Numerical Investigation of Indoor Airflow using a Precursor Inlet Strategy.

Report Submitted by

Gokulpriyadharsan M

B.Sc(Honors).Physics
Kirori mal college
University of Delhi

Guide

Dr. Harikrishnan S

Assistant Professor
Division of Mechanical Engineering, School of Engineering
Cochin University of Science and Technology

Mentor

Mr. Nikhil Chitnavis

PhD Scholar (Applied Mechanics)
Indian Institute of Technology Madras

FOSSEE, IIT Bombay
February 8, 2026

Contents

List of Tables

List of Figures

1 Abstract

The following study presents a OpenFOAM based investigation into the indoor airflow patterns of a classic mixing ventilation scenario, modeled based on the foundational **Neilsen's case** [1] and modified by adding a precursor channel for more realistic inlet profile. The objective is to validate this case study and predict jet patterns and velocity profile at critical areas wich are essential for indoor ventilation cases. Instead of using codes in blockMesh,a CAD based approach is used by integrating FreeCAD and cfMesh to create the computational domain and mesh.

2 Introduction

Background of the topic

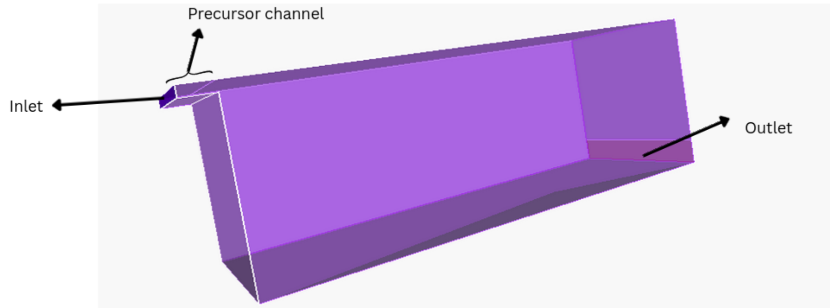
CFD has become an important tool in modelling and testing various engineering and architecturals models for Heat transfer,Ventilation and cooling effeciency. In creating and analysing indoor ventilation, a important aspect of the study is a exerimental validation. Once the CFD model is validated with experimental data, then the results tend to be more accurate and precise and reliable.

Motivation of the study

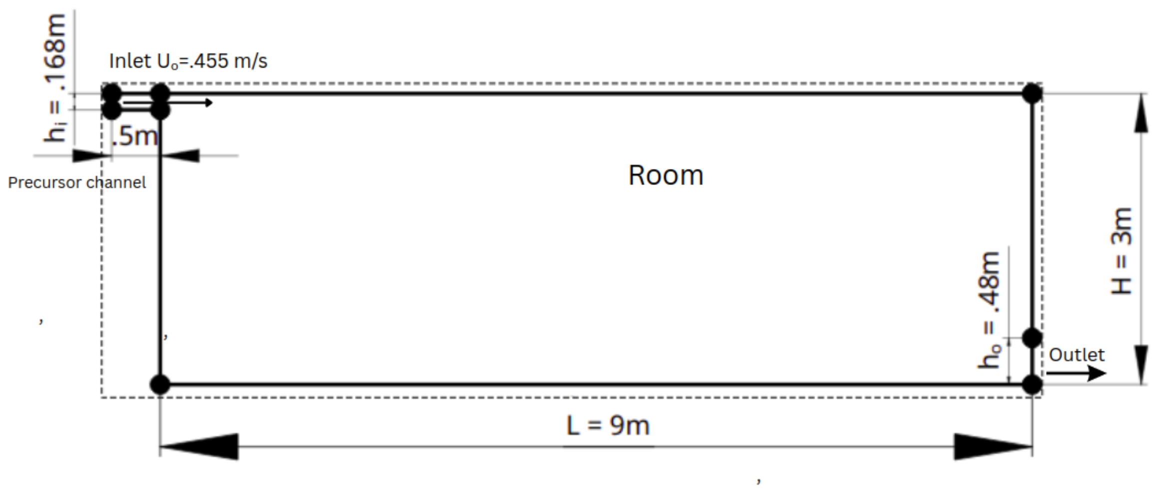
The IEA Annex-20 Nielsen[1] case represents a pioneering study in which CFD flow predictions were validated against experimental results. This study has since become a benchmark for indoor airflow simulations, providing a standard geometry and flow conditions for researchers to test their numerical models. In Indoor HVAC, usually one of the common constraint to be overlooked is the inlet boudary, it is pretty common to assume a uniform one , which is not so realistic. Kang, Luyang, Hooff and Twan [2] demonstrated that results show significant deviations if inlet boundary conditions are not modeled correctly. Thus, even though the geometry in Nielsen case[1] is fairly straightforward, instead of going for a uniform flow, we intended to create a developed flow using a precursor channel before entering the room

2.1 Problem Statement

The objective of this study is to simulate and validate the isothermal airflow described in the standard IEA-Annex 20 Nielsen geometry[1] using OpenFOAM.The results are validated against the experimental data provided by Nielsen with experimental values. The problem's constraints and overview details are summarized in this table 1



(a) 3D Isometric View



(b) 2D Side Profile showing Precursor

Figure 1: Computational domain geometry showing .

