FOSSEE, IIT Bombay
OpenFOAM Project
December 1, 2025

CFD Workbench Evaluation and Testing using OpenFOAM FreeCAD

Tithi Biswas
B.Tech - Computer Science(AI
ML)
Kalinga Institute of Industrial Technology ,Bhubhaneshwar

Abstract

This is a comparative review of the two most well-known open-source workflows in CFD, between the integrated FreeCAD CfdOF workbench and the native OpenFOAM environment. It was performed mainly to measure the reliability, user accessibility, and effectiveness of the procedures for both tools within an engineering design pipeline. The work methodology in the current report was performed by parallel realization of the simulation of fluid flow: the first one with the use of a GUI-driven interface of the CfdOF workbench, allowing the management of geometry preparation, meshing, and solver configuration, and the second approach employing the traditional script-based workflow in OpenFOAM, which uses command-line utilities for mesh generation and case management. Both workflows were analyzed based on their ease of setup, the flexibility of mesh control, and the consistency of the resulting flow fields. A comparison would show that while both use the same solvers, each offers distinct advantages, depending on the needs of the user. On the one hand, the entry barrier for CfdOF workbench in CFD is much lower, with the pre-processing phase being much smoother than compared to native OpenFOAM, where granular control is needed for complex or non-standard simulations. The importance of the practical application of these open-source tools and the ability to achieve realistic results through the combination of FreeCAD and OpenFOAM demonstrates the success and the accomplishment of the aim to make advanced tools more accessible to the technical community.

Acknowledgement

I would like to express my sincere gratitude to the FOSSEE Project at IIT Bombay for providing me with this exceptional opportunity to contribute to the development and evaluation of open-source CFD software. This internship has greatly broadened my understanding of the importance and impact of community-driven tools in the field of engineering.

I am deeply thankful to my mentors, Diptangshu Dey and Vedant Dubey, for their constant guidance, encouragement, and the well-structured environment they created, which helped me navigate the complexities of this project. Their expertise in project organization and execution played a vital role in the successful completion of this work.

I would also like to extend my heartfelt thanks to Mr. Parees Palkar for his valuable training and support with the FreeCAD CfdOF workbench. His insights were crucial in bridging the gap between geometric modeling and simulation setup.

I am equally grateful to Mr. Evan Fernandes for his specialized training and guidance in OpenFOAM. His in-depth knowledge of solver configurations and command-line workflows formed the foundation of the comparative analysis presented in this report.

Finally, I would like to acknowledge the collaborative and supportive environment fostered by the entire FOSSEE team. Their collective efforts have been instrumental in the progress of this project. This experience has significantly strengthened my technical skills in CFD and reinforced my commitment to advancing open-source engineering solutions for the global community.

Contents

List of Figures

1 Introduction

Computational Fluid Dynamics (CFD) enables the numerical analysis of fluid flow, heat transfer, and related physical phenomena by solving governing equations on discrete computational domains. With the increasing availability of open-source tools, CFD has become accessible to a broader academic and research community.

OpenFOAM is one of the most powerful open-source CFD platforms, offering a wide range of solvers for incompressible, compressible, laminar, and turbulent flows. However, its reliance on command-line execution and manual dictionary editing poses challenges for new users. To overcome this limitation, graphical interfaces such as the FreeCAD CfdOF workbench have been developed.

This internship focused on evaluating the effectiveness of the CfdOF workbench for setting up and executing CFD simulations using OpenFOAM, with emphasis on benchmark flow problems.

2 Computational Fluid Dynamics Background

2.1 Governing Equations

CFD simulations are governed by the Navier–Stokes equations, which represent conservation of mass and momentum:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} \tag{2}$$

where $\mathbf{u}$ is velocity, p is pressure, ρ is density, and μ is viscosity.

2.2 CFD Workflow

A standard CFD workflow consists of:

- Pre-processing (geometry creation and meshing)
- Solving (numerical computation)
- Post-processing (visualization and analysis)

3 Overview of OpenFOAM

OpenFOAM (Open Field Operation and Manipulation) is an open-source Computational Fluid Dynamics (CFD) toolbox primarily developed in C++ and widely used for solving continuum mechanics problems such as fluid flow, turbulence modelling, heat transfer, multiphase flow, and compressible aerodynamics. Unlike commercial CFD packages that rely heavily on closed GUIs, OpenFOAM follows an open architecture in which simulation setup is performed through editable dictionary files. This design provides exceptional flexibility and makes OpenFOAM highly suitable for academic research, solver customization, and large-scale simulations.

3.1 Core Philosophy and Workflow

OpenFOAM operates on the concept of a *case-based workflow*. Each CFD problem is represented as a case folder containing all the settings, mesh information, initial conditions and solver controls. The simulation workflow generally follows three major stages:

1. **Pre-processing:** geometry definition, mesh generation, boundary and initial conditions.
2. **Solving:** numerical computation using appropriate solver (steady/transient, laminar/turbulent, incompressible/compressible).
3. **Post-processing:** visualization and data extraction, typically using ParaView.

This modular design allows users to update only the required settings without changing the full simulation workflow. The pre-processing stage can use utilities such as `blockMesh`, `snappyHexMesh` and `checkMesh`, while the solving stage employs solvers like `icoFoam`, `simpleFoam`, `pisoFoam`, and `pimpleFoam`, depending on flow physics.

3.2 Case Directory Structure

A standard OpenFOAM case directory contains three essential folders:

- **`0/`**: Stores the initial and boundary conditions for the flow variables such as velocity (`U`), pressure (`p`), turbulence quantities (`k, epsilon`), temperature (`T`), etc.
- **`constant/`**: Contains the computational mesh in `polyMesh/` as well as physical properties like transport coefficients and turbulence models.
- **`system/`**: Contains control dictionaries that govern solver execution and discretization.

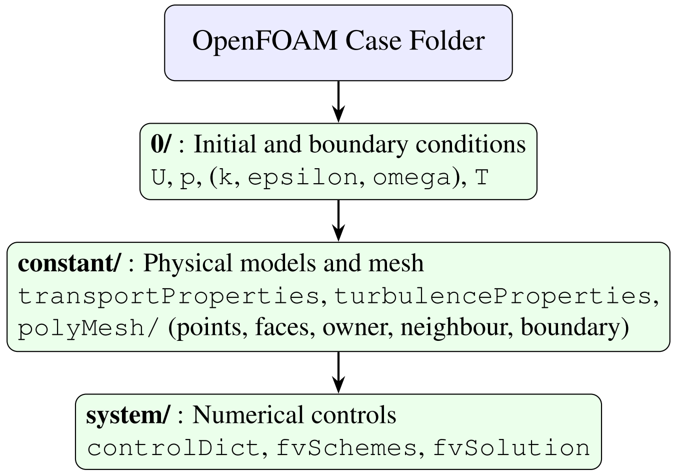

Figure 1: Standard OpenFOAM case directory structure and role of key dictionaries.

3.3 System Dictionaries and Their Role

The `system/` directory is a critical part of OpenFOAM as it controls the numerical accuracy, convergence, and solver behavior:

- **`controlDict`**: Controls simulation time, write intervals, and run-time control.
- **`fvSchemes`**: Defines discretization schemes for gradient, divergence and laplacian terms.
- **`fvSolution`**: Contains linear solver settings, tolerances, relaxation factors, and coupling algorithms (SIMPLE, PISO, PIMPLE).

Since these files are text-based, OpenFOAM provides full transparency and reproducibility, but minor syntax errors (missing semicolons, incorrect patch names) can cause simulation failures—making the workflow challenging for beginners.

3.4 Terminal-Based Workflow and Case Execution

The conventional mode of running OpenFOAM cases is through the Command Line Interface (CLI). A typical execution procedure includes:

1. Sourcing the OpenFOAM environment (`source /opt/openfoam*/etc/bashrc`)
2. Navigating to the case directory
3. Running mesh utilities (e.g., `blockMesh`)
4. Checking mesh quality (`checkMesh`)
5. Running solver (e.g., `icoFoam` or `simpleFoam`)
6. Visualizing results (`paraFoam`)

The CLI workflow enables strong automation using bash scripts and Python, which is essential for parametric studies and high-performance computing (HPC) executions. However, the workflow demands both CFD knowledge and Linux command proficiency. .

4 FreeCAD and CfdOF Workbench

FreeCAD is a modern open-source parametric CAD software widely used for engineering design and modelling. It supports modular workbenches such as Part Design, Sketcher, Draft, FEM, and TechDraw, making it capable of both design and analysis workflows. Due to its parametric nature, modifications to the CAD model automatically update the geometry and related design constraints, which is highly useful in iterative engineering workflows.

4.1 FreeCAD Architecture and Engineering Workflow

FreeCAD is built on the OpenCASCADE geometry kernel and provides a feature-based modelling approach, similar to commercial CAD environments. A standard workflow involves:

1. Sketching and constraining 2D geometry
2. Creating 3D features such as pads, pockets, and revolutions
3. Refining geometry using fillets, chamfers and boolean operations
4. Exporting or using the model for engineering analysis

This architecture makes FreeCAD an effective pre-processing environment for CFD, where domain creation and geometry cleanup are essential prior to simulation setup.

FreeCAD CAD Model(Parametric Geometry)

↓

CfdOF Analysis Container(Solver, Mesh, Physics)

↓

Mesh Generation(GUI Meshing + Refinement)

↓

Boundary Conditions(Face-based Assignment)

↓

Automatic Case Writing(`0/`, `constant/`, `system/`)

↓

OpenFOAM Solver Execution(`icoFoam`/`simpleFoam`)

↓

Post-processing(ParaView / Residual Monitoring)

Figure 2: Workflow architecture showing integration of FreeCAD CfdOF workbench with OpenFOAM.

4.2 CfdOF Workbench: Purpose and Role

The CfdOF (CFD-OpenFOAM) workbench is a specialized FreeCAD add-on designed to integrate OpenFOAM CFD simulations within a graphical user interface. It does not replace OpenFOAM solvers; instead, it acts as a bridge between CAD modelling and OpenFOAM's case-based simulation framework. Through this integration, users can perform geometry preparation, meshing, boundary condition assignment, case writing and solver execution directly inside FreeCAD.

4.3 Analysis Container Concept

A central feature of the CfdOF workbench is the **CFD Analysis Container**. This object organizes all simulation parameters within the FreeCAD project tree. The analysis container stores:

- Solver selection and run control
- Mesh type and refinement parameters
- Material properties (e.g., viscosity, density)
- Boundary condition assignments linked to geometry faces

This approach provides a structured workflow and reduces manual editing of case dictionaries.

Additionally, because the analysis container is integrated into the FreeCAD tree view, changes in geometry can be systematically propagated through mesh and simulation setup, making the workflow partially non-destructive and parametric.

4.4 Operational Workflow in CfdOF

The CfdOF workbench follows a guided CFD workflow:

1. **Geometry Selection:** User selects the solid/fluid domain from the CAD model.
2. **Mesh Generation:** Mesh parameters are set via GUI and meshing tools are executed.
3. **Boundary Assignment:** Faces are selected in 3D view and assigned as inlet, outlet, walls etc.
4. **Solver Configuration:** Solver type (steady/transient) and turbulence models are selected.
5. **Case Writing:** The workbench auto-generates the OpenFOAM case directory.

5 Methodology

This internship followed a structured CFD workflow to evaluate and test OpenFOAM simulations through the FreeCAD CfdOF workbench. The overall methodology was designed to systematically study the complete CFD pipeline starting from geometry modelling up to post-processing and workflow comparison. The two benchmark cases (lid-driven cavity flow and 2D channel flow) were treated as validation problems to understand mesh quality, solver stability, convergence monitoring, and visualization of flow features.

5.1 Workflow Overview

The workflow adopted during the internship consisted of three standard CFD phases:

1. **Pre-processing:** Geometry creation and meshing
2. **Solving:** Numerical simulation using OpenFOAM solvers
3. **Post-processing:** Visualization and extraction of results

This classical CFD workflow was implemented using two approaches: (i) GUI-based workflow via FreeCAD CfdOF workbench, and (ii) analysis of the generated case structure in native OpenFOAM format.

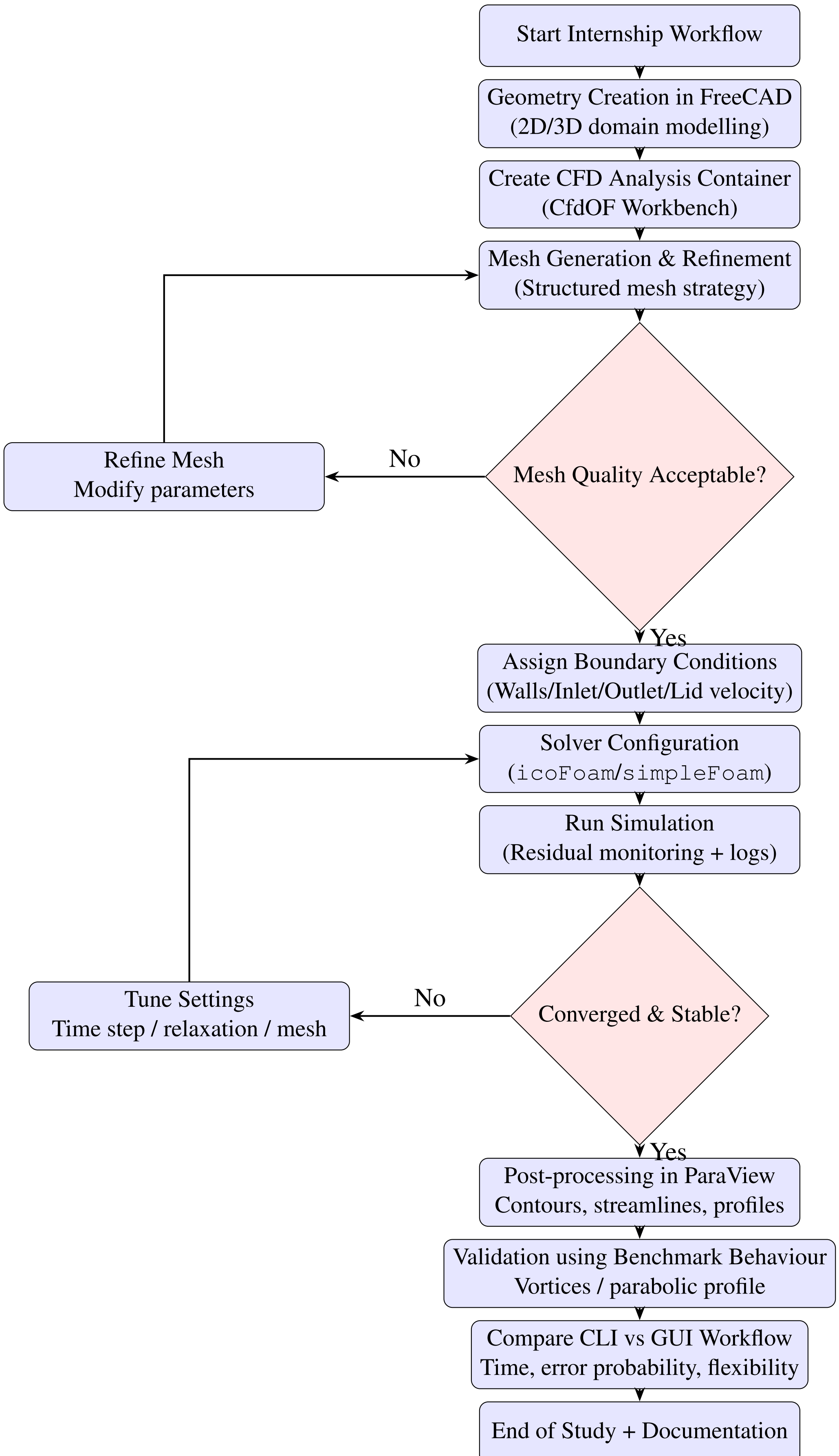

Figure 3: Complete methodology flowchart followed during the internship for CFD workbench evaluation.

5.2 Software Tools and Environment

The following tools were used throughout the internship:

- **FreeCAD:** Geometry modelling and parametric domain preparation
- **CfdOF Workbench:** CFD setup interface for OpenFOAM (meshing, BC assignment, solver selection)
- **OpenFOAM:** Solver execution and numerical computation
- **ParaView:** Flow field visualization and contour plotting

5.3 Geometry Creation and Domain Preparation

For each case study, a clean computational domain was prepared in FreeCAD. The domain creation process involved:

- Selecting an appropriate 2D geometry representation for incompressible benchmark problems
- Defining boundary faces clearly for inlet/outlet/walls/moving wall conditions
- Ensuring geometry validity by avoiding gaps or overlapping edges

This stage was critical because accurate geometry definition ensures that mesh patches and boundary conditions correspond correctly to physical regions in the CFD problem.

5.4 CFD Analysis Setup using the CfdOF Workbench

A new CFD analysis container was created for each problem using the CfdOF workflow. The analysis container served as the central object storing:

- Solver parameters and run controls
- Meshing setup and refinement levels
- Material properties (viscosity, density models)
- Boundary condition definitions linked to CAD faces

This approach made the workflow non-fragmented, since the complete setup remained embedded inside the FreeCAD project tree.

5.5 Mesh Generation Strategy and Quality Verification

Meshing was performed using structured grid generation suitable for the benchmark cases. Mesh generation involved:

- Choosing cell size based on expected velocity gradients and vortex formation regions
- Applying refinement near walls to resolve boundary-layer behaviour

- Ensuring consistent mesh transitions to reduce numerical instability

After meshing, mesh verification was performed. Mesh quality checks are essential in CFD because issues such as high non-orthogonality or skewness can lead to divergence or inaccurate results. The following mesh quality criteria were monitored:

- Non-orthogonality
- Skewness
- Aspect ratio
- Cell volume and face quality

In OpenFOAM, this is typically verified using `checkMesh`, while in CfdOF the quality is also viewable through the GUI-based mesh inspection tools.

5.6 Boundary Condition Assignment

Boundary conditions were assigned based on physics and benchmark definitions:

- **Lid-driven cavity:** moving wall velocity at top; no-slip walls on other boundaries
- **2D channel:** fixed inlet velocity, pressure outlet, and no-slip wall boundaries

A key advantage of the CfdOF workbench is that boundary conditions are assigned directly by selecting faces of the geometry, reducing errors due to incorrect patch naming or misidentification.

5.7 Solver Selection and Numerical Configuration

Solver selection was performed according to the nature of each flow:

- **Lid-driven cavity:** `icoFoam` (transient, incompressible, laminar solver)
- **2D channel flow:** `simpleFoam` (steady-state, incompressible solver)

Numerical configuration includes discretization schemes and solver tolerances. In OpenFOAM, these parameters are defined in:

- `fvSchemes` (spatial discretization)
- `fvSolution` (linear solvers and relaxation factors)
- `controlDict` (time settings and write intervals)

CfdOF automatically generates these dictionaries based on GUI selections, ensuring syntactic correctness and standard solver stability.

5.8 Simulation Execution and Convergence Monitoring

After case writing, simulations were executed using the inbuilt solver execution option in CfdOF. Solver progress was monitored through:

- Residual plots for velocity and pressure fields
- Stability of field variables over iterations/time steps
- Observation of convergence patterns and oscillations

Convergence quality was assessed using both residual decay and physical correctness of flow patterns, as residuals alone may not always guarantee a valid physical solution.

5.9 Post-processing and Result Validation

Post-processing was performed using ParaView (via `paraFoam`). The following visualizations were generated:

- Velocity magnitude contours
- Pressure distribution plots
- Streamlines and vortex structures
- Velocity profile extraction (channel case)

Validation was performed using expected benchmark behaviours:

- Presence of primary and secondary vortices in cavity flow
- Parabolic laminar velocity profile in channel flow

5.10 Workflow Comparison and Evaluation Metrics

Finally, the two workflows were compared based on:

- Setup time and complexity
- Error probability in boundary condition and mesh assignment
- Ability to modify geometry and update simulations
- Flexibility for advanced numerical tuning
- Scalability and automation potential

This comparative analysis helped quantify the strengths and limitations of each approach, showing how GUI-based CFD (CfdOF) simplifies case setup, while CLI-based OpenFOAM retains maximum flexibility and control for research-level simulations.

6 Case Study 1: Lid-Driven Cavity Flow

6.1 Problem Description and Significance

The lid-driven cavity flow is one of the most widely used benchmark problems in Computational Fluid Dynamics (CFD). The test consists of a square cavity filled with an incompressible Newtonian fluid, where the top wall (lid) moves with a constant tangential velocity while the remaining walls remain stationary. Despite its simple geometry, the flow exhibits complex vortex formation and recirculation patterns, making it ideal for validating solver accuracy, discretization schemes, and boundary condition implementation.

This case was assigned during the internship to strengthen the understanding of incompressible flow behaviour, vorticity formation, and numerical stability in CFD simulations.

6.2 Geometry Creation

The geometry was created in FreeCAD as a two-dimensional square cavity domain. The cavity was designed with clean edges to ensure high-quality meshing. The domain size was chosen as a unit square for simplicity and better comparison with standard benchmark literature.

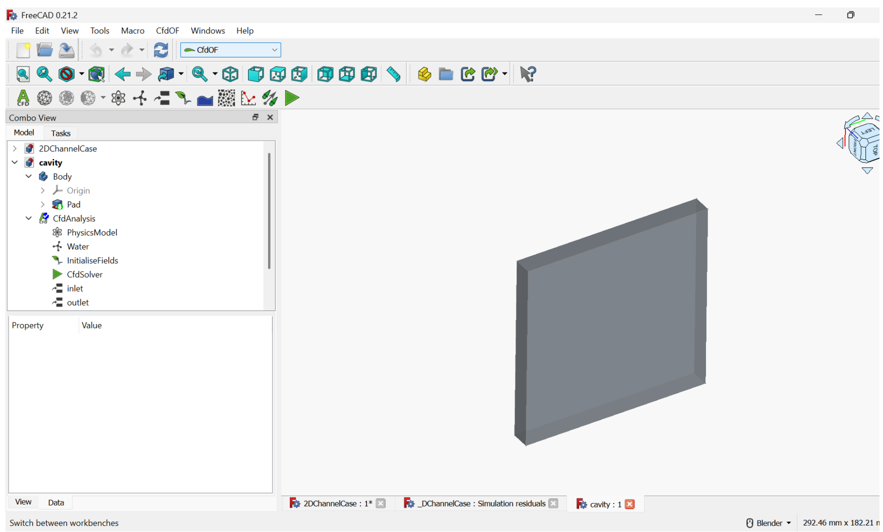

Figure 4: Lid-driven cavity geometry created in FreeCAD.

6.3 Mesh Generation and Quality

A structured mesh was generated using the meshing utilities available through the CfdOF workbench. The mesh resolution was selected to capture vortex patterns effectively while maintaining computational efficiency. Mesh refinement was applied uniformly across the cavity domain to minimize numerical diffusion and to resolve boundary-layer gradients near the walls.

Mesh quality was verified using `checkMesh`, ensuring that non-orthogonality, skewness, and aspect ratio remained within acceptable solver limits.

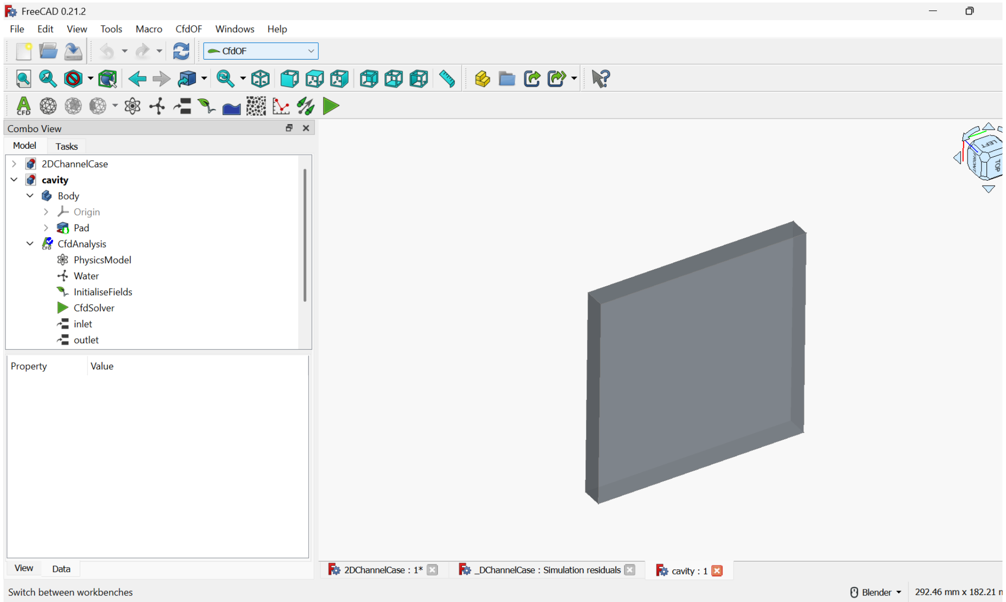

Figure 5: Structured mesh of the lid-driven cavity flow case.

6.4 Boundary Conditions

The following boundary conditions were applied:

- **Moving Lid (Top wall):** Fixed tangential velocity in x-direction ($U = U_{lid}$), zero normal velocity.
- **Stationary Walls (Side and Bottom walls):** No-slip condition ($U = 0$).
- **Pressure Field:** Zero-gradient pressure on walls to maintain incompressibility constraints.

6.5 Solver Setup

The simulation was performed using the `icoFoam` solver for transient incompressible laminar flow. Numerical settings such as time step size and discretization schemes were chosen to maintain numerical stability and convergence.

6.6 Simulation Execution and Convergence Monitoring

The simulation was executed directly from the FreeCAD CfdOF interface after case writing. Residuals for pressure and velocity were monitored to ensure stable convergence behaviour. The convergence trend indicated smooth decay of residuals, confirming that the solution evolved towards a physically steady vortex structure.

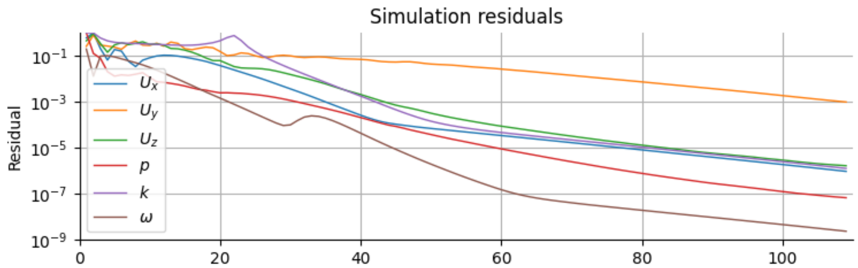

Figure 6: Residual plot for lid-driven cavity simulation.

6.7 Results and Discussion

The simulation results were visualized using ParaView. Velocity contours demonstrated the formation of a primary vortex in the cavity centre along with secondary vortices near the bottom corners. These vortex structures are consistent with benchmark results reported in CFD literature, validating the correctness of solver settings and boundary conditions.

The lid-driven cavity case helped in understanding how mesh density and discretization impact recirculating flows. It also served as a foundation for learning how OpenFOAM handles incompressibility constraints via pressure-velocity coupling. .

7 Case Study 2: Two-Dimensional Channel Flow

7.1 Problem Description and Significance

The two-dimensional channel flow case represents incompressible viscous flow between two parallel plates. It is a fundamental validation case used in CFD to study flow development, wall shear behaviour, and velocity profile formation. Under laminar conditions, the fully developed velocity profile becomes parabolic, consistent with the analytical Hagen–Poiseuille flow solution.

This case was assigned to understand steady-state solver behaviour, inlet/outlet boundary condition effects, and wall-bounded flow modelling.

7.2 Geometry Definition

A rectangular channel geometry was created in FreeCAD with sufficient length-to-height ratio to allow velocity profile development. The domain was defined such that flow enters at the inlet, develops along the channel, and exits at the outlet.

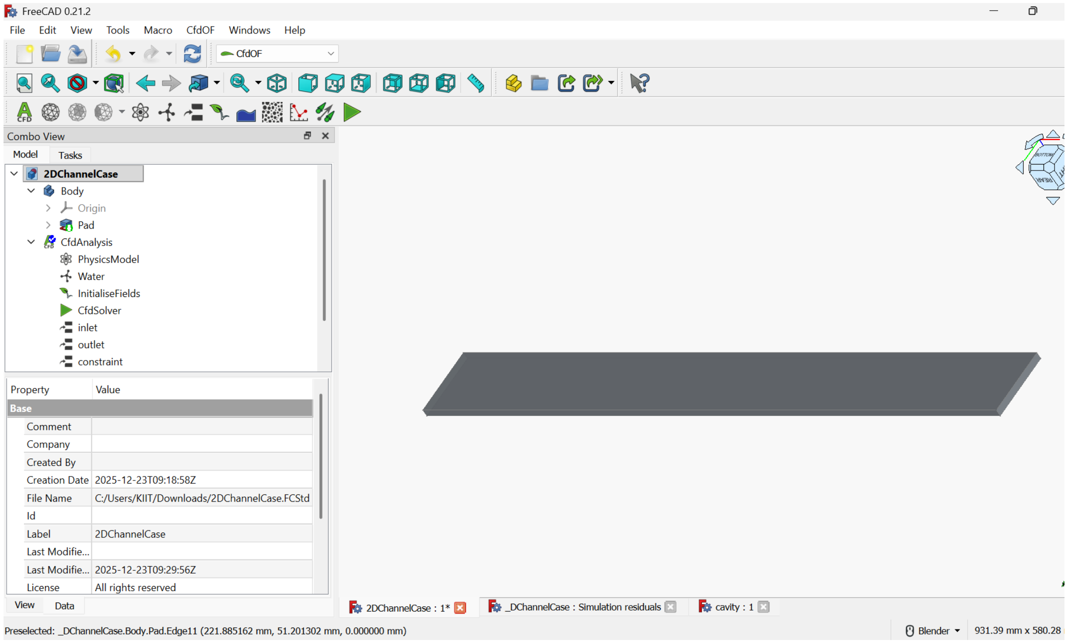

Figure 7: 2D channel geometry created in FreeCAD.

7.3 Mesh Generation Strategy

A structured mesh was generated for the channel domain. To accurately capture wall shear and velocity gradients, finer mesh spacing was applied near the channel walls. This is important because boundary layers form near walls and require sufficient resolution for accurate CFD predictions.

Mesh quality checks were performed to ensure stable solver execution.

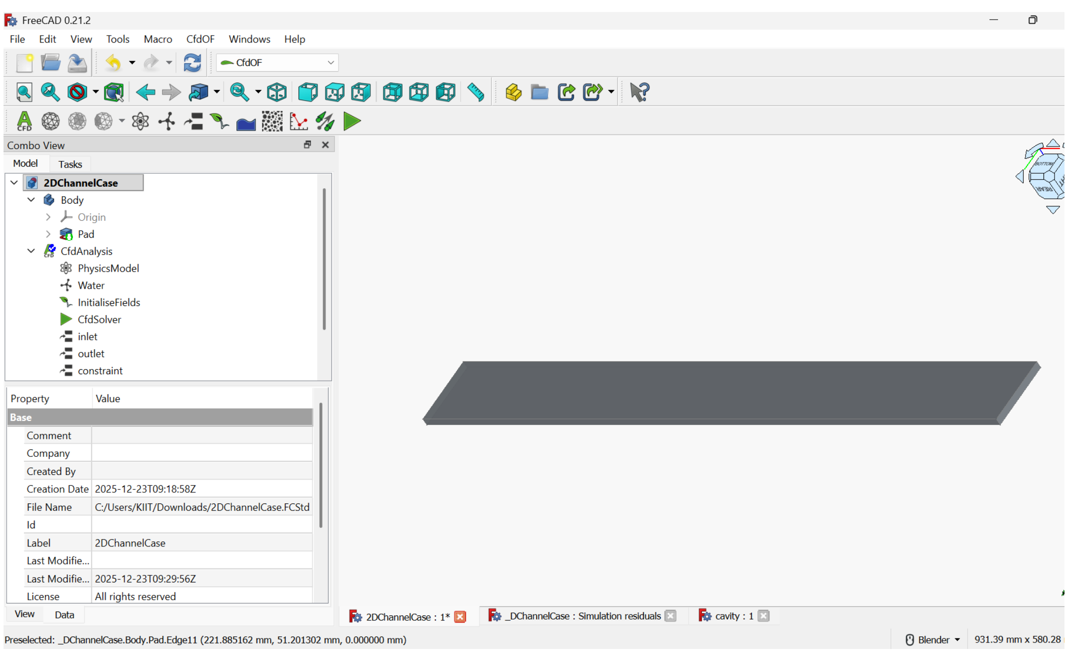

Figure 8: Structured mesh for the 2D channel flow case.

7.4 Boundary Conditions

- **Inlet:** Fixed uniform velocity profile ($U = U_{in}$).
- **Outlet:** Fixed reference pressure ($p = 0$) with zero-gradient velocity.
- **Walls:** No-slip condition ($U = 0$).

7.5 Solver and Numerical Settings

The `simpleFoam` solver was used for steady-state incompressible flow simulations. Discretization schemes and solver tolerances were selected to obtain stable convergence. Under steady conditions, residual monitoring confirmed that velocity and pressure solutions reached convergence.

7.6 Residual Monitoring

Residual plots showed gradual reduction, ensuring convergence of the steady-state solution. Stable residual decay indicates a valid numerical solution.

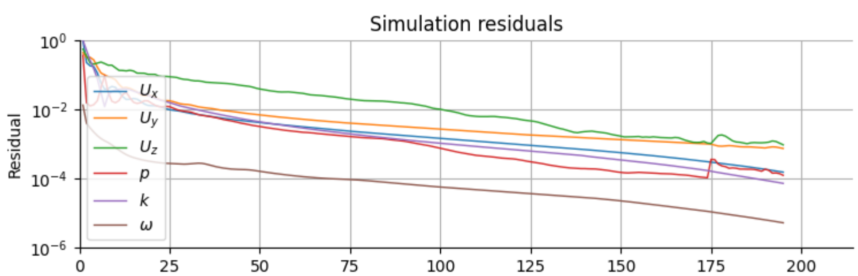

Figure 9: Residual plot for 2D channel steady-state simulation.

7.7 Results and Validation

Velocity contours and streamline visualization confirmed smooth flow development inside the channel. Near the outlet, the velocity profile approached a parabolic distribution, which matches theoretical expectations for laminar fully developed channel flow.

7.8 Velocity Profile Plot (Optional Validation)

The velocity profile extracted at the outlet region was plotted and compared with expected analytical behaviour. A near-parabolic curve confirmed that the simulation correctly captured laminar flow physics and boundary layer development.

This case strengthened understanding of steady-state solver control, pressure-velocity coupling, and wall-bounded flow development, which are essential foundations for advanced CFD problems.

8 Comparison of CLI and GUI Workflows

Feature	Native OpenFOAM (CLI Workflow)	FreeCAD CfdOF Workbench (GUI Workflow)
User Interface	Terminal-based; interaction via shell commands and text editors (nano/vim/VSCode).	Graphical interface with menus, dialogs and 3D view; face-based selection for setup.
Learning Curve	Steep; requires understanding Linux + OpenFOAM dictionaries and workflow.	Beginner-friendly; reduces entry barrier through guided setup.
Case Structure Management	Manual handling of `0/`, `constant/`, `system/`. High chance of human error.	Case directories generated automatically using an analysis container; consistent output structure.
Geometry Handling	Geometry imported (STL/OBJ). Changes require re-export + remeshing.	Parametric CAD geometry; modifications can be updated within same FreeCAD project.
Mesh Generation Workflow	Full control: `blockMesh`, `snappyHexMesh`, `cfMesh`; supports advanced refinement and snapping.	Simplified meshing options with visual controls; refinement is easier but may limit deep customization.
Boundary Condition Assignment	Boundary patches defined in dictionaries; error-prone for beginners due to patch naming.	BCs assigned by selecting faces; reduces mismatch errors and improves clarity.
Solver Selection	Complete freedom: all OpenFOAM solvers + custom solvers.	Mostly supports standard solvers available in UI presets; custom solvers require manual edits.
Numerical Schemes Control	Full access to discretization (`fvSchemes`) and solver tuning (`fvSolution`).	Basic numerical tuning available; advanced schemes often require CLI editing after export.
Error Handling & Debugging	Requires reading logs/residuals manually; high skill required but strong transparency.	Faster feedback and guided logs; easier debugging for common configuration mistakes.
Automation & Scripting	Strong advantage: easy automation using bash/python; supports parametric studies and optimization.	Limited automation; primarily interactive and manual workflow.
Parallel Processing	Full HPC/MPI support (`decomposePar`, `mpirun`, clusters).	Parallel runs possible but simplified; less suited for cluster-scale workflows.
Post-processing	ParaView launch via `paraFoam`; strong for automated result extraction with utilities.	One-click visualization integration; convenient for basic post-processing.
Reproducibility	Highly reproducible via version-controlled dictionaries and scripts.	Reproducible if FreeCAD project and exported case files are archived correctly.
Best Use Case	Advanced CFD research, automation, HPC scaling, custom physics and solver development.	Education, beginner training, rapid prototyping, design iterations for standard flow problems.

Table 1: Comparative analysis between Native OpenFOAM command-line workflow and FreeCAD CfdOF GUI-based workflow.

9 Learning Outcomes

- Practical understanding of CFD fundamentals
- Experience with OpenFOAM solvers
- GUI-based CFD workflow using FreeCAD
- Mesh generation and boundary condition assignment
- Result analysis using ParaView

10 Conclusion

This internship successfully demonstrated the effectiveness of the FreeCAD CfdOF workbench for CFD simulations using OpenFOAM. Through benchmark case studies, the GUI-based workflow was shown to reduce setup complexity while maintaining numerical accuracy. The experience strengthened practical CFD skills and highlighted the importance of open-source tools in engineering education.

References

[1] J. D. Anderson, *Computational Fluid Dynamics*, McGraw-Hill, 1995.

[2] H. K. Versteeg and W. Malalasekera, *An Introduction to CFD*, Pearson, 2007.

[3] OpenFOAM Foundation, *OpenFOAM User Guide*, 2023.

[4] FreeCAD Documentation, https://www.freecad.org

[5] Ahrens et al., *ParaView*, 2005.