10.2514/6.1998-3007

[59] Poirier, Diane, Robert Bush, Raymond Cosner, Christopher Rumsey, and Douglas McCarthy. "Advances in the CGNS Database Standard for Aerodynamics and CFD." In *38th Aerospace Sciences Meeting and Exhibit*, p. 681. 2000. https://doi.org/10.2514/6.2000-681

[60] Rumsey, Christopher, Bruce Wedan, Thomas Hauser, and Marc Poinot. "Recent updates to the CFD general notation system (CGNS)." In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 1264. 2012. https://doi.org/10.2514/6.2012-1264

[61] Legensky, S., D. Edwards, R. Bush, D. Poirier, C. Rumsey, R. Cosner, and C. Towne. "CFD general notation system (CGNS)-status and future directions." In *40th AIAA Aerospace Sciences Meeting & Exhibit*, p. 752. 2002. https://doi.org/10.2514/6.2002-752

[62] Wittig, Klaus. "CalculiX USER'S MANUAL-CalculiX GraphiX, Version 2.17." *CalculiX* (2016).

[63] Banks, Jeffrey W., Benjamin B. Buckner, William D. Henshaw, Michael J. Jenkinson, Alexander V. Kildishev, Gregor Kovačič, Ludmila J. Prokopeva, and Donald W. Schwendeman. "A high-order accurate scheme for Maxwell's equations with a generalized dispersive material (GDM) model and material interfaces." *Journal of Computational Physics* 412 (2020): 109424. https://doi.org/10.1016/j.jcp.2020.109424

[64] Meng, F., J. W. Banks, W. D. Henshaw, and D. W. Schwendeman. "Fourth-order accurate fractional-step IMEX schemes for the incompressible Navier-Stokes equations on moving overlapping grids." *Computer Methods in Applied Mechanics and Engineering* 366 (2020): 113040. https://doi.org/10.1016/j.cma.2020.113040

[65] Serino, Daniel A., Jeffrey W. Banks, William D. Henshaw, and Donald W. Schwendeman. "A stable added-mass partitioned (AMP) algorithm for elastic solids and incompressible flow." *Journal of Computational Physics* 399 (2019): 108923. https://doi.org/10.1016/j.jcp.2019.108923

[66] Henshaw, William D. "Automatic grid generation." *Acta Numerica* 5 (1996): 121-148. https://doi.org/10.1017/S0962492900002634

[67] Chen, Goong, Qingang Xiong, Philip J. Morris, Eric G. Paterson, Alexey Sergeev, and Y. Wang. "OpenFOAM for computational fluid dynamics." *Notices of the AMS* 61, no. 4 (2014): 354-363. https://doi.org/10.1090/noti1095

[68] Lysenko, Dmitry A., Ivar S. Ertesvåg, and Kjell E. Rian. "Modeling of turbulent separated flows using OpenFOAM." *Computers & Fluids* 80 (2013): 408-422. https://doi.org/10.1016/j.compfluid.2012.01.015

[69] Spekreijse, S. P. "ENGRID: A graphical interactive code for the computation of structured grids for blocked flow domains." In *10th NAL Symposium on Aircraft Computational Aerodynamics*, pp. 51-56. 1992.

[70] Spekreijse, Stephanus Petrus, and Jan Willem Boerstoel. "Multiblock grid generation. Part II: Multiblock aspects." In *27th Computational Fluid Dynamics Course, at the Von Karman Institute (VKI) for Fluid Dynamics*, Belgium, pp. 51. 1996.

[71] Ohkuma, Toshiaki, Yoichi Suzuki, Yoshitaka Kojima, and Yoshio Tago. "Distributed computing middleware for small and medium-size enterprise." In *First International Conference on e-Science and Grid Computing (e-Science'05)*, pp. 8-pp. IEEE, 2005.

[72] OpenFOAM. "4.3 Mesh generation with the blockMesh utility." *OpenFOAM*, 2023. https://www.openfoam.com/documentation/user-guide/4-mesh-generation-and-conversion/4.3-mesh-generation-with-the-blockmesh-utility.

[73] Fernandes del Pozo, David, Mairtin Mc Namara, Bernardo J. Vitória Pessanha, Peter Baldwin, Jeroen Lauwaert, Joris W. Thybaut, and Ingmar Nopens. "Computational Fluid Dynamics Study of a Pharmaceutical Full-Scale Hydrogenation Reactor." *Processes* 10, no. 6 (2022): 1163. https://doi.org/10.3390/pr10061163

[74] Inatome, Hiroki, and Masato Soga. "Development of an AR Drawing System with Point Cloud Data suitable for Real-time Gripping Movement by using Kinect." *Procedia Computer Science* 126 (2018): 2050-2057. https://doi.org/10.1016/j.procs.2018.07.247

[75] Resolved Analytics. "Navigating the CFD Software Landscape: A Comprehensive Comparison and User Insights." *Resolved Analytics*. Accessed December 15, 2023. https://www.resolvedanalytics.com/theflux/comparing-cfd-software.

