Research in Industrial Projects for Students

Sponsor

AECOM

Final Report

Comparison of Computational Fluid Dynamics Software Packages in Modeling Pollutant Dispersion

Student Members

Emily Ho (Project Manager), *Princeton University*, emilyho@princeton.edu

Evan Dientsman, *College of William & Mary*

Justin Jiaze Sun, *University of Hong Kong*

Alex Si Ao Wang, *Hong Kong University of Science & Technology*

Academic Mentor

Chi-Wai Yu, macwyu@ust.hk

Sponsoring Mentors

Josh Lam

Karl An, karl.an@aecom.com

Date: August 12, 2016

This project was jointly supported by AECOM and NSF Grant 1460018.

Contents

Abstract **4**

Acknowledgments **5**

1 Introduction **6**
1.1 AECOM 6
1.2 Project Background 6
1.3 Problem Statement 6

2 The Mathematics **8**
2.1 Navier-Stokes Equations 8
2.2 Reynolds-Averaged Navier-Stokes Equations (RANS) 8
2.3 The Scalar Transport Equation 9
2.4 Turbulence Models 9

3 Validation of OpenFOAM against Fluent and Experimental Data **11**
3.1 Pre-Processing 11
3.2 Solving for Velocity and Pollutant Concentration 13
3.3 Sampling the data 14

4 Results **15**
4.1 Simulation Results 15
4.2 Analysis 16
4.3 Discussion 18

5 Conclusion **20**

A Constants **21**

B Formulae **22**

C Additional Figures **23**

Bibliography **30**

List of Figures

3.1 TPU Geometric Model 12
3.2 CFD Mesh 13
3.3 Sample Locations 14

4.1 Fluent Residual Plot 15
4.2 OpenFOAM Residual Plot 16
4.3 Example Velocity Plots 17
4.4 Example Concentration Plots 17
4.5 Velocity Magnitude Error 18
4.6 Concentration Error 18

5.1 Example Simulation for Urban Environment 20

C.1 CFD Geometric Model 23
C.2 Wall y^+ Values 24
C.3 Inlet Velocity Profile 24
C.4 Inlet TKE (k) Profile 25
C.5 Inlet TKE Dissipation Rate (ϵ) Profile 25
C.6 Blue Wall Normalized Velocity Magnitude 26
C.7 Blue Wall Normalized Concentration 26
C.8 Cyan Wall Normalized Velocity Magnitude 27
C.9 Cyan Wall Normalized Concentration 27
C.10 Green Wall Normalized Velocity Magnitude 28
C.11 Green Wall Normalized Concentration 28
C.12 Red Wall Normalized Velocity Magnitude 29
C.13 Red Wall Normalized Concentration 29

Abstract

When designing a building, city planners and engineers must take into account a building's effect on air pollution in order to minimize the negative health and environmental impacts. The first step in accomplishing this task is to understand how wind and pollutants move around buildings. In the past, scientists have conducted expensive wind tunnel experiments to gather this information. However, as computers have become increasingly more powerful, the scientific community has shifted toward using relatively inexpensive computational fluid dynamics (CFD) software to replace the wind tunnels. CFD software employs numerical methods to solve the partial differential equations that describe the motion of wind and pollutants. Currently, the leading commercial CFD software is Fluent, but one open-source software called OpenFOAM has recently started gaining traction. Under the direction of AECOM, an engineering consulting firm, our goal is to test the performance of OpenFOAM against Fluent to justify the use of OpenFOAM in research and consulting. We will use a wind tunnel experiment taken from Tokyo Polytechnic University (TPU) to serve as a control case in evaluating performance. The wind tunnel experiment consists of a single block with wind entering perpendicular to the front face and pollutant being released from the rear. With additional software packages, we can create the same geometric setup on the computer, then solve the equations of motion for wind velocity and pollutant concentration in each software package. Our computer simulations demonstrate that Fluent provides closer results to the wind tunnel data than OpenFOAM does. However, further work still needs to be done in studying OpenFOAM's capabilities and performance in different cases.

Acknowledgments

We would like to thank our industrial mentors, Karl An and Josh Lam, for their instruction and support. We are also grateful for the help provided by our academic mentor, Dr. Chi-Wai Yu. Many thanks to the Hong Kong University of Science and Technology for hosting, and to the Institute for Pure and Applied Mathematics for organizing the RIPS program. This project was supported by NSF Grant 1460018.

The OpenFOAM simulations presented in this article were performed on computational resources supported by the Princeton Institute for Computational Science and Engineering (PICSciE) and the Office of Information Technology's High Performance Computing Center and Visualization Laboratory at Princeton University.

Chapter 1

Introduction

1.1 AECOM

AECOM is a globally recognized engineering firm that offers design, consulting, construction, and management services to over 150 countries around the world, employing around 95,000 individuals. With major offices in North America, Europe, Asia, and Australia, AECOM provides technical and management solutions in Architecture & Design, Asset Management, Construction, Engineering, Environmental Services, Planning & Consulting, Risk Management & Resilience, Technical Services, etc., to clients from around the world. In 2015, Engineering News-Record ranked AECOM first on its Top Global Design Firm and Top Green Building Design Firms, and the 2016 Fortune 500 list ranked AECOM number 156.

1.2 Project Background

Air pollution has been a serious issue around the world for decades. As millions of people live and work in densely packed urban areas, it is essential to study the wind flow and pollutant dispersion patterns in these districts so that solutions can be proposed accordingly. Such problems are usually tackled either by performing wind tunnel experiments, or — thanks to modern digital computers — by running simulations using computational fluid dynamics (CFD) software. Because air quality consulting and engineering are a major component of AECOM's environmental services, CFD plays an instrumental role in the company's daily operations.