Geometry Specification

The computational domain consists of two distinct regions: a precursor channel and the main test chamber (Nielsen's room).The geometry was modified from the standard benchmark by extending the inlet upstream to facilitate the development of a realistic velocity profile through precursor channel before entering the room.

The main domain represents the standard two-dimensional Room airflow test case specified by Nielsen [1]. It is a rectangular room with a length-to-height ratio (L/H) of 3.0. The inlet is located at the top left corner, and the outlet is located at the bottom right corner.

Table 1: Summary of Problem Parameters and Numerical Setup

Category	Parameter	Value / Description
Geometry	Dimensions ($L \times H \times W$)	$9.0m \times 3.0m \times 3.0m$
	Inlet Height (h)	$0.168m$ ($h/H = 0.056$)
	Outlet Height (t)	$0.48m$ ($t/H = 0.16$)
Fluid Properties	Fluid Type	Air (Incompressible, Isothermal)
	Kinematic Viscosity (ν)	$1.5 \times 10^{-5}\, m^2/s$
	Density (ρ)	$1.205\, kg/m^3$
Boundary Conditions	**Inlet**	Fixed Velocity ($U_x = 0.455\, m/s$), Turbulence ($I \approx 4\%$)
	Outlet	Zero Gradient (U), Fixed Pressure ($p = 0$)
	Walls	No-slip ($U = 0$), Standard Wall Functions (k, ϵ, ν_t)
Numerical Setup	Solver	`incompressibleFluid` (OpenFOAM v11)
	Turbulence Model	Standard k-ϵ
	Mesh Type	Hex-dominant Unstructured
	Parallelization	4 Subdomains (Scotch Decomposition)

3 Governing Equations

The simulation uses the steady-state Reynolds-Averaged Navier-Stokes (RANS) equations for incompressible flow [3].

3.1 Mass Conservation (Continuity Equation)

The continuity equation closes the system by providing a constraint on the velocity field.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{1}$$

3.2 Momentum Conservation Equations

The momentum conservation equations are set of scalar partial differential equations (one for each spatial direction).

These three equations contain four unknown variables: the velocity components u, v, w and the pressure p. This system is solved iteratively alongside the continuity equation (Eq. 1) to solve for all four variables.

X-Momentum

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\nu_{eff}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\nu_{eff}\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(\nu_{eff}\frac{\partial u}{\partial z}\right) \tag{2}$$

Y-Momentum

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\nu_{eff}\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\nu_{eff}\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial z}\left(\nu_{eff}\frac{\partial v}{\partial z}\right) \tag{3}$$

Z-Momentum

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{\partial p}{\partial z} + \frac{\partial}{\partial x}\left(\nu_{eff}\frac{\partial w}{\partial x}\right) + \frac{\partial}{\partial y}\left(\nu_{eff}\frac{\partial w}{\partial y}\right) + \frac{\partial}{\partial z}\left(\nu_{eff}\frac{\partial w}{\partial z}\right) \tag{4}$$

3.3 Turbulence Integration

The system is closed by defining the effective viscosity ν_{eff}, which varies spatially based on the turbulence variables calculated by the k-ϵ model:

$$\nu_{eff}(x, y, z) = \nu + C_\mu \frac{k^2}{\epsilon} \tag{5}$$

In this study, the standard k–ε turbulence model is chosen it works better performance for Indoor airflows as by Almeida, Dias de[4].

The inlet turbulence quantities are specified using a turbulence intensity of 4%. The characteristic turbulence length scale is assumed to be 10% of the inlet length $(.168m)$. Based on these assumptions, the turbulent kinetic energy (k) and its dissipation rate (ε) at the inlet are calculated using the standard empirical relations

$$k = \frac{3}{2}(UI)^2, \tag{6}$$

$$\varepsilon = C_\mu^{3/4}\frac{k^{3/2}}{L}, \tag{7}$$

where U is the mean inlet velocity, I is the turbulence intensity, L is the turbulence length scale, and $C_\mu = 0.09$.

Standard wall functions are used for near-wall treatment, ensuring an appropriate balance between numerical accuracy and computational efficiency for indoor airflow simulations.