[76] Economon, Thomas D., Francisco Palacios, Sean R. Copeland, Trent W. Lukaczyk, and Juan J. Alonso. "SU2: An open-source suite for multiphysics simulation and design." *AIAA Journal* 54, no. 3 (2016): 828-846. https://doi.org/10.2514/1.J053813

[77] Palacios, Francisco, Thomas D. Economon, Aniket Aranake, Sean R. Copeland, Amrita K. Lonkar, Trent W. Lukaczyk, David E. Manosalvas et al. "Stanford university unstructured (SU2): Analysis and design technology for turbulent flows." In *52nd Aerospace Sciences Meeting*, p. 0243. 2014. https://doi.org/10.2514/6.2014-0243

[78] Ricks, Nathan, Panagiotis Tsirikoglou, Francesco Contino, and Ghader Ghorbaniasl. "A CFD-based methodology for aerodynamic-aeroacoustic shape optimization of airfoils." In *AIAA Scitech 2020 Forum*, p. 1729. 2020. https://doi.org/10.2514/6.2020-1729

[79] Economon, Thomas, Francisco Palacios, and Juan Alonso. "A coupled-adjoint method for aerodynamic and aeroacoustic optimization." In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 5598. 2012. https://doi.org/10.2514/6.2012-5598

[80] Archambeau, Frédéric, Namane Méchitoua, and Marc Sakiz. "Code Saturne: A finite volume code for the computation of turbulent incompressible flows-Industrial applications." *International Journal on Finite Volumes* 1, no. 1 (2004).

[81] Turka, A., C. Moulinecb, A. G. Sunderlandb, and C. Aykanata. "Code Saturne Optimizations in Preprocessing." *SMALL* 5 (2012): 335.

[82] Leclercq, Christophe, Antoine Archer, and R. Fortes-Patella. "Numerical investigations on cavitation intensity for 3D homogeneous unsteady viscous flows." In *IOP Conference Series: Earth and Environmental Science*, vol. 49, no. 9, p. 092007. IOP Publishing, 2016. https://doi.org/10.1088/1755-1315/49/9/092007

[83] Keen, Timothy R., Timothy J. Campbell, James D. Dykes, and Paul J. Martin. "Gerris Flow Solver: Implementation and Application." *Defense Technical Information Center* (2013). https://doi.org/10.21236/ADA588626

[84] Popinet, Stéphane. "Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries." *Journal of Computational Physics* 190, no. 2 (2003): 572-600. https://doi.org/10.1016/S0021-9991(03)00298-5

[85] An, Hyunuk, Soonyoung Yu, Giha Lee, and Yeonsu Kim. "Analysis of an open source quadtree grid shallow water flow solver for flood simulation." *Quaternary International* 384 (2015): 118-128. https://doi.org/10.1016/j.quaint.2015.01.032

[86] Lani, Andrea, Nadege Villedie, Khalil Bensassi, Lilla Koloszar, Martin Vymazal, Sarp M. Yalim, and Marco Panesi. "COOLFluiD: an open computational platform for multi-physics simulation and research." In *21st AIAA Computational Fluid Dynamics Conference*, p. 2589. 2013. https://doi.org/10.2514/6.2013-2589

[87] Lani, Andrea, Tiago Quintino, Dries Kimpe, Herman Deconinck, Stefan Vandewalle, and Stefaan Poedts. "Reusable object-oriented solutions for numerical simulation of PDEs in a high performance environment." *Scientific Programming* 14, no. 2 (2006): 111-139. https://doi.org/10.1155/2006/393058

[88] Hecht, Frédéric. "New development in FreeFem++." *Journal of Numerical Mathematics* 20, no. 3-4 (2012): 251-266. https://doi.org/10.1515/jnum-2012-0013

[89] Hecht, Frédéric, Olivier Pironneau, A. Le Hyaric, and K. Ohtsuka. "FreeFem++ manual." *Laboratoire Jacques Louis Lions* (2005).

[90] Font, Roberto, and Francisco Peria. "The Finite Element Method with FreeFem++ for beginners." *Electronic Journal of Mathematics & Technology* 7, no. 4 (2013).