1.3 Problem Statement

Before using CFD to study pollutant dispersion in various environments, researchers first must confirm its ability to accurately simulate the physical processes to a reasonable degree. To that end, we seek to compare the accuracy of two different CFD software packages: ANSYS Fluent v17.0 and OpenFOAM v1606+. Fluent is a popular commercial CFD software package which includes a wide range of physical modelling capabilities, such as turbulence modelling, acoustics, heat transfer, and radiation. However, Fluent lacks flexibility because its source code cannot be changed, which presents difficulties in certain situations.

As an alternative, OpenFOAM, developed by OpenCFD Ltd., is a free, open-source CFD software used extensively by both academic and commercial organizations in many fields of engineering and science. It also incorporates a wide range of modelling capabilities, and is able to solve complex fluid flows involving turbulence, heat transfer, acoustics, chemical reactions, solid mechanics, and electromagnetics. Because of the versatile nature of open-source code and the fact that no license is required, OpenFOAM presents an attractive option.

The main goal of this project was to validate the robustness of the wind and pollutant field simulation results obtained by OpenFOAM and ANSYS Fluent with experimentally collected wind tunnel data. Afterwards, both packages can be used to predict wind patterns in real environments.

Chapter 2

The Mathematics

The governing equations of the study of fluid dynamics cannot be solved analytically, so researchers must resort to numerical methods. In this chapter, we present the mathematics upon which CFD software is based.

2.1 Navier-Stokes Equations

The Navier-Stokes equations are the foundation of the study of fluid dynamics [6]. We assume that the fluids in our case are isothermal and incompressible; that is, both the temperature (298K) and the fluid density ρ are constant throughout. In Einstein notation, which will be used from here on, the incompressible NS equations are as follows:

$$\begin{aligned} &\text{(Continuity)} \quad \frac{\partial u_i}{\partial x_i} = 0, \\ &\text{(Momentum)} \quad \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2}, \end{aligned}$$

where the flow velocity $\mathbf{u}(\mathbf{x}, t) = u_1(\mathbf{x}, t)\hat{\imath} + u_2(\mathbf{x}, t)\hat{\jmath} + u_3(\mathbf{x}, t)\hat{k}$ is a function of space and time, p is the pressure, μ is the dynamic viscosity, and $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity.

2.2 Reynolds-Averaged Navier-Stokes Equations (RANS)

The Reynolds decomposition separates any scalar variable $\phi(\mathbf{x}, t)$ of space (denoted by $\mathbf{x}$) and time (denoted by t) into a time-averaged component and a fluctuating component [6]:

$$\Phi(\mathbf{x}, t) = \bar{\phi}(\mathbf{x}) + \phi'(\mathbf{x}, t),$$

where Reynolds time-averaging is defined as

$$\bar{\phi}(\mathbf{x}) \equiv \lim_{\tau \to \infty} \frac{1}{\tau} \int_t^{t+\tau} \phi(\mathbf{x}, t)\, \mathrm{d}t.$$

This technique allows us to glean information about the overall flow, which is otherwise difficult to do because of the transient nature of turbulence.

The RANS equations are shown below, following the convention $u(\mathbf{x}, t) = U(\mathbf{x}) + u'(\mathbf{x}, t)$, where $U(\mathbf{x}) = \bar{u}(\mathbf{x}, t)$:

$$\text{(Continuity)} \quad \frac{\partial u_i'}{\partial x_i} = 0,$$

$$\text{(Momentum)} \quad \frac{\partial}{\partial t}(\rho U_i) + \frac{\partial}{\partial x_j}(\rho U_i U_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}(\mu \overline{S_{ij}}) - \frac{\partial}{\partial x_j}(\rho \overline{u_i' u_j'}),$$

where $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ is the mean strain rate tensor.

The last term of the RANS equation of momentum, called the Reynolds Stress, is nonlinear and thus poses difficulties when we try to solve the system. Hence, it is simplified using the Boussinesq eddy viscosity assumption as shown below:

$$\rho \overline{u_i' u_j'} = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3} k \rho \delta_{ij},$$

where μ_t is the eddy viscosity [6]. The new terms introduced in this assumption require additional equations; this is known as the closure problem.

2.3 The Scalar Transport Equation

The transport equation describes the change in quantity at a point in space as a result of local flow and concentration gradient:

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (D \nabla \phi) - \nabla \cdot (\mathbf{u}\phi) + R,$$

where ϕ is a generic scalar as in the previous section, D is a diffusion coefficient, $\mathbf{u}$ is the velocity of the moving quantity, and R is a source term providing for sources or sinks.

This transport equation is crucial to CFD software and even more so to the simulations carried out in this project. The additional equations required to solve RANS are merely adaptations of the transport equation to different parameters.

2.3.1 Mass Transport

Mass species transport is described by a modified form of the scalar transport equation [1]:

$$\frac{\partial T}{\partial t} = -\nabla \cdot (\mathbf{u}T) + \nabla^2 \left(\left(D_T + \frac{\mu_t}{Sc_t}\right) T\right) + R,$$

where $\nabla \cdot (\mathbf{u}T)$ and $\nabla^2 \left(\left(D_T + \frac{\mu_t}{Sc_t}\right) T\right)$ represent the convection and diffusion terms respectively, and the turbulent Schmidt number Sc_t is the ratio of momentum diffusivity and mass diffusivity. This modified from of the scalar transport equation above includes the effects of turbulence. Once the velocity field $\mathbf{u}$ and pressure field p are solved in each iteration, values of $\mathbf{u}$ are put into the scalar transport equation to solve for T. Here, we use T instead of ϕ because OpenFOAM uses T.

