

Open Source CFD

Tutorial #1: OpenFOAM with terminal

Incompressible Steady Flow: pitzDaily with simpleFoam

Robert Castilla

Department of Fluid Mechanics

2025-26

Version 1.0

Learning Objectives:

- Navigate OpenFOAM case directory structure
- Understand basic OpenFOAM file formats
- Run a simple steady-state simulation
- Analyze basic results in terminal
- Troubleshoot common errors

This tutorial introduces the fundamental workflow of Computational Fluid Dynamics (CFD) using OpenFOAM in a terminal environment. Through the classic `pitzDaily` case with the `simpleFoam` solver, students will learn to set up, run, and analyze an incompressible steady-state flow simulation. No graphical interface is required—all operations are performed via command line, making this ideal for remote servers and building core CFD skills.

Contents

1. Introduction and Theoretical Background	3
1.1. What is pitzDaily?	3
1.2. Governing Equations	3
1.3. Case Overview	4

2. Terminal Environment Setup	4
2.1. Accessing OpenFOAM	4
2.2. Navigating to Tutorials	4
3. Case Structure Exploration	5
3.1. Understanding OpenFOAM Directory Structure	5
3.2. Examining Key Files	5
4. Running the Simulation	6
4.1. Pre-processing	6
4.2. Running simpleFoam	7
4.3. Monitoring Convergence	7
5. Post-Processing in Terminal	8
5.1. Basic Results Examination	8
5.2. Using OpenFOAM Command Line Tools	8
5.3. Basic Visualization in paraview	9
6. Common Issues and Troubleshooting	9
6.1. Error: "command not found"	9
6.2. Error: "cannot find blockMeshDict"	10
6.3. Simulation Diverges (Residuals Grow)	10
7. Student Exercises and Extensions	11
7.1. Basic Exercises (Required)	11
7.2. Advanced Exercises (Optional)	11
8. Deliverables and Assessment	11
8.1. Required Files	12
8.2. Report Questions	12
8.3. Submission Format	12
A. Appendix: Useful Terminal Commands Reference	13
A.1. OpenFOAM-Specific	13
A.2. General Linux Commands	13
B. Quick Reference: File Locations in pitzDaily	13

1. Introduction and Theoretical Background

1.1. What is pitzDaily?

The `pitzDaily` case simulates 2D steady, turbulent flow in a backward-facing step configuration. It's included in the OpenFOAM tutorials and serves as an excellent starting point due to its:

- Simple geometry but interesting physics (recirculation zone)
- Complete setup with all necessary files
- Quick runtime (minutes on most computers)

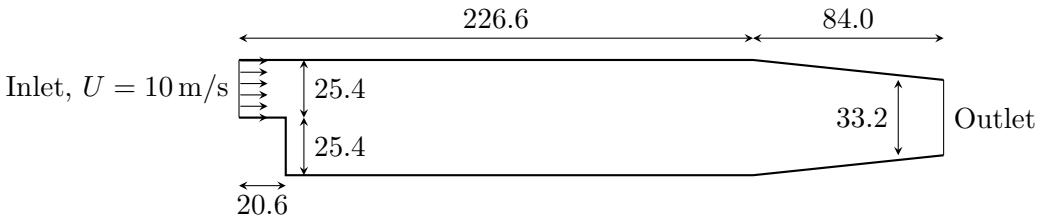


Figure 1: pitzDaily geometry. Lengths are in millimeter.

More information can be found on the [OpenFOAM website](#).

1.2. Governing Equations

The `simpleFoam` solver implements the incompressible Reynolds-Averaged Navier-Stokes (RANS) equations:

$$\nabla \cdot \mathbf{U} = 0 \quad (1)$$

$$\nabla \cdot (\mathbf{U}\mathbf{U}) = -\nabla \left(\frac{p}{\rho} \right) + \nabla \cdot (\nu_{\text{eff}} \nabla \mathbf{U}) \quad (2)$$

where $\nu_{\text{eff}} = \nu + \nu_t$ is the effective viscosity (molecular + turbulence viscosity).

Note that, since the fluid is incompressible, instead of pressure p , this equation deals with kinematic pressure , $\frac{p}{\rho}$, that has units of m^2/s^2 .