[91] Schrijvers, Patrick Crepier. "ReFresco 2023.1 unveiled." *MARIN Report* 135 (2023).

[92] Angeli, P-E., Ulrich Bieder, and Gauthier Fauchet. "Overview of the TrioCFD code: Main features, VetV procedures and typical applications to nuclear engineering." In *NURETH 16-16th International Topical Meeting on Nuclear Reactor Thermalhydraulics*. 2015.

[93] Panunzio, Domenico, Maria-Adela Puscas, and Romain Lagrange. "FSI-vibrations of immersed cylinders. Simulations with the engineering open-source code TrioCFD. Test cases and experimental comparisons." *Comptes Rendus. Mécanique* 350, no. G3 (2022): 451-476. https://doi.org/10.5802/crmeca.126

[94] Angeli, P-E., M-A. Puscas, G. Fauchet, and A. Cartalade. "FVCA8 benchmark for the Stokes and Navier-Stokes equations with the TrioCFD code-benchmark session." In *Finite Volumes for Complex Applications VIII-Methods and Theoretical Aspects: FVCA 8*, Lille, France, June 2017 8, pp. 181-202. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-57397-7_12

[95] Lani, Andrea, Tiago Quintino, Dries Kimpe, Herman Deconinck, Stefan Vandewalle, and Stefaan Poedts. "The COOLFluiD framework: design solutions for high performance object oriented scientific computing software." In *Computational Science-ICCS 2005: 5th International Conference, Atlanta, GA, USA, May 22-25, 2005. Proceedings, Part I 5*, pp. 279-286. Springer Berlin Heidelberg, 2005. https://doi.org/10.1007/11428831_35

[96] Lehmann, Moritz. "Fluidx3d." *GitHub*. Accessed November 1, 2023. https://github.com/ProjectPhysX/FluidX3D.

[97] Lehmann, Moritz. "Computational study of microplastic transport at the water-air interface with a memory-optimized lattice Boltzmann method." *PhD diss., Universität Bayreuth, Bayreuther Graduiertenschule für Mathematik und Naturwissenschaften-BayNAT*, 2023.

[98] Lehmann, Moritz. "Esoteric pull and esoteric push: Two simple in-place streaming schemes for the lattice Boltzmann method on GPUs." *Computation* 10, no. 6 (2022): 92. https://doi.org/10.3390/computation10060092

[99] Lehmann, Moritz, Lisa Marie Oehlschlägel, Fabian P. Häusl, Andreas Held, and Stephan Gekle. "Ejection of marine microplastics by raindrops: a computational and experimental study." *Microplastics and Nanoplastics* 1 (2021): 1-19. https://doi.org/10.1186/s43591-021-00018-8

[100] Lehmann, Moritz, Mathias J. Krause, Giorgio Amati, Marcello Sega, Jens Harting, and Stephan Gekle. "Accuracy and performance of the lattice Boltzmann method with 64-bit, 32-bit, and customized 16-bit number formats." *Physical Review E* 106, no. 1 (2022): 015308. https://doi.org/10.1103/PhysRevE.106.015308

[101] Lehmann, Moritz. "Combined scientific CFD simulation and interactive raytracing with OpenCL." In *Proceedings of the 10th International Workshop on OpenCL*, pp. 1-2. 2022. https://doi.org/10.1145/3529538.3529542

[102] Alvarez, Eduardo J., Judd Mehr, and Andrew Ning. "FLOWUnsteady: an interactional aerodynamics solver for multirotor aircraft and wind energy." In *AIAA Aviation 2022 Forum*, p. 3218. 2022. https://doi.org/10.2514/6.2022-3218

[103] Alvarez, Eduardo J. "Reformulated vortex particle method and meshless large eddy simulation of multirotor aircraft." *PhD diss., Brigham Young University*, 2022.

[104] Latt, Jonas, Orestis Malaspinas, Dimitrios Kontaxakis, Andrea Parmigiani, Daniel Lagrava, Federico Brogi, Mohamed Ben Belgacem et al. "Palabos: parallel lattice Boltzmann solver." *Computers & Mathematics with Applications* 81 (2021): 334-350. https://doi.org/10.1016/j.camwa.2020.03.022

[105] Krüger, Timm, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. "The lattice Boltzmann method." *Springer International Publishing* 10, no. 978-3 (2017): 4-15. https://doi.org/10.1007/978-3-319-44649-3_1