2.4 Turbulence Models

Turbulence models "close" the system in order to solve the RANS by adapting the scalar transport equation to turbulent quantities. The two most popular steady-state models are the $k - \omega$ and $k - \epsilon$ models, where k is the turbulent kinetic energy, ϵ is the turbulent

dissipation rate, and ω is the specific turbulent dissipation rate. Despite the prevalence of these models, they both have major deficiencies; the $k-\omega$ model is overly dependent on the free stream boundary conditions, while the $k-\epsilon$ model is not sensitive to adverse pressure gradients (when pressure increases in the direction of flow) [13]. To overcome these difficulties, more advanced models have been proposed based on the original $k-\omega$ and $k-\epsilon$ models.

2.4.1 The Realisable $k-\epsilon$ (RKE) Model

The RKE model [14, 16] retains the turbulent kinetic energy equation of the standard model and revises the dissipation equation, assuming that $\epsilon \approx \nu\overline{\omega_i\omega_i}$ at high Reynolds numbers (the Reynolds number is defined as the ratio of inertial to viscous forces, indicating the fluid's tendency toward turbulent flow), where ω_i is the vorticity, or curl of the velocity vector. This simplifies the standard $k-\epsilon$ model to the following two equations:

$$\begin{cases} \dfrac{\partial(\rho k)}{\partial t} + \dfrac{\partial(\rho U_j k)}{\partial x_j} = \dfrac{\partial}{\partial x_j}\left((\mu + \dfrac{\mu_t}{\sigma_k})\dfrac{\partial k}{\partial x_j}\right) + \tau_{ij}\dfrac{\partial U_i}{\partial x_j} - \rho\epsilon \\ \\ \frac{\partial(\rho\epsilon)}{\partial t} + \frac{\partial(\rho\epsilon U_j)}{\partial x_j} = (\mu + \frac{\mu_t}{\sigma_\epsilon})\nabla^2\epsilon + C_1 S\rho\epsilon - C_2\frac{\rho\epsilon^2}{k+\sqrt{\nu\epsilon}} \end{cases}$$

where

$$C_1 = \max[0.43, \eta/(\eta+5)],\ C_2 = 1.0,\ \sigma_k = 1.0, \text{ and } \sigma_\epsilon = 1.2.$$

This project relies exclusively on the RKE model, although future work could include validating the CFD software packages using other turbulence models, such as the $k-\omega$ Shear-Stress Transport (SST) model.

Chapter 3

Validation of OpenFOAM against Fluent and Experimental Data

In order to use CFD software as a prediction tool, researchers must first assess the software's accuracy. Accordingly, we used the results of a real wind tunnel experiment and compared these results against the simulated data from the software packages. The wind tunnel data used in this project came from an experiment conducted by Tokyo Polytechnic University (TPU). A link to a wepbage for the experiment can be found here: www.wind.arch.t-kougei.ac.jp/info_center/pollution/Isothermal_Flow.html. Figure 3.1 provides a diagram of the geometric model. The model consists of a wind inlet, a single block, and a small hole on the leeward side of the block continuously releasing C_2H_4 tracer gas, also referred to as pollutant.

3.1 Pre-Processing

3.1.1 Geometry and Meshing

The first task in simulating a case is setting up the geometry, which involves setting the correct physical dimensions and creating the general shape of the structures. The geometry is set up according to the specifications shown in Figure 3.1, where the dimensions of the solution space are $1m \times 1m \times 0.6m$. We used ANSYS DesignModeler to create our geometric model following the same setup. (See Figure C.1 in Appendix C for a visual of the ANSYS geometry.) The computational domain must be much larger than the block so that boundary effects can be ignored. Next, meshing discretizes the geometry, determining the points at which information is stored and equations are solved.

Meshing greatly affects the accuracy of the solution. While a finer mesh generally produces better results, having too many cells drastically increases computation time. A dimensionless wall value known as the y^+ value determines the near-wall cell length necessary for an accurate solution. The formula is given as

$$y^+ \equiv \frac{u_* y}{\nu},$$

where y is the distance to the nearest wall, u_* is the friction velocity at the nearest wall, and ν is the kinematic viscosity. Essentially, this metric requires a finer mesh near slower fluids. An example of how one might tailor the mesh for this purpose can be seen in Figure 3.2.