IMPORTANT: The solver `simpleFoam` keeps the name p for the kinematic pressure.

1.3. Case Overview

Parameter	Value
Solver	<code>simpleFoam</code> (steady, incompressible)
Turbulence model	$k-\epsilon$ (standard)
Flow regime	Turbulent
Geometry	2D backward-facing step
Reynolds number	$\sim 2 \times 10^4$ (based on step height)
Expected runtime	5-10 seconds

Table 1: pitzDaily case specifications

2. Terminal Environment Setup

2.1. Accessing OpenFOAM

Open your terminal and verify OpenFOAM is available:

```
foamVersion
```

IMPORTANT: If you see "command not found," you may need to source the OpenFOAM environment first:

```
# For system-wide installation
source /usr/lib/openfoam/openfoam-2312/etc/bashrc
```

2.2. Navigating to Tutorials

OpenFOAM includes a comprehensive set of tutorials. Navigate to the incompressible tutorials:

```
# Go to OpenFOAM tutorials directory
cd $FOAM_TUTORIALS

# List available tutorials
ls

# Navigate to incompressible simpleFoam tutorials
cd incompressible/simpleFoam
```

3. Case Structure Exploration

3.1. Understanding OpenFOAM Directory Structure

Copy the pitzDaily case in a local folder

```
# Make a local folder
mkdir -p $HOME/Tutorials/Tutorial_1

# Copy the pitzDaily tutorial in this folder
cp -r ./pitzDaily $HOME/Tutorials/Tutorial_1

# Go to the tutorial place
cd $HOME/Tutorials/Tutorial_1
```

IMPORTANT: Never play with a tutorial in the OpenFOAM distribution home folder. Always make a copy. In this way you will have always a "clean" copy.

Examine the pitzDaily case structure:

```
# Go to the pitzDaily case
cd pitzDaily

# List all contents with details
ls -la

# See the directory structure
tree -L 2
```

IMPORTANT: Every OpenFOAM case has this standard structure:

```
pitzDaily/
|-- 0/           # Initial and boundary conditions
|-- constant/    # Physical properties and mesh
|   |-- polyMesh/ # Mesh files
|   '-- transportProperties
\-- system/      # Solution control
    |-- controlDict
    |-- fvSchemes
    '-- fvSolution
```

3.2. Examining Key Files

Let's look at the most important configuration files:

Step 1: Initial Conditions (0/ directory):

```
# View velocity boundary conditions  
cat 0/U  
  
# View pressure boundary conditions  
cat 0/p  
  
# View turbulence variables  
cat 0/k  
cat 0/epsilon
```

Step 2: Physical Properties:

```
# Check fluid properties  
cat constant/transportProperties
```

This shows:

```
transportModel Newtonian;  
  
nu 1e-05;
```

Step 3: Solution Control:

```
# Check solver settings  
cat system/fvSolution  
  
# Check discretization schemes  
cat system/fvSchemes  
  
# Check runtime control  
cat system/controlDict
```

TIP: Use `less` or `more` for longer files: `less system/controlDict`

4. Running the Simulation

4.1. Pre-processing

Before running, we need to generate the mesh:

```
# Generate the mesh using blockMesh, with the foamJob utility  
foamJob -screen -log-app blockMesh  
  
# Check mesh quality  
foamJob -screen -log-app checkMesh
```

WARNING: If `blockMesh` fails, check for error messages. Common issues include missing dictionaries or syntax errors in `system/blockMeshDict`.

The utility `foamJob` creates a log file.

```
# Check the content of the log files
less log.blockMesh
less log.checkMesh
```

4.2. Running simpleFoam

Now run the simulation:

```
# Run the solver
foamJob -s -log-app simpleFoam
```

4.3. Monitoring Convergence

Monitor convergence:

```
# Watch residuals in real-time (if running in foreground)
# OR check the log file periodically:

# Show last 20 lines of residuals
tail -20 log.simpleFoam | grep "Solving for"

# Check final residuals
tail -50 log.simpleFoam | grep "Final residual"
```

Plot it using `gnuplot`

```
# Generate residual files from the log file
foamLog log.simpleFoam

# Visualize the residuals of the main variables with gnuplot
gnuplot
gnuplot> set logscale y
gnuplot> plot 'logs/p_0' with lines , 'logs/Ux_0' with lines, 'logs/Uy_0' with
lines, 'logs/epsilon_0' with lines, 'logs/k_0' with lines
```

You will get a picture similar to that in Figure 2.