[106] Cantwell, Chris D., David Moxey, Andrew Comerford, Alessandro Bolis, Gabriele Rocco, Gianmarco Mengaldo, Daniele De Grazia et al. "Nektar++: An open-source spectral/hp element framework." *Computer Physics Communications* 192 (2015): 205-219. https://doi.org/10.1016/j.cpc.2015.02.008

[107] Moxey, David, Chris D. Cantwell, Yan Bao, Andrea Cassinelli, Giacomo Castiglioni, Sehun Chun, Emilia Juda et al. "Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods." *Computer Physics Communications* 249 (2020): 107110. https://doi.org/10.1016/j.cpc.2019.107110

[108] Karniadakis, George, and Spencer J. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, USA, 2005. https://doi.org/10.1093/acprof:oso/9780198528692.001.0001

[109] Scroggs, Matthew W., Jørgen S. Dokken, Chris N. Richardson, and Garth N. Wells. "Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes." *ACM Transactions on Mathematical Software (TOMS)* 48, no. 2 (2022): 1-23. https://doi.org/10.1145/3524456

[110] Scroggs, Matthew W., Igor A. Baratta, Chris N. Richardson, and Garth N. Wells. "Basix: a runtime finite element basis evaluation library." *Journal of Open Source Software* 7, no. 73 (2022): 3982. https://doi.org/10.21105/joss.03982

[111] Alnæs, Martin, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. "The FEniCS project version 1.5." *Archive of Numerical Software* 3, no. 100 (2015).

[112] Logg, Anders, Kent-Andre Mardal, and Garth Wells, eds. *Automated solution of differential equations by the finite element method: The FEniCS book*. Vol. 84. Springer Science & Business Media, 2012. https://doi.org/10.1007/978-3-642-23099-8

[113] Logg, Anders, Garth N. Wells, and Johan Hake. "DOLFIN: A C++/Python finite element library." In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, pp. 173-225. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. https://doi.org/10.1007/978-3-642-23099-8_10

[114] Logg, Anders, and Garth N. Wells. "DOLFIN: Automated finite element computing." *ACM Transactions on Mathematical Software (TOMS)* 37, no. 2 (2010): 1-28. https://doi.org/10.1145/1731022.1731030

[115] Kirby, Robert C., and Anders Logg. "A compiler for variational forms." *ACM Transactions on Mathematical Software (TOMS)* 32, no. 3 (2006): 417-444. https://doi.org/10.1145/1163641.1163644

[116] Ølgaard, Kristian B., and Garth N. Wells. "Optimizations for quadrature representations of finite element tensors through automated code generation." *ACM Transactions on Mathematical Software (TOMS)* 37, no. 1 (2010): 1-23. https://doi.org/10.1145/1644001.1644009

[117] Alnæs, Martin S., Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells. "Unified form language: A domain-specific language for weak formulations of partial differential equations." *ACM Transactions on Mathematical Software (TOMS)* 40, no. 2 (2014): 1-37. https://doi.org/10.1145/2566630

[118] Kirby, Robert C. "Algorithm 839: FIAT, a new paradigm for computing finite element basis functions." *ACM Transactions on Mathematical Software (TOMS)* 30, no. 4 (2004): 502-516. https://doi.org/10.1145/1039813.1039820

[119] Safinowski, Marcin, Maciej Szudarek, Roman Szewczyk, and Wojciech Winiarski. "Capabilities of an open-source software, Elmer FEM, in finite element analysis of fluid flow." In *Recent Advances in Systems, Control and Information Technology: Proceedings of the International Conference SCIT 2016*, May 20-21, 2016, Warsaw, Poland, pp. 118-126. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-48923-0_16

[120] Gagliardini, O., T. Zwinger, F. Gillet-Chaulet, G. Durand, L. Favier, B. De Fleurian, R. Greve et al. "Capabilities and performance of Elmer/Ice, a new-generation ice sheet model." *Geoscientific Model Development* 6, no. 4 (2013): 1299-1318. https://doi.org/10.5194/gmd-6-1299-2013

[121] Gagliardini, O., and T. Zwinger. "The ISMIP-HOM benchmark experiments performed using the Finite-Element code Elmer." *The Cryosphere* 2, no. 1 (2008): 67-76. https://doi.org/10.5194/tc-2-67-2008

[122] Kim, Min Ah, Joong-Youn Lee, Gibeom Gu, Young-Ju Her, Sehoon Lee, Soo Hyung Park, Kyu Hong Kim, and Kumwon Cho. "BARAM: Birtual wind-tunnel system for CFD simulation." *Journal of Computational Fluids Engineering* 20, no. 4 (2015): 28-35. https://doi.org/10.6112/kscfe.2015.20.4.028

[123] NextFoam. "Company." NextFoam. Accessed November 1, 2023. https://www.nextfoam.co.kr/foam-Intro.php.