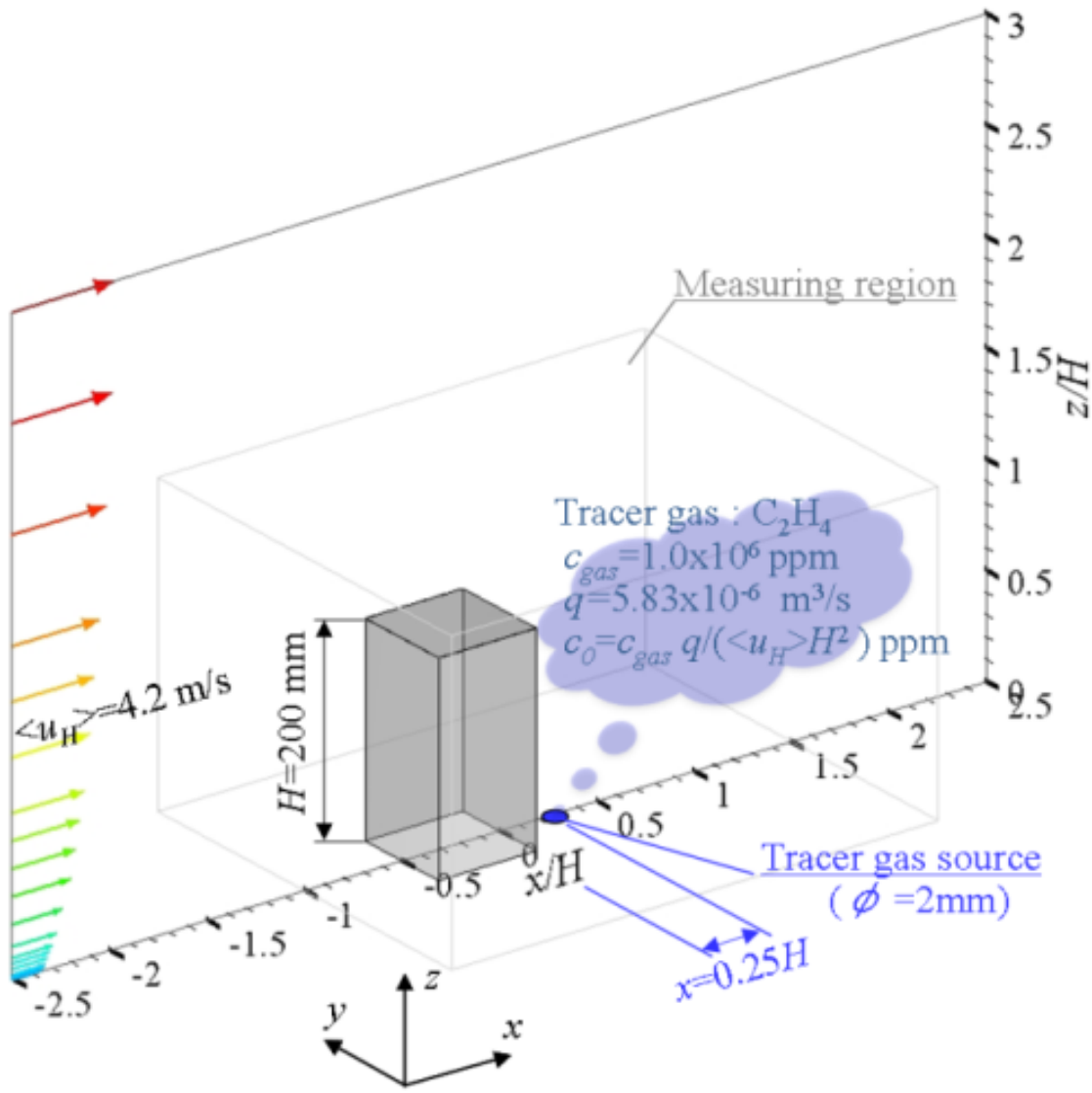

Figure 3.1: TPU Geometric Model [17]

We used the meshing software ANSYS ICEM to create our mesh, which consists of 454,484 hexahedral cells with a finer mesh in certain areas to obtain an optimal y^+ value. For the RKE turbulence model, a y^+ value below 30 is required [7]. It is important to note that the components needed to calculate the y^+ value can only be found after running the simulation, so the final mesh is the result of continuous refinement over many trials. Figure C.2 shows the y^+ of the final mesh. Lastly, an OpenFOAM utility, `fluentMeshToFoam`, makes the mesh readable by OpenFOAM software.

3.1.2 Boundary Conditions

In order to initialise the simulation and mimic physical conditions, a set of constraints must be imposed on the boundaries of the computational domain. These constraints are also necessary to obtain a unique solution. First, the inlet face is set as a velocity inlet, where the inlet profiles for velocity $\mathbf{u}$, turbulent kinetic energy k, and turbulent dissipation rate ϵ are set using curves fitted to the TPU experimental data. The shapes of these