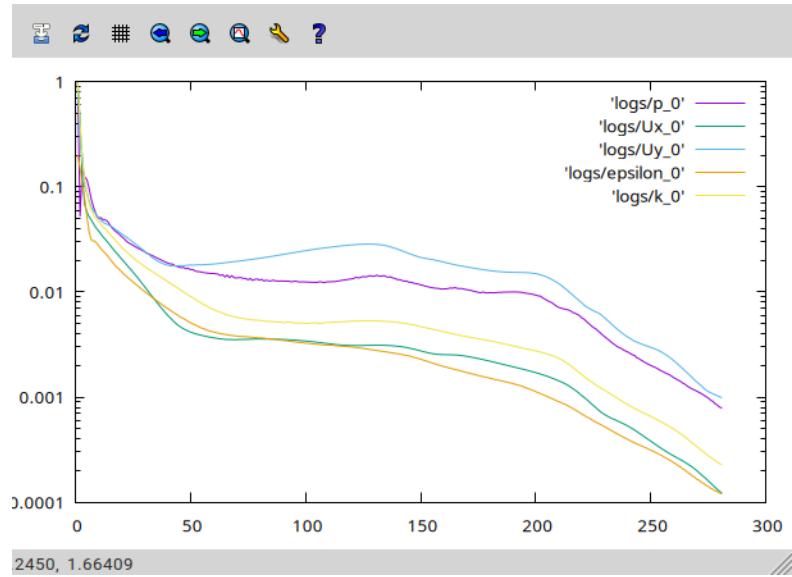


Figure 2: Plot of residuals

5. Post-Processing in Terminal

5.1. Basic Results Examination

OpenFOAM stores results in time directories. Examine the final results:

```
# List time directories
foamListTimes

# Go to final time directory (e.g., typically 281 for pitzDaily)
cd 281

# List available fields
ls
```

5.2. Using OpenFOAM Command Line Tools

Check field statistics:

```
# Go back to case root
cd ..

# Get pressure statistics
postProcess -func writeCellCentres
postProcess -func "patchAverage(name=inlet,p)"
postProcess -func minMaxMagnitudes
```

IMPORTANT: Besides the screen, the information from `postProcess` is also saved in the `postProcessing` folder

TIP: The command `postProcess -list` will list all the functions available.

5.3. Basic Visualization in paraview

Usually the program `paraview` is used to visualize the results from OpenFOAM simulations. A wrapper `paraFoam` is available to call the program after performing some internal operations

```
# Run the paraFoam script  
paraFoam
```