4 Boundary and initial conditions

The boundary constraints are kept corresponding to the experimental parameters defined by Nielsen's[1]. A summary of the applied boundary and initial conditions is provided in Table 2. The inlet conditions were applied to the precursor channel's entrance such that the flow develops naturally before entering the room.A pressure driven oulet is employed at the room outlet to allow for natural flow extraction. Proper wall functions are emplyed to capture near wall turbulence effects.

Table 2: Detailed Boundary and Initial Conditions

Variable	Initial Field	Inlet	Outlet	Walls
Velocity (U)	Uniform $(0, 0, 0)$	Fixed value $(0.455, 0, 0)\,\mathrm{m/s}$	Zero gradient	No slip
Pressure (p)	Uniform $8.33 \times 10^4\,\mathrm{Pa}$	Zero gradient	Total pressure ($p_0 = 0$)	Zero gradient
Turbulent K.E. (k)	$4.97 \times 10^{-4}\,\mathrm{m^2/s^2}$	Fixed value 4.97×10^{-4}	Zero gradient	`kqRWallFunction`
Dissipation (ϵ)	$1.08 \times 10^{-4}\,\mathrm{m^2/s^3}$	Fixed value 1.08×10^{-4}	Zero gradient	`epsilonWallFunction`
Turbulent Viscosity (ν_t)	Uniform 0	Calculated	Calculated	`nutkWallFunction`

5 Mesh Generation

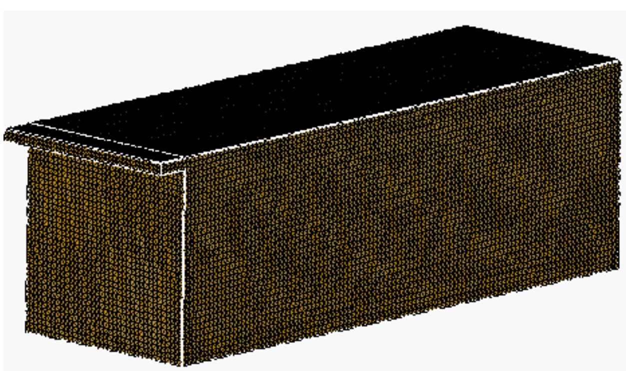
The computational domain was created using **FreeCAD (v0.21)**, an open-source parametric 3D modeler, in addition with **CfdOF** workbench.

1. **2D Profile Sketching:** The side profile of the domain was first drawn on the XY plane. This sketch included both the main test room (9.0m × 3.0m) and the upstream precursor channel (.5m × 0.168m) as a single continuous region.

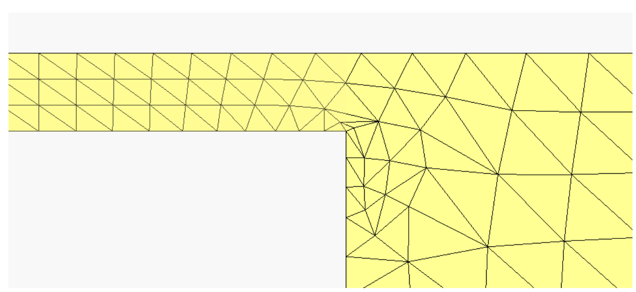
2. **3D Extrusion:** The 2D profile was extruded in the Z-direction by 3.0m to create the full fluid volume.

5.1 Mesh Details

cfMesh - cartesianMesh utility was chosen to create mesh. cfMesh is chosen for its robustness and ease of use. With a 2 degree refinement on precursor channel, cfMesh can directly handle abrupt spacial grid density changes by adding a smooth transition of grid density. A full meshing statergy is explained in the Appendix section.



(a) Generated Mesh (3D)



(b) Shows transition of refinement

Figure 2: Mesh

Mesh Statistics

The final mesh consists of approximately **141448 cells** . The grid is highly structured, with over 99% of the elements being hexahedral, which ensures minimal numerical diffusion and high accuracy for the alignment of the flow with the grid lines.

A breakdown of the cell types is provided in Table 3.

Table 3: Mesh Cell Types and Statistics

Cell Type	**Count**	**Percentage**
Hexahedra	140,578	99.38%
Prisms	0	0.00%
Wedges	0	0.00%
Pyramids	302	0.21%
Tet Wedges	0	0.00%
Tetrahedra	568	0.40%
Polyhedra	0	0.00%
Total	**141,448**	**100%**

Mesh Quality Metrics

The mesh successfully passed all geometric checks in OpenFOAM. The key quality parameters are summarized below in the Table 3, confirming that the grid is suitable for the RANS simulation:

Table 4: Mesh Quality Metrics

Parameter	**Value**
Non-Orthogonality (Average)	2.16°
Non-Orthogonality (Maximum)	61.54°
Skewness (Maximum)	0.99
Aspect Ratio (Maximum)	9.90

Bounding Box: The domain spans from $(-.5\ -3\ 0)$ to $(9\ 0\ 3)$, confirming the inclusion of the precursor channel and the full dimensions of the room.