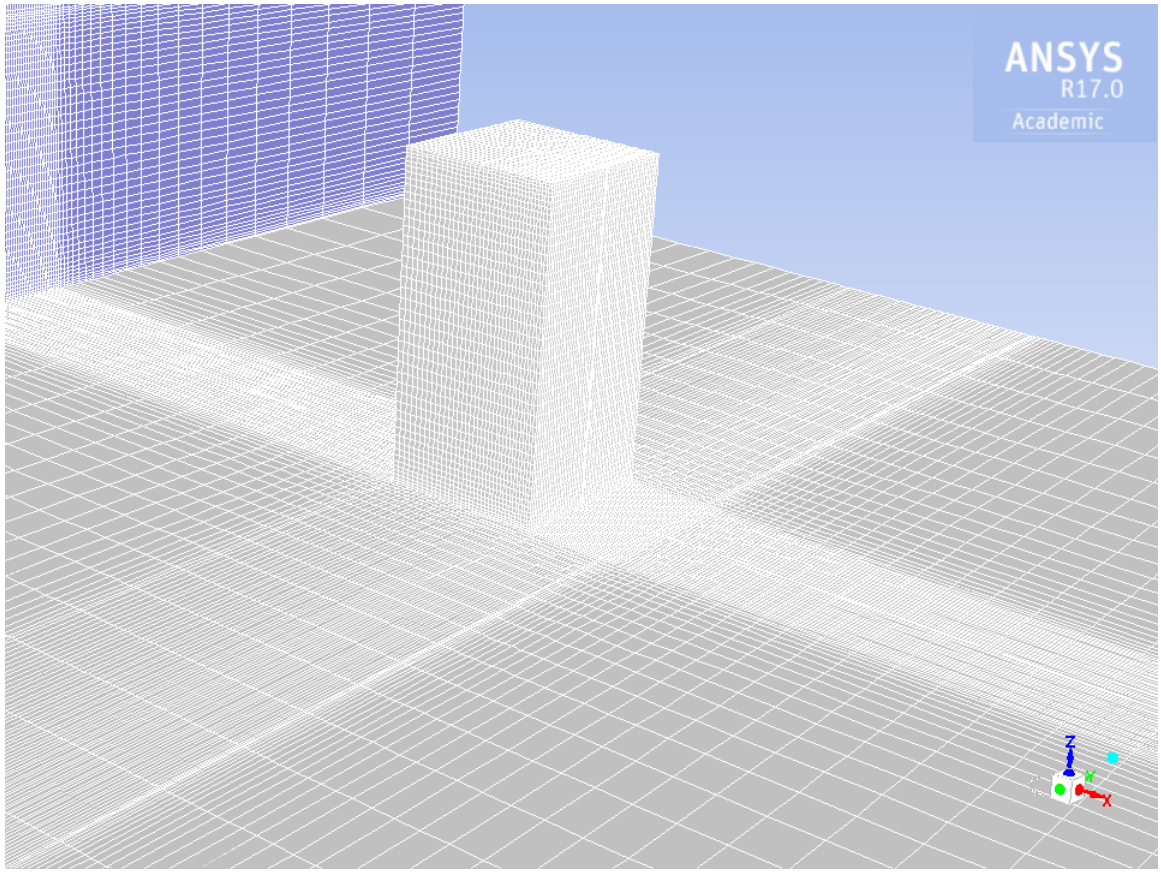

Figure 3.2: CFD Mesh

curves are an empirical effort to replicate real world conditions and are commonly used in CFD simulations [7]. For example, the velocity profile follows a power law, creating slower velocities near the ground where buildings and obstacles would impede movement. The other profiles on the inlet adhere to similar logic. For the exact formulae of the curves and fitting parameters, see Appendices A and B. Figures C.3, C.4, and C.5 show the profiles for $\mathbf{u}$, k, and ϵ, respectively. The remaining boundary conditions are outflow for the outlet, symmetry for the top and sides of the solution space, velocity inlet for the area source, and wall for all remaining surfaces.

3.2 Solving for Velocity and Pollutant Concentration

Both software packages utilise the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm to solve for velocity, pressure, and turbulent quantities. In regard to tracking the pollutant dispersion in OpenFOAM, we use a modified solver called `simpleScalarFoam`, which combines the SIMPLE solver `simpleFoam` with a passive scalar transport solver `scalarTransportFoam`.

However, some shortcomings of `simpleScalarFoam` must be noted. This solver treats the tracer as a passive scalar, so that C_2H_4 has no physical effect on the surrounding fluid. To be consistent, we also treated the pollutant as a passive scalar in Fluent. Additionally, for OpenFOAM, the scalar transport equation does not account for gravitational acceleration

of the tracer particles.

Before running the simulation, we must first select a number of different options in both Fluent and OpenFOAM. For a detailed description of the simulation setup of each software package, including near-wall functions, species transport options, and more, please refer to the file `tpu_case_file.cas` for Fluent and the folder `OpenFOAM` for OpenFOAM in the supplementary files.

3.3 Sampling the data

The TPU group measured wind velocity and pollutant concentration along four different planes, shown in Figure 3.3. Also shown in the figure are the various points where data was collected. Velocity and concentration were normalized in the same manner as the TPU data. We compared points from the planes using the same units for velocity and concentration.

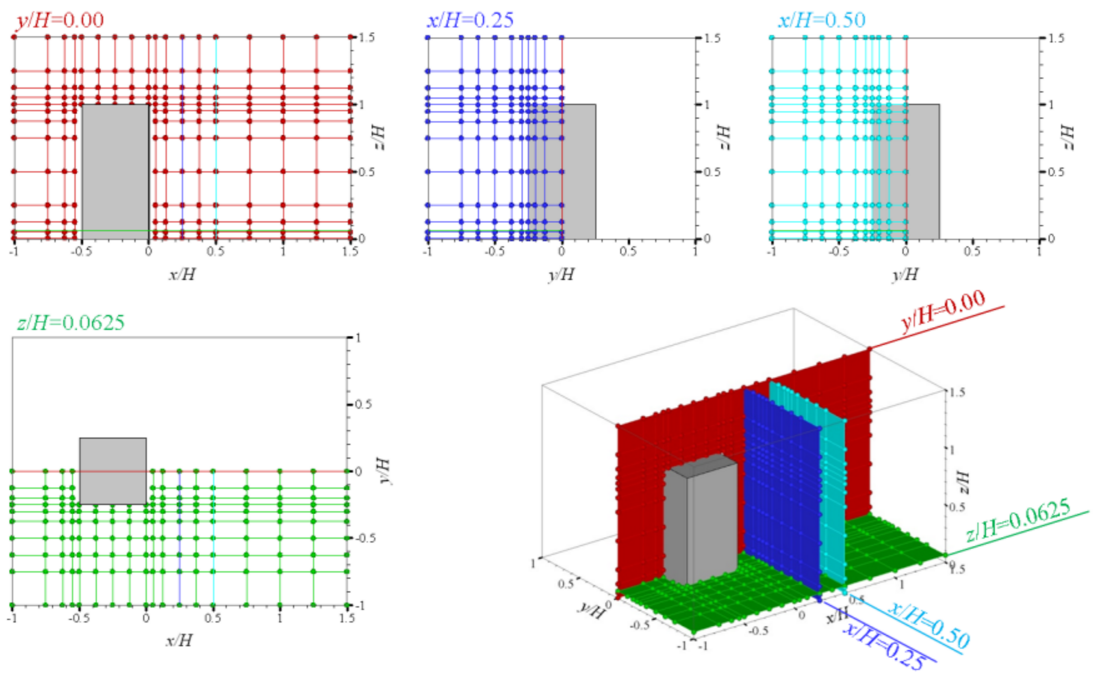

Figure 3.3: Sample Locations [17]