[124] GNU Operating System. "GNU General Public License." *GNU*. December 25, 2014. https://www.gnu.org/licenses/gpl-3.0.en.html.

[125] University of Liverpool. "Open Source CFD Solver." *University of Liverpool*. Accessed November 1, 2023. https://www.liverpool.ac.uk/flight-science/cfd/cfd_tools/open_source/.

[126] Antoniadis, Antonis F., Dimitris Drikakis, Pericles S. Farmakis, Lin Fu, Ioannis Kokkinakis, Xesús Nogueira, Paulo ASF Silva, Martin Skote, Vladimir Titarev, and Panagiotis Tsoutsanis. "UCNS3D: An open-source high-order finite-volume unstructured CFD solver." *Computer Physics Communications* 279 (2022): 108453. https://doi.org/10.1016/j.cpc.2022.108453

[127] Ducrozet, Guillaume, Félicien Bonnefoy, David Le Touzé, and Pierre Ferrant. "HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method." *Computer Physics Communications* 203 (2016): 245-254. https://doi.org/10.1016/j.cpc.2016.02.017

[128] Choi, YoungMyung, Maite Gouin, Guillaume Ducrozet, Benjamin Bouscasse, and Pierre Ferrant. "Grid2Grid: HOS wrapper program for CFD solvers." *arXiv preprint arXiv:1801.00026* (2017).

[129] Wenkel, Simon. "List of Open-Source CFD Software." *SimonWenkel*. Accessed November 1, 2023. https://www.simonwenkel.com/lists/software/list-of-opensource-CFD-software.html.

[130] Arndt, Daniel, Wolfgang Bangerth, Maximilian Bergbauer, Marco Feder, Marc Fehling, Johannes Heinz, Timo Heister et al. "The deal. II Library, Version 9.5." *Journal of Numerical Mathematics* 31, no. 3 (2023): 1-19. https://doi.org/10.1515/jnma-2023-0089

[131] Arndt, Daniel, Wolfgang Bangerth, Denis Davydov, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, Jean-Paul Pelteret, Bruno Turcksin, and David Wells. "The deal. II finite element library: Design, features, and insights." *Computers & Mathematics with Applications* 81 (2021): 407-422. https://doi.org/10.1016/j.camwa.2020.02.022

[132] Dadvand, Pooyan, Riccardo Rossi, Marisa Gil, Xavier Martorell, Jordi Cotela, Estanislao Juanpere, Sergio R. Idelsohn, and Eugenio Oñate. "Migration of a generic multi-physics framework to HPC environments." *Computers & Fluids* 80 (2013): 301-309. https://doi.org/10.1016/j.compfluid.2012.02.004

[133] Dadvand, Pooyan, Riccardo Rossi, and Eugenio Oñate. "An object-oriented environment for developing finite element codes for multi-disciplinary applications." *Archives of Computational Methods in Engineering* 17 (2010): 253-297. https://doi.org/10.1007/s11831-010-9045-2

[134] Ferrándiz, Vicente Mataix, Philipp Bucher, Rubén Zorrilla, Riccardo Rossi, S. Warnakulasuriya, A. Cornejo, C. Roig et al. "KratosMultiphysics/Kratos: Release 9.2." *GitHub*, 2022. https://github.com/KratosMultiphysics/Kratos.

[135] Minamoto, Yuki. "Flowsquare 4.0: theory and computation." *Flowsquare* (2013).

[136] Minamoto, Yuki. "FlowSquare." *Flowsquare*. Accessed October 27, 2023. http://flowsquare.com.

[137] Karnakov, Petr, and Sergey Litvinov. "Aphros documentation." *Aphros*, 2021. https://doi.org/10.1145/3394277.3401856

[138] Karnakov, Petr, Fabian Wermelinger, Sergey Litvinov, and Petros Koumoutsakos. "Aphros: High Performance Software for Multiphase Flows with Large Scale Bubble and Drop Clusters." In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pp. 10. 2020. https://doi.org/10.1145/3394277.3401856

[139] Espinoza, D. E. R., Vincent Casseau, T. J. Scanlon, and R. E. Brown. "An open-source hybrid CFD-DSMC solver for high speed flows." In *AIP Conference Proceedings*, vol. 1786, no. 1. AIP Publishing, 2016. https://doi.org/10.1063/1.4967557

[140] HyStrath. "hyStrath." *GitHub*. Accessed November 1, 2023. https://github.com/hystrath/hyStrath/.

[141] Wagner, Gregory L., and Navid C. Constantinou. "Fourierflows v0.10.4." *GitHub*, 2023. https://github.com/FourierFlows/FourierFlows.jl.

[142] Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B. Shah. "Julia: A fresh approach to numerical computing." *SIAM Review* 59, no. 1 (2017): 65-98. https://doi.org/10.1137/141000671

[143] NVIDIA. "Nvidia flow SDK manual." *NVIDIA*. Accessed November 1, 2023. https://developer.nvidia.com/nvidiaflow.

[144] NVIDIA. "Nvidia gameworks flow documentation." *NVIDIA*. Accessed November 1, 2023. https://docs.nvidia.com/gameworks/content/artisttools/Flow/FLOWUe4Intro.html.FlowrequiresaDX.

[145] Urban, Cameron. "Ptera software: A fast, easy-to-use, and open-source package for analyzing flapping-wing flight." *GitHub*. November 1, 2023. https://github.com/camUrban/PteraSoftware.

[146] Urban, Cameron, and Ramesh K. Agarwal. "Validation and Optimization of Ptera Software: An Open-Source Unsteady Flow Simulator for Flapping Wings." In *AIAA SCITECH 2022 Forum*, p. 1967. 2022. https://doi.org/10.2514/6.2022-1967

[147] Domino, Stefan. "Sierra low mach module: Nalu theory manual 1.0." *SAND Report SAND2015-3107W, Sandia National Laboratories Unclassified Unlimited Release (UUR)*, 2015.

[148] Coffey, Todd, Alan Williams, Manoj Bhardwaj, David Galze, Tolulope Okusanya, Nathaniel Roehrig, Christopher Wilson, Nathan Crane, and Patrick Xavier. "SIERRA Toolkit v. 2.0." *U.S. Department of Energy, Office of Scientific and Technical Information*. September 14, 2016. https://www.osti.gov/biblio/1326625.

[149] Heroux, Michael A., Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq et al. "An overview of the Trilinos project." *ACM Transactions on Mathematical Software (TOMS)* 31, no. 3 (2005): 397-423. https://doi.org/10.1145/1089014.1089021

[150] Bonamy, Cyrille, Julien Chauchat, Pierre Augier, Antoine Mathieu, Quentin Clemencot, R'emi Chassagne, Guillaume Maurice, Alban Gilletta, Matthias Renaud, and Gabriel Gonçalves. "FluidFoam Release v0.2.3." *GitHub*. April 12, 2022. https://github.com/fluiddyn/fluidfoam/releases/tag/v0.2.3.

[151] Laizet, Sylvain, and Ning Li. "Incompact3d: A powerful tool to tackle turbulence problems with up to O (105) computational cores." *International Journal for Numerical Methods in Fluids* 67, no. 11 (2011): 1735-1757. https://doi.org/10.1002/fld.2480

[152] Metcalf, Michael, and John K. Reid. *Fortran 90/95 explained*. Oxford University Press, Inc., 1999.

[153] Jones, C. A., P. Boronski, A. S. Brun, G. A. Glatzmaier, T. Gastine, M. S. Miesch, and J. Wicht. "Anelastic convection-driven dynamo benchmarks." *Icarus* 216, no. 1 (2011): 120-135. https://doi.org/10.1016/j.icarus.2011.08.014

[154] Squillacote, Amy Henderson, James Ahrens, Charles Law, Berk Geveci, Kenneth Moreland, and Brad King. *The paraview guide*. Vol. 366. Clifton Park, NY: Kitware, 2007.

[155] Ayachit, Utkarsh. *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.

[156] Schroeder, Will, Kenneth M. Martin, and William E. Lorensen. *The visualization toolkit an object-oriented approach to 3D graphics*. Prentice-Hall, Inc., 1998.

[157] Langtangen, Hans Petter, ed. *Python scripting for computational science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. https://doi.org/10.1007/978-3-540-73916-6

[158] Childs, Hank, Eric Brugger, Kathleen Bonnell, Jeremy Meredith, Mark Miller, Brad Whitlock, and Nelson Max. *A contract based system for large data visualization*. IEEE Visualization, 2005.

[159] Kokko, Edwin J., Harry E. Martz Jr, Diane J. Chinn, Henry R. Childs, Jessie A. Jackson, David H. Chambers, Daniel J. Schneberk, and Grace A. Clark. "As-built modeling of objects for performance assessment." *Journal of Computing and Information Science in Engineering* 6, no. 4 (2006): 405-417. https://doi.org/10.1115/1.2353856

[160] Childs, Hank, Mark Duchaineau, and Kwan-Liu Ma. "A scalable, hybrid scheme for volume rendering massive data sets." In *Proceedings of the 6th Eurographics conference on Parallel Graphics and Visualization*, pp. 153-161. 2006.

[161] Childs, Hank R. *An Analysis Framework Addressing the Scale and Legibility of Large Scientific Data Sets*. No. UCRL-TH-226455. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2006. https://doi.org/10.2172/900438

[162] Lichtenbelt, Barthold, Randy Crane, and Shaz Naqvi. *Introduction to volume rendering*. Prentice-Hall, Inc., 1998.

[163] Folk, Mike, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. "An overview of the HDF5 technology suite and its applications." In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, pp. 36-47. 2011. https://doi.org/10.1145/1966895.1966900

[164] Rew, Russ, Glenn Davis, Steve Emmerson, Harvey Davies, and Ed Hartnett. "NetCDF user's Guide." *University Corporation for Atmospheric Research* (1993).

[165] Rew, Russ, and Glenn Davis. "NetCDF: an interface for scientific data access." *IEEE Computer Graphics and Applications* 10, no. 4 (1990): 76-82. https://doi.org/10.1109/38.56302

[166] Heber, Gerd, Chris Pelkie, Andrew Dolgert, Jim Gray, and David Thompson. *Supporting Finite Element Analysis with a Relational Database Backend; Part III: OpenDX-Where the Numbers Come Alive*. Technical Report MSR-TR-2005-151, 2005.

[167] Ramachandran, Prabhu, and Gaël Varoquaux. "Mayavi: 3D visualization of scientific data." *Computing in Science & Engineering* 13, no. 2 (2011): 40-51. https://doi.org/10.1109/MCSE.2011.35

[168] Varoquaux, Gaël, and Prabhu Ramachandran. "Mayavi: Making 3D data visualization reusable." In *SciPy 2008: 7th Python in Science Conference*, vol. 1, no. 1, p. 51. 2008.

[169] Ruiz-Rodríguez, Manuel Antonio, Christopher D. Cooper, Walter Rocchia, Mosè Casalegno, Yossef López De Los Santos, and Guido Raos. "Modeling of the electrostatic interaction and catalytic activity of [NiFe] hydrogenases on a planar electrode." *The Journal of Physical Chemistry B* 126, no. 43 (2022): 8777-8790. https://doi.org/10.1021/acs.jpcb.2c05371

[170] VCell. "VisIt Visualization Tool." *VCell*. Accessed October 15, 2023. https://vcell.org/webstart/VCell_Tutorials/VCell_Help/topics/ch_1/Introduction/LaunchVisITvisualization.html.

[171] Ramachandran, Prabhu, and Gaël Varoquaux. "Mayavi2: 3D Scientific Data Visualization and Plottin'." *Astrophysics Source Code Library* (2012): ascl-1205.

[172] Williams, Thomas, Colin Kelley, Christoph Bersch, Hans-Bernhard Bröker, John Campbell, Robert Cunningham, David Denholm et al. "gnuplot." *An Interactive Plotting Program* (2004).

[173] Janert, Philipp K. *Gnuplot in action: understanding data with graphs*. Simon and Schuster, 2016.

[174] Dorney, Daniel, Lisa Griffin, Bogdan Marcu, and Morgan Williams. "Unsteady flow interactions between the LH2 feed line and SSME LPFP inducer." In *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, p. 5073. 2006. https://doi.org/10.2514/6.2006-5073

[175] bjaraujo. "ENigMA - Extended Numerical Multiphysics Analysis." *GitHub*. Accessed November 1, 2023. https://github.com/bjaraujo/ENigMA.

[176] Navas González, Rubén. "Study for the computational resolution of conservation equations of mass, momentum and energy." *Bachelor's thesis, Universitat Politècnica de Catalunya*, 2022.

[177] SimScale. "High-fidelity engineering simulation on any hardware." *SimScale*. Accessed November 10, 2023. https://www.simscale.com/.

[178] Benoit, Christophe, Stéphanie Péron, and Sâm Landier. "Cassiopee: a CFD pre-and post-processing tool." *Aerospace Science and Technology* 45 (2015): 272-283. https://doi.org/10.1016/j.ast.2015.05.023

[179] Eaton, John Wesley, David Bateman, and Søren Hauberg. *Gnu octave*. London: Network theory, 2013.

[180] Eaton, John W. "GNU Octave and reproducible research." *Journal of Process Control* 22, no. 8 (2012): 1433-1438. https://doi.org/10.1016/j.jprocont.2012.04.006

[181] Liu, Wei, Song Lian, Xin Fang, Zhenyu Shang, Hao Wu, Hao Zhu, and Simon Hu. "An open-source and experimentally guided CFD strategy for predicting air distribution in data centers with air-cooling." *Building and Environment* 242 (2023): 110542. https://doi.org/10.1016/j.buildenv.2023.110542

[182] Paritala, Phani Kumari, Haveena Anbananthan, Jacob Hautaniemi, Macauley Smith, Antony George, Mark Allenby, Jessica Benitez Mendieta et al. "Reproducibility of the computational fluid dynamic analysis of a cerebral aneurysm monitored over a decade." *Scientific Reports* 13, no. 1 (2023): 219. https://doi.org/10.1038/s41598-022-27354-w

[183] Tripathi, Manish, Pranshul Pandey, and Rajkumar S. Pant. "Drag mitigation of trilobed airship hull through aerodynamic comparison with conventional single-lobed hull." *Journal of Aerospace Engineering* 36, no. 6 (2023): 04023073. https://doi.org/10.1061/JAEEEZ.ASENG-4793

[184] Singh, Balbir, Usman Ikhtiar, Mohamad Firzan, Dong Huizhen, and Kamarul Arifin Ahmad. "Numerical Analysis of a Mobile Leakage-Detection System for a Water Pipeline Network." *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences* 87, no. 1 (2021): 134-150. https://doi.org/10.37934/arfmts.87.1.134150

[185] Kumar, Yasasvi Harish, Utkarsh Tripathi, and Balbir Singh. "Design, mathematical modeling, and stability of a reconfigurable multirotor aerial vehicle." *Journal of Aerospace Engineering* 33, no. 2 (2020): 04019112. https://doi.org/10.1061/(ASCE)AS.1943-5525.0001102

[186] Singh, Balbir, Noorfaizal Yidris, Adi Azriff Basri, Raghuvir Pai, and Kamarul Arifin Ahmad. "Study of mosquito aerodynamics for imitation as a small robot and flight in a low-density environment." *Micromachines* 12, no. 5 (2021): 511. https://doi.org/10.3390/mi12050511

[187] Singh, Balbir, Adi Azriff basri, Noorfaizal Yidris, Raghuvir Pai, and Kamarul Arifin Ahmad. "Unsteady Flow Topology Around an Insect-Inspired Flapping Wing Pico Aerial Vehicle." In *High Performance Computing in Biomimetics: Modeling, Architecture and Applications*, pp. 241-250. Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-97-1017-1_11

[188] Abas, Mohd Firdaus Bin, Balbir Singh, Kamarul Arifin Ahmad, Eddie Yin Kwee Ng, Tabrej Khan, and Tamer A. Sebaey. "Dwarf kingfisher-inspired bionic flapping wing and its aerodynamic performance at lowest flight speed." *Biomimetics* 7, no. 3 (2022): 123. https://doi.org/10.3390/biomimetics7030123

[189] Murugaiah, Manikandan. "Application of Machine Learning and Deep Learning in High Performance Computing." In *High Performance Computing in Biomimetics: Modeling, Architecture and Applications*, pp. 271-286. Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-97-1017-1_14

[190] Singh, Balbir, Adi Azriff basri, Noorfaizal Yidris, Raghuvir Pai, and Kamarul Arifin Ahmad. "Machine Learning Based Dynamic Mode Decomposition of Vector Flow Field Around Mosquito-Inspired Flapping Wing." In *High Performance Computing in Biomimetics: Modeling, Architecture and Applications*, pp. 251-258. Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-97-1017-1_12

[191] Piscaglia, Federico, and Federico Ghioldi. "GPU acceleration of CFD simulations in OpenFOAM." *Aerospace* 10, no. 9 (2023): 792. https://doi.org/10.3390/aerospace10090792

[192] Roja, P., T. Sankar Reddy, S. M. Ibrahim, Giulio Lorenzini, and Nor Azwadi Che Sidik. "The Effect of thermophoresis on MHD stream of a micropolar liquid through a porous medium with variable heat and mass flux and thermal radiation." *CFD Letters* 14, no. 4 (2022): 118-136. https://doi.org/10.37934/cfdl.14.4.118136

[193] Arifin, Mohammad Danil, Frengki Mohamad Felayati, and Andi Haris Muhammad. "Flow separation evaluation on tubercle ship propeller." *CFD Letters* 14, no. 4 (2022): 43-50. https://doi.org/10.37934/cfdl.14.4.4350