5.2 Numerical Methods

The simulation was performed using OpenFOAM 11, which utilizes a modular solver architecture. The specific solver module selected was `incompressibleFluid`.

For steady-state simulations [5], this solver employs the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm, making it methodologically equivalent to the traditional `simpleFoam` application found in earlier versions and ESI releases.

Case Decomposition

Scotch decomposition method is used to decompose the case for 4 processors and mpiexe utility is used to run the case in parallel

Discretization Schemes

The governing equations were discretized using the Finite Volume Method (FVM) with Gaussian integration. The specific schemes selected in the `fvSchemes` dictionary are chosen in this study.

- **Time Discretization:**
  - `steadyState`: Used for steady-state analysis (no temporal derivatives $\partial_t = 0$).
- **Divergence (Convection) Terms:**
  - **Velocity ($\nabla \cdot (\phi \mathbf{U})$):** `bounded Gauss linearUpwindV grad(U)`.
    This is a second-order, bounded scheme that minimizes numerical diffusion, better suited for closed ventilation flows
  - **Turbulence ($\nabla \cdot (\phi k), \nabla \cdot (\phi \epsilon)$):** `bounded Gauss upwind`.
    A first-order bounded scheme is used for turbulence quantities to ensure stability and prevent unrealisitc oscilltions for turbulence.

6 Implementation in OpenFOAM

Case Setup Structure

The case directory is organized according to the standard OpenFOAM structure:

- `0/`: Contains initial and boundary conditions for velocity (U), pressure (p), and turbulence variables (k, ϵ, ν_t).
- `constant/`: Stores the mesh data (`polyMesh`) and physical properties dictionaries (`turbulenceProperties`, `transportProperties`).
- `system/`: Contains solver settings (`controlDict`), discretization schemes (`fvSchemes`), solution algorithms (`fvSolution`), and decomposition settings (`decomposeParDict`).
- `Allrun / Allclean`: Scripts to automate the case execution and cleanup.

Control Dictionary

The `controlDict` settings control the time stepping and data output. Key parameters from the simulation are:

- **Start Time:** 0
- **End Time:** 4000 (iterations)
- **DeltaT:** 1 (Geometric-algebraic step for steady solver)
- **Write Interval:** 100
- **Write Control:** timeStep

Running Procedure

To reduce computational time, the simulation was executed in parallel using mpiexe.The domain was decomposed into 4 sub-domains using the **Scotch** method.

The procedure used is as follows:

1. **Decomposition:** The domain was split using the settings in `decomposeParDict`:

   ```
   method          scotch;
   numberOfSubdomains 4;
   ```

   Command: `decomposePar`

2. **Parallel Execution:** The solver can be run on 4 processors using the command:

   ```
   mpirun -np 4 incompressibleFluid -parallel
   ```

3. **Reconstruction:** After convergence, the fields were reconstructed back into a single domain:

   ```
   reconstructPar
   ```

Total Time

The simulation converged on 2228th iterations.Convergence was monitored by checking that the residuals for all fields with a established tolerance (10^{-5}).

7 Results and Discussions

Validation

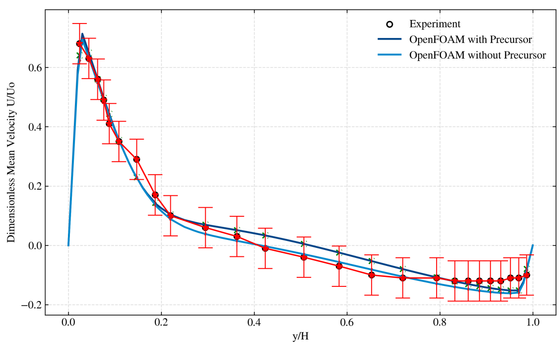
To validate the accuracy of the numerical model, the simulation results were compared against the experimental benchmark data provided by Nielsen [1]. The experimental data was originally acquired using Laser Doppler Anemometry (LDA), which provides high-fidelity measurements of the flow field.

Velocity Profiles

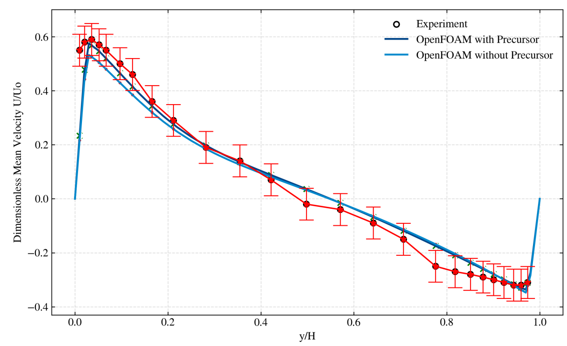
Quantitative validation was performed by comparing the normalized streamwise velocity profiles. The velocity magnitude (U) was normalized by the inlet velocity ($U_0 = 0.455$ m/s), and plotted against the distance.