Chapter 4

Results

After sampling the data and comparing the velocity and pollutant distributions, we found that Fluent outperforms OpenFOAM in almost every case. In this section, we discuss our findings in more detail and employ various metrics to evaluate the performance of Fluent and OpenFOAM.

4.1 Simulation Results

4.1.1 Residuals

Figure 4.1 and 4.2 show how the residuals of $\mathbf{u}$, k, ϵ, and p change when simulations are run over extended time periods. Ideally, the curves should approach zero, indicating that the solution is properly converging. The Fluent data was extracted when the residuals were sufficiently small, close to 5,000 iterations. The oscillation of the OpenFOAM residuals in Figure 4.2 suggest that the solution may not be reliable. The OpenFOAM data was extracted after 400 time steps. T represents the pollutant mass stored in each cell.

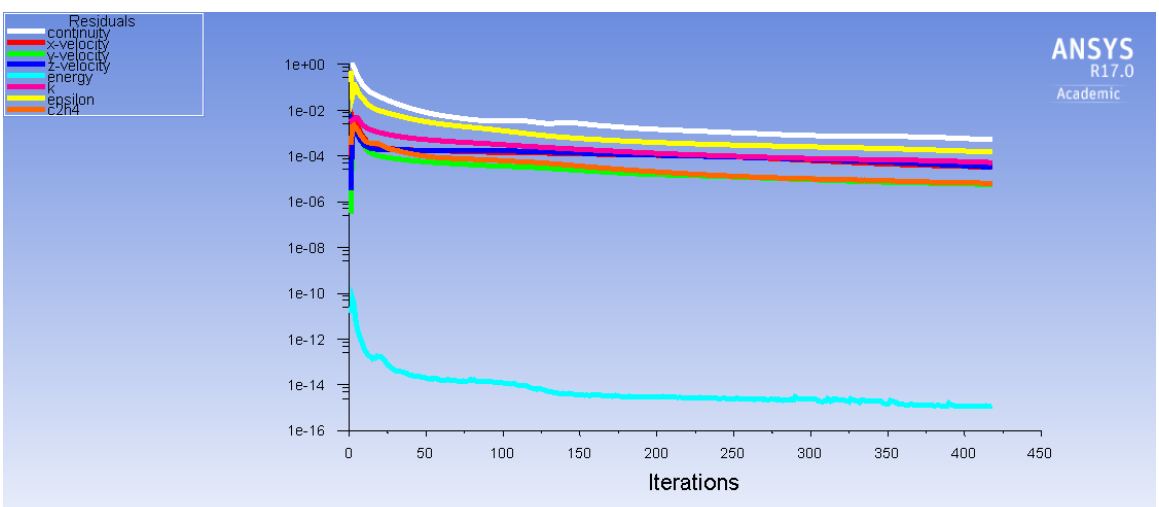

Figure 4.1: Fluent Residual Plot

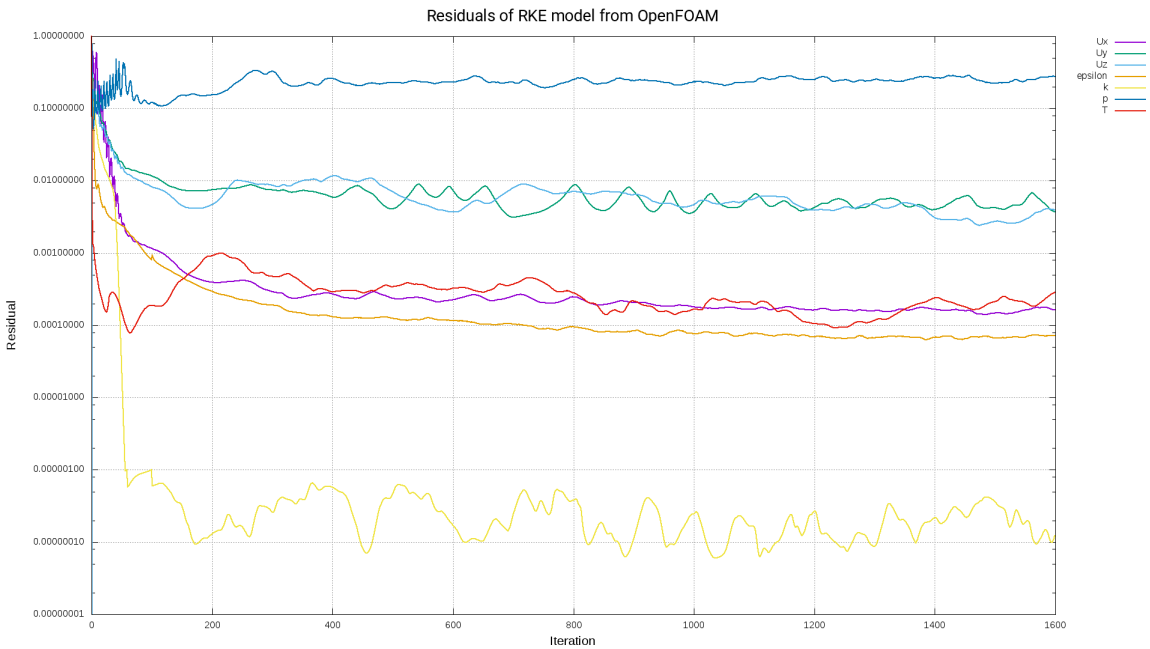

Figure 4.2: OpenFOAM Residual Plot

4.1.2 Simulated Velocity and Concentration Distributions

Figures 4.3 and 4.4 display contour plots of the wind-tunnel velocity and pollutant concentration results against the simulated Fluent and OpenFOAM results for the blue wall ($x = 0.05$m; see Figure 3.3 for further reference). When collecting data, we sampled points from 0 to 1 along the y-axis and from 0 to 1.5 along the z-axis. Similar axes were chosen for other walls in order to capture points in areas of interest. As was done in the TPU experiment, velocity was normalized by dividing by $\langle u_H \rangle = 4.2\ m/s$. Concentration, measured in mole fraction ppm, was also normalized by dividing by $c_0 = \frac{cq}{\langle u_H \rangle H^2}$, where $c = 10^6\ ppm$ and $q = 5.83 \times 10^{-6}\ m^3/s$. All statistics in the following sections use the normalized concentrations. For the concentration contour plots, we took the natural logarithm of the normalized concentration for visualization purposes only. Contour plots of the error are also shown, which we will discuss in Section 4.2. Finally, the lower right-hand scatter plots graph the data points from the TPU results against the corresponding data points from the CFD results so that a perfect fit would produce points only along the red line. The purple lines in these plots represent a 20% error range.

Similar results for the other planes can be found in Appendix C. Outliers are removed from all of the figures and from the data sets summarized in the tables of the following section.

4.2 Analysis

In comparing Fluent to the experimental data and OpenFOAM to the experimental data, we used three different statistical measures: mean absolute relative error, median absolute relative error, and the fraction of relative error less than 50%. Absolute relative error is defined as