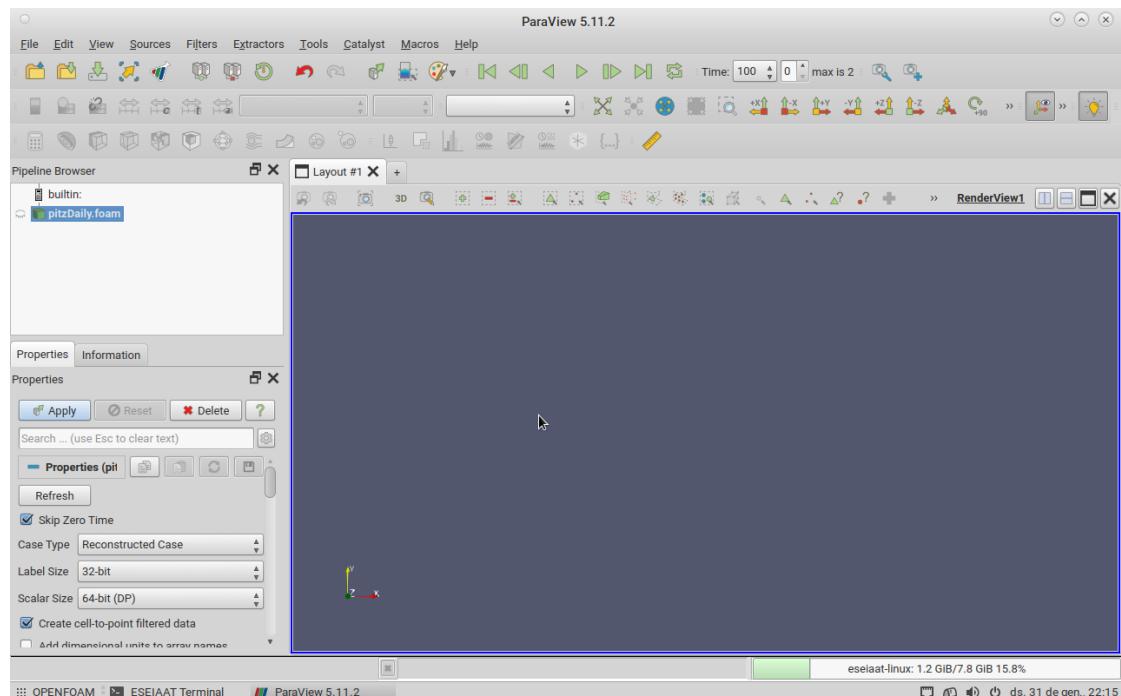


Figure 3: Start screen for paraview, called with `paraFoam`.

Click on the "Apply" button. By default, you will see the pressure distribution (Figure 4).

6. Common Issues and Troubleshooting

6.1. Error: "command not found"

- **Cause:** OpenFOAM environment not sourced

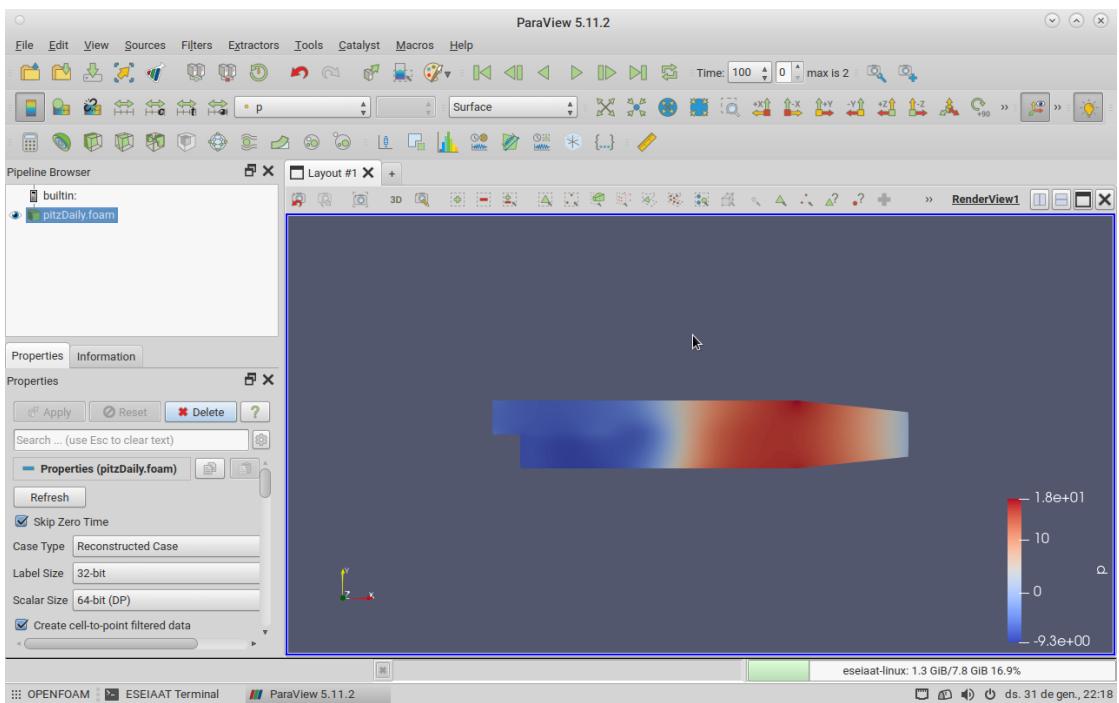


Figure 4: Pressure distribution of the flow.

- **Solution:**

```
source /path/to/OpenFOAM/etc/bashrc
```

6.2. Error: "cannot find blockMeshDict"

- **Cause:** Not in correct directory or file missing
- **Solution:**

```
# Check current directory
pwd

# List contents of system
ls system
```

6.3. Simulation Diverges (Residuals Grow)

- **Cause:** Poor initial conditions or solver settings
- **Solution:**

1. Reduce under-relaxation factors in `system/fvSolution`
2. Check boundary conditions are physically realistic
3. Try running `potentialFoam` first to get better initial fields

7. Student Exercises and Extensions

7.1. Basic Exercises (Required)

1. **Boundary Condition Modification:**
 - Change inlet velocity from 10 m s^{-1} to 20 m s^{-1}
 - Update both `0/U` and `0/k/0/epsilon` appropriately
 - **Question:** How does Reynolds number affect recirculation zone size?
2. **Turbulence Model Comparison:**
 - Change from $k-\epsilon$ to $k-\omega$ SST in `constant/turbulenceProperties`
 - **Question:** Which model converges faster? Which gives higher k values?

7.2. Advanced Exercises (Optional)

1. **Mesh Refinement Study:**
 - Modify `system/blockMeshDict` to double cells in x-direction
 - Compare velocity profiles at outlet
2. **Custom Geometry:**
 - Create a simple 2D channel (modify `blockMeshDict`)
 - Reuse boundary conditions from `pitzDaily`
 - Compare pressure drop with analytical solution

8. Deliverables and Assessment

Submit the following to Moodle:

8.1. Required Files

1. Modified `system/controlDict` with your student ID in comments
2. Final `log.simpleFoam` file
3. Screen capture of terminal showing:
 - Successful `blockMesh` execution
 - Final residuals from `simpleFoam`
 - Output of `checkMesh`

8.2. Report Questions

Answer the following in a PDF report:

1. What are the final residuals for U_x , U_y , p , k , and ϵ ?
2. How many iterations were needed for convergence?
3. What is the maximum pressure in the domain? Minimum?
4. Based on the recirculation zone length, what is the reattachment length?
5. If you changed parameters in exercises, discuss how they affected results.

8.3. Submission Format

- **Filename:** StudentID_pitzDaily.zip
- **Contents:**

```
StudentID_pitzDaily/
|-- report.pdf
|-- controlDict_modified
|-- log.simpleFoam
|-- screenshots/
|   |-- blockMesh.png
|   |-- finalResiduals.png
|   |-- checkMesh.png
\-- README.txt (any notes/issues encountered)
```

A. Appendix: Useful Terminal Commands Reference

A.1. OpenFOAM-Specific

Command	Purpose
<code>foamJob</code>	Run an openfoam command saving the log file
<code>blockMesh</code>	Generate block-structured mesh
<code>checkMesh</code>	Check mesh quality
<code>simpleFoam</code>	Run steady incompressible solver
<code>foamLog</code>	Parse log file for residuals
<code>postProcess</code>	Execute function objects
<code>paraFoam</code>	Launch visualization (if GUI available)
<code>reconstructPar</code>	Reconstruct parallel case

A.2. General Linux Commands

Command	Purpose
<code>grep "keyword" file</code>	Search for text in file
<code>tail -f log</code>	Follow log file in real-time
<code>head -n 20 file</code>	Show first 20 lines
<code>cd ~</code>	Go to home directory
<code>pwd</code>	Print working directory
<code>ls -la</code>	List all files with details
<code>tree -L 2</code>	Show directory tree (2 levels)

B. Quick Reference: File Locations in pitzDaily

- Mesh definition: `constant/polyMesh/blockMeshDict`
- Boundary conditions: `0/U`, `0/p`, `0/k`, `0/epsilon`
- Solver settings: `system/fvSolution`
- Discretization schemes: `system/fvSchemes`
- Runtime control: `system/controlDict`
- Material properties: `constant/transportProperties`, `constant/turbulenceProperties`

Acknowledgments: The author acknowledges the assistance of DeepSeek, an AI assistant created by DeepSeek Company, for providing valuable support in structuring this document, formatting LaTeX code, and developing the tutorial content.

Disclaimer: All content has been reviewed, modified, and validated by the author. The author takes full responsibility for the accuracy, completeness, and educational quality of this material.