Data was extracted from the symmetry plane at four critical locations to capture both the wall jet development and the recirculation region:

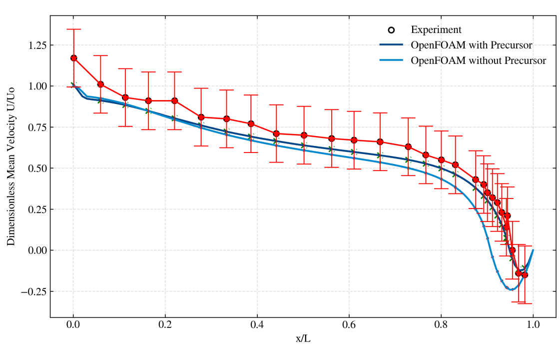
- **Vertical Profiles:** At $x = H$ (3.0 m) and $x = 2H$ (6.0 m). These profiles validate the decay of the wall jet and the velocity distribution in the occupied zone.
- **Horizontal Profiles:** At $y = h/2$ (near the floor) and $y = H - h/2$ (near the ceiling/inlet). These profiles help assess the boundary layer attachment and the return flow near the floor.



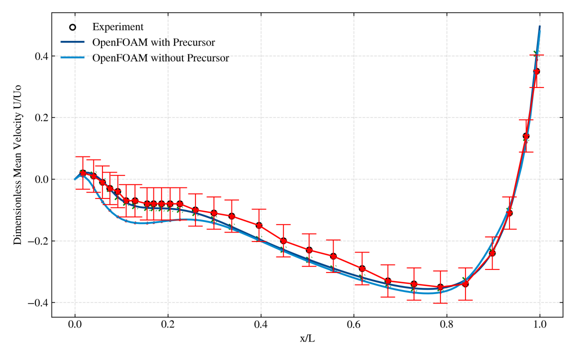
(a) Vertical Profile at $x = H$



(b) Vertical Profile at $x = 2H$



(c) Horizontal at $y = H - h/2$



(d) Horizontal at $y = h/2$

Figure 3: Validation of velocity profiles against experimental data at critical locations with precursor channel.

Improvement due to precursor:

As seen from fig3, the errors have significantly reduced after including the precursor channel The values especially at the boundary is predicted better when the flow is allowed to develop through a precursor. The errors are quantified in the table5.

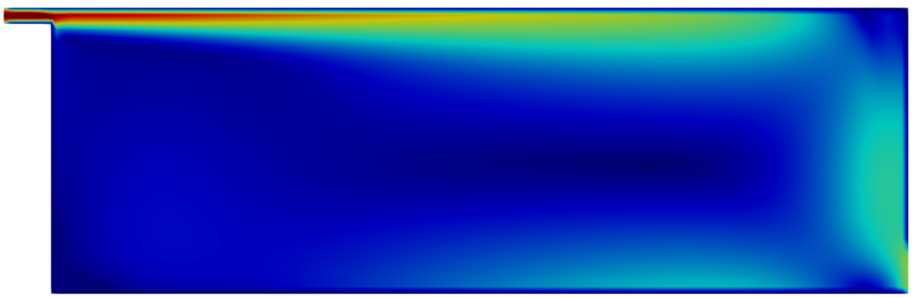
For the error calculation, values close to zero were excluded in order to avoid artificial inflation of the relative error arising from near-zero absolute reference values. This filtering ensures that the reported error percentages remain physically meaningful and numerically stable

Table 5: Error Percentage With and Without Precursor

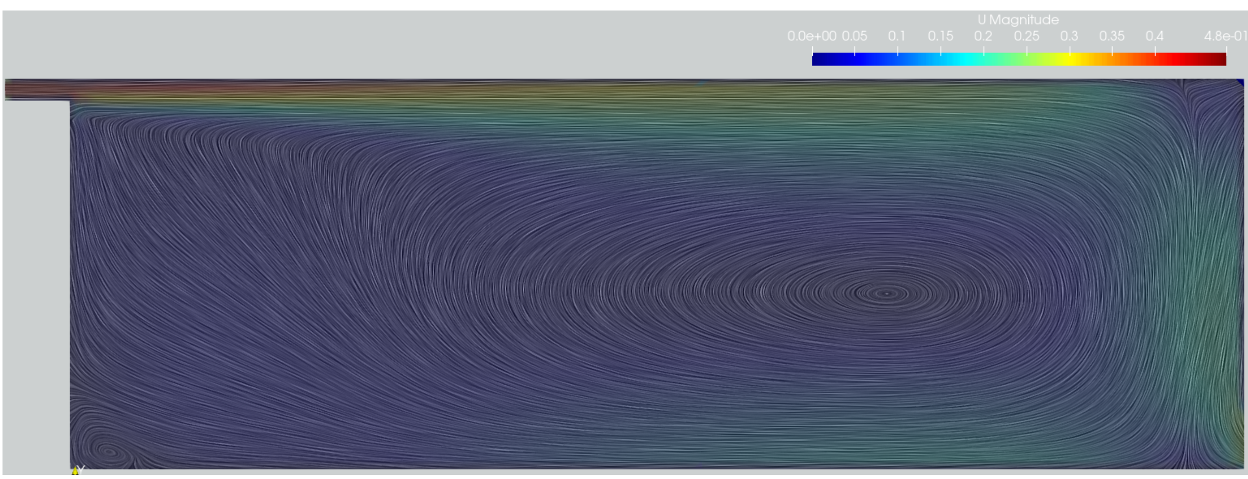
Location	**With Precursor (%)**	**Without Precursor (%)**
$x = H$	17.41	18.46
$x = 2H$	12.38	15.57
$y = h/2$	16.68	51.41
$y = H - h/2$	13.55	36.00

Nature of Flow

The qualitative analysis of the flow field confirms the expected airflow patterns for a ceiling-slot ventilated room.



(a) Velocity Profile - Symmentry plane



(b) Velocity Contour

Figure 4: Velocity profiles and contours in the symmetry plane.

As observed in Fig. 4(a), the high-momentum flow remains attached to the upper boundary, demonstrating a clear Coanda effect.

The streamline pattern shown in Fig. 4(b) shows the formation of a primary circulation zone in the center of the room reveals the formation of a dominant primary circulation zone occupying most of the enclosure.

8 Summary

This study successfully validated the capabilities of OpenFOAM v11 in predicting indoor airflow patterns for the standard Nielsen benchmark. By utilizing the modular `incompressibleFluid` solver and employing a precursor simulation strategy to generate realistic inlet boundary conditions, the numerical model achieved high fidelity against experimental data.

The quantitative validation performed against Nielsen's measurements indicates that:

- The simulation correctly captures the dominant flow features, including the **Coanda effect** of the ceiling jet and the large recirculation zone in the center of the room.
- The average error across the domain lies well within the acceptable margins considered for computational techniques

These results confirm that with appropriate parameter tuning (specifically the inlet turbulence profiles and discretization schemes), the standard k-ϵ model in OpenFOAM is a reliable tool for isothermal indoor airflow prediction.

Limitations of the study

While the results are satisfactory for validation purposes, the current study has certain limitations inherent to the methodology:

Isothermal Conditions: This validation focused strictly on isothermal flow (constant temperature). In real-world HVAC scenarios, buoyancy forces driven by temperature differences (heating or cooling loads) significantly alter the flow patterns, which were not accounted for in this specific validation case

Future work

To extend the applicability of this research to more complex and realistic scenarios, the following future work is proposed:

Non-Isothermal Simulation: The current model should be extended to include the energy equation (buoyancy effects) to simulate heating and cooling scenarios, as discussed in the broader literature [6].

Transient Analysis (LES/URANS): Implementing Large Eddy Simulation (LES) or Unsteady RANS (URANS) would allow for the analysis of time-dependent flow structures and turbulence statistics, providing a more detailed assessment of draft risk.

Pollutant Transport: A scalar transport equation could be added to the solver to simulate the dispersion of contaminants (e.g., CO_2 or pathogens) and calculate the "Age of Air" to assess ventilation efficiency by introducing a lagrangian particle coupling inside eulerian field loops.

Complex Geometry: Future studies could introduce obstacles such as furniture or thermal mannequins to evaluate the impact of obstruction on airflow distribution in the occupied zone.

References

[1] Peter V Nielsen. *Specification of a Two-Dimensional Test Case*. Tech. rep. R9040. Aalborg University, 1990.

[2] Luyang Kang and Twan van Hooff. "Influence of inlet boundary conditions on 3D steady RANS simulations of non-isothermal mechanical ventilation in a generic enclosure". In: *International Journal of Thermal Sciences* 182 (2022), p. 107792.

[3] Christopher J. Greenshields. *Notes on CFD: General Principles*. Accessed: 2024-05-20. CFD Direct. URL: `https://doc.cfd.direct/notes/cfd-general-principles/`.

[4] Célia Patrícia Dias de Almeida. "Turbulence Models Study Applied to Indoor Space Ventilation". Master's Dissertation. Instituto Superior de Engenharia do Porto, 2021.

[5] The OpenFOAM Foundation. *OpenFOAM v11 User Guide: 3.5 Solver Modules*. Accessed: 2024-05-20. CFD Direct. 2023. URL: `https://doc.cfd.direct/openfoam/user-guide-v11/solvers-modules`.

[6] S Murakami and S Kato. "Diffusion characteristics of airborne particles with gravitational settling in a convection-dominant indoor flow field". In: *ASHRAE Transactions* 95.2 (1989). Paper NO. 3551, pp. 388–397.

9 Annexure:Mesh Generation and meshCase setup

9.1 Setting up FreeCAD and cfMesh

This section outlines the procedure for installing the software environment required for geometry generation and meshing.

9.1.1 1. Installing FreeCAD (AppImage)

To ensure full system access and avoid permission issues associated with containerized packages (like Snap or Flatpak), the AppImage version of FreeCAD is recommended.

1. **Download:** Navigate to the official FreeCAD downloads page (`https://www.freecad.org/downloads.php`) and download the **Linux AppImage** (64-bit).

9.1.2 2. Installing the CfdOF Workbench

The CfdOF workbench is an external add-on that integrates OpenFOAM solvers and meshing tools into FreeCAD.

1. Open FreeCAD and navigate to **Tools** → **Addon Manager**.
2. In the search bar, type `CfdOF`.
3. Select the workbench from the list and click **Install**.
4. Restart FreeCAD to apply the changes.

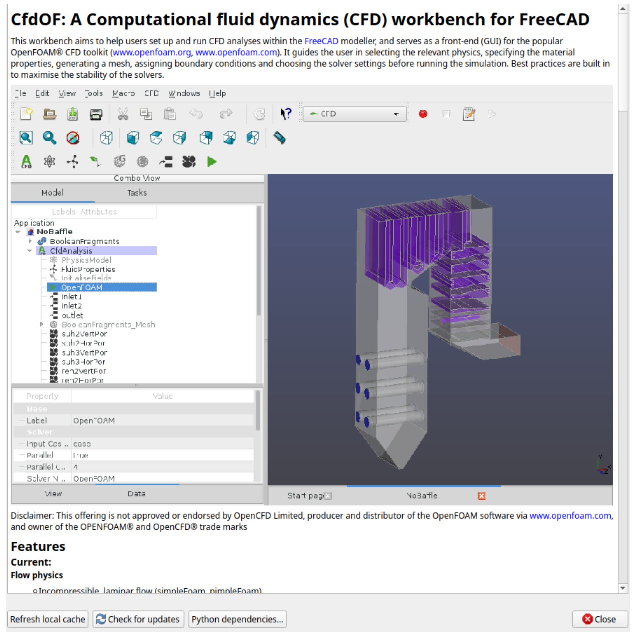


Figure 5: cfdOF installation

9.1.3 3. Configuring cfMesh Dependencies

CfdOF requires external meshing utilities (cfMesh) to be present on the system. The workbench includes a utility to download these automatically.

1. Switch to the **CfdOF** workbench using the dropdown menu.
2. Navigate to **Edit** → **Preferences**.
3. Select **CfdOF** from the sidebar and open the **Updates/Dependencies** tab.
4. Click **Install** next to the **cfMesh** and **HISA** entries.
5. Ensure the installation path is writable and confirmed.

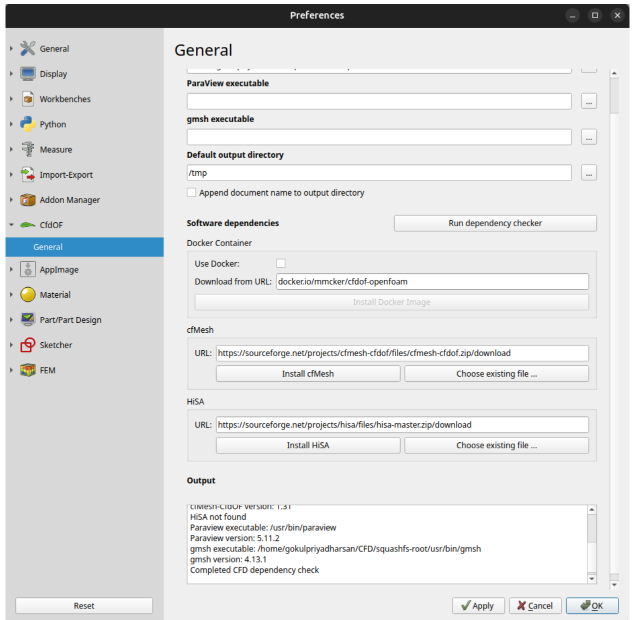


Figure 6: cfdOF installation

9.2 Geometry Generation and Meshing

9.2.1 Domain Cration

The geometry for the simulation was constructed using **FreeCAD**, an open-source parametric 3D modeler. The **Part** and **Part Design** workbenches were used to create the 3D computational domain profile.

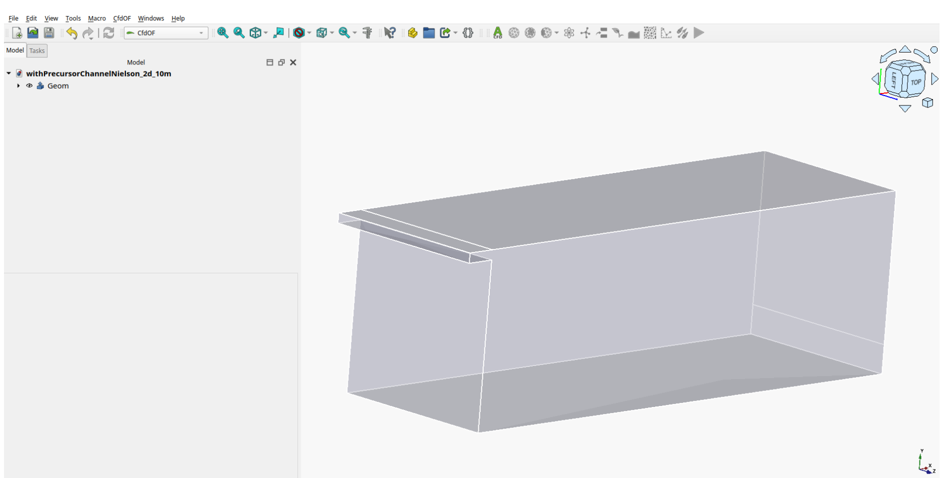


Figure 7: Computational domain created in FreeCAD (Part Workbench)

9.2.2 Naming Boundaries

Once the solid geometry is created, the **CfdOF** workbench (computational fluid dynamics OpenFOAM addon) was used to define the simulation boundaries.

Specific faces of the geometry were selected and assigned patch names (e.g., *Inlet*, *Outlet*, *Walls*) types (such as `patch` or `wall`). This step ensures that the boundary conditions are correctly mapped when the mesh is exported to OpenFOAM.

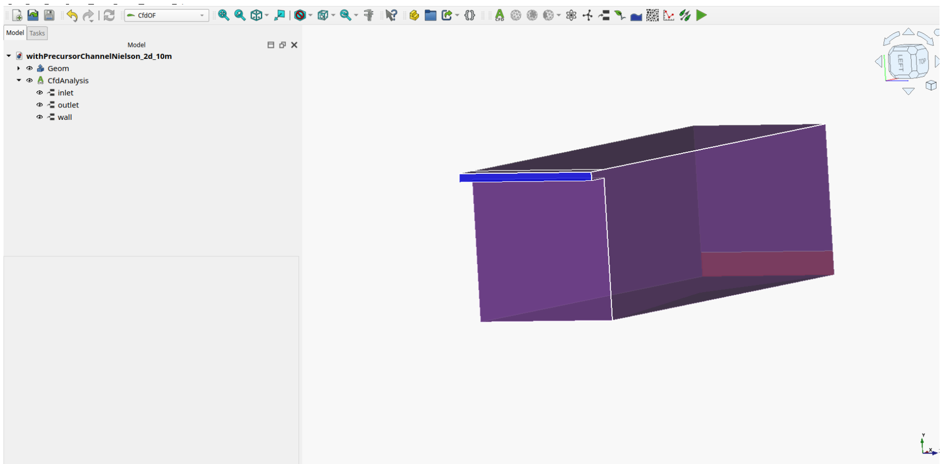


Figure 8: Assignment of boundary patches in CfdOF

9.2.3 Mesh Generation using CfdOF

The meshing process was handled by the **CfdOF** workbench, which acts as a GUI frontend for OpenFOAM meshing tools such as**SnappyHexMesh**,**cfMesh**,**gMesh** .

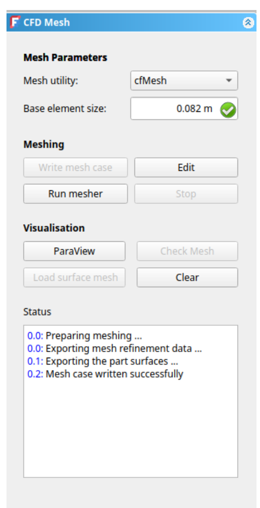


Figure 9: Meshing parameters window

A meshCase directory would be creating with these stl files inside it. The meshCase file system is structurally similar to OpenFOAM file system, thus parameters for meshing can be edited flexibly.