$$\left| \frac{TPU - CFD}{TPU} \right|$$

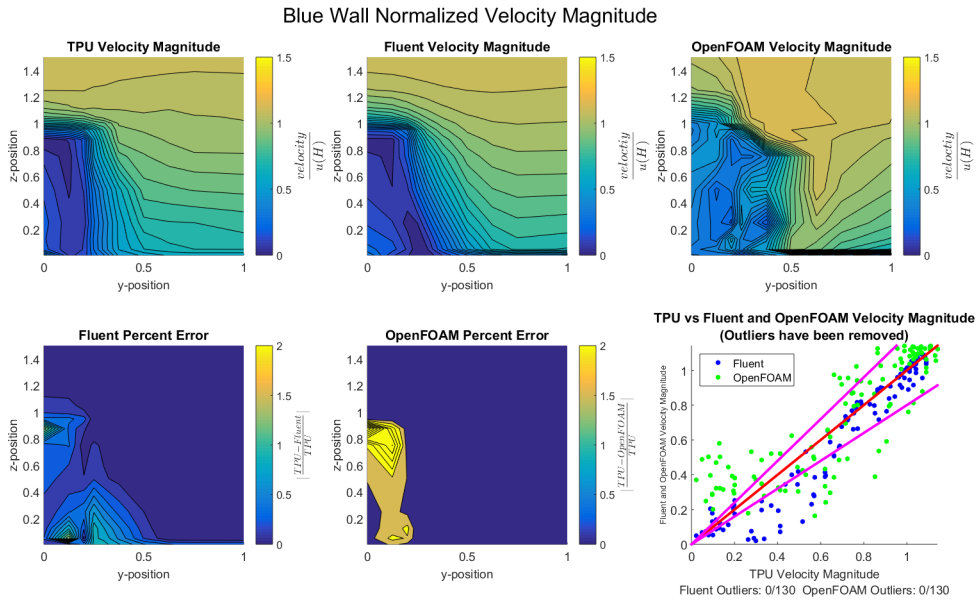

Figure 4.3: Example Velocity Plots

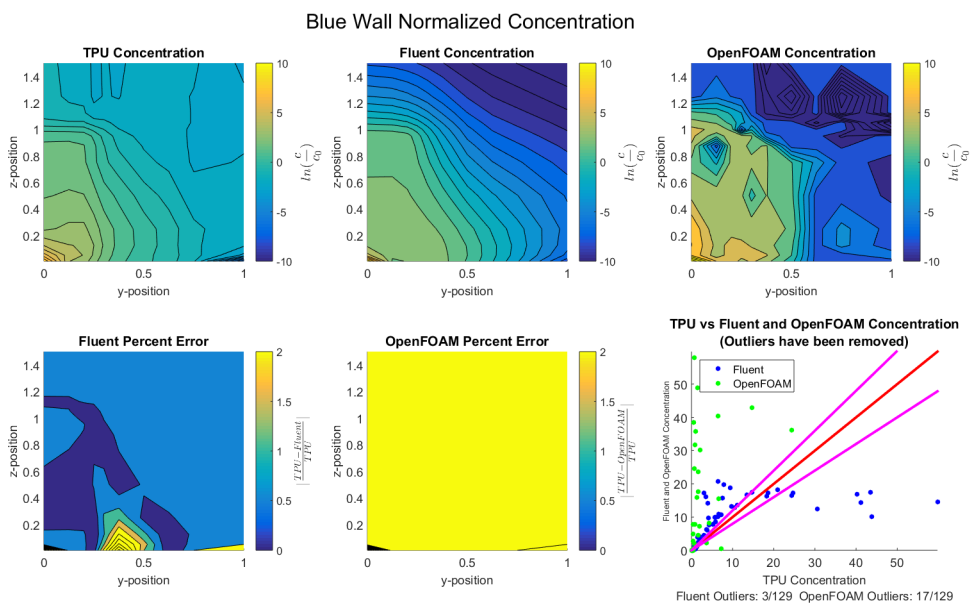

Figure 4.4: Example Concentration Plots

where the numbers being compared correspond to the same point. We then take the mean and median of the errors at all points. Likewise, we take the fraction of all the errors less than 50% for the third measurement. Table 4.5 and 4.6 summarize these metrics for each wall.

When analyzing our definition for error, it is important to note the effects of over-predicting and underpredicting. From Table 4.6, we see that OpenFOAM overpredicts concentration values for the blue wall. This is evident when we examine the definition of absolute relative error. The only way this value can be larger than 1 is if the variable CFD is two times larger than TPU. Additionally, OpenFOAM underpredicts concentration for the cyan wall. Again, referring to the error formula, as the CFD values approaches 0, the

error approaches 1. Therefore, error caused by underprediction is bounded by 1 while error caused by overprediction is unbounded. Thus, all error above 1 should be treated in the same way.

From Tables 4.5 and 4.6, we conclude that Fluent outperforms OpenFOAM in predicting velocity and concentration. Furthermore, the low error for velocity in Fluent justifies our choice of the RKE model and confirms the quality of our mesh; however, Fluent and OpenFOAM both have relatively large errors for concentration. The contour plots of error in Appendix C reveal the areas where Fluent and OpenFOAM differ the most from the TPU data. Most error comes from regions behind the block, confirming the theory that areas with slower, more turbulent flows are more difficult to model.

	Blue		Cyan		Green		Red	
	Fluent	FOAM	Fluent	FOAM	Fluent	FOAM	Fluent	FOAM
Percent Mean Absolute Error	**0.18**	0.62	**0.26**	0.52	**0.43**	0.64	**0.44**	1.17
Percent Median Absolute Error	**0.05**	0.16	**0.09**	0.18	0.38	0.38	**0.20**	0.36
Fraction of Error Less Than 50%	**0.89**	0.82	**0.86**	0.77	0.57	**0.65**	**0.76**	0.62

Table 4.5: Velocity Magnitude Error

	Blue		Cyan		Green		Red	
	Fluent	FOAM	Fluent	FOAM	Fluent	FOAM	Fluent	FOAM
Percent Mean Absolute Error	**0.82**	6.91	**0.62**	1.00	**0.75**	2.16	**0.53**	9.67
Percent Median Absolute Error	**0.85**	1.01	**0.65**	1.00	**0.56**	1.01	**0.53**	1.12
Fraction of Error Less Than 50%	**0.24**	0.06	**0.36**	0.00	**0.45**	0.12	**0.47**	0.10

Table 4.6: Concentration Error

4.3 Discussion

Various features of both Fluent and OpenFOAM may explain the discrepancies between CFD and TPU data. Any turbulence model makes assumptions, so no model is totally free

from inaccuracies. Moving forward, we could test different models, such as the $k - \omega$ SST model. Moreover, neither software package accounts for the physical effects of the tracer gas. As our software treats the situation, the pollutant does not react or change the distributions of velocity, viscosity, etc., whereas in real life, the emission source would certainly affect those physical properties of the gas around it. This problem may be better addressed by using a different solver (e.g. `interFoam` or a different multiphase solver), which could account for the fluids' different viscosity values. Our cases also do not take into consideration the roughness of the wind tunnel walls, treating all walls as perfectly smooth. A proper model for the roughness of the ground could improve the simulation, especially since the source of the tracer gas is located on the ground, where the wind velocity at the near wall regions would be greatly affected by the roughness. Finally, we would like to note that although our studies take the TPU data as the ideal case, the experiment may also be prone to error.

There are three main explanations for OpenFOAM simulation's poor concentration predictions. Firstly, it should be noted that, at each time step, the velocity equations are solved prior to solving the mass transport equation, where the solution for the velocity field obtained from the former is substituted into the latter in order to calculate the pollutant concentration. However, as shown in Table 4.5, Fluent already outperforms OpenFOAM in predicting the velocity magnitude, which might provide an explanation for the large errors as seen in Table 4.6. Another contributing factor to the concentration error is that the pollutant source is set up as a volume source in OpenFOAM, where no boundary condition for momentum or velocity is imposed. While implementing an area source coupled with a velocity inlet may better mimic the TPU experimental setup, we faced challenges in constructing this scenario in OpenFOAM. Relating to the multiphase issue, OpenFOAM might achieve a more accurate simulation if the kinematic viscosity at each point were calculated via the viscosity of constituents of the gas mixture by using a general equation [5]:

$$\mu_m = \frac{\mu_1}{1 + C\frac{x_2}{x_1}\frac{\mu_1}{\rho_1}} + \frac{\mu_2}{1 + C\frac{x_1}{x_2}\frac{\mu_2}{\rho_2}}.$$

The inlet profiles of $\mathbf{u}$, k and ϵ are closely fitted to the TPU data, so the remaining parameters that can be adjusted are the mass diffusivity, turbulent Schmidt number, roughness of the ground, kinematic viscosity, and the initial conditions for all variables. Theoretically, mass diffusivity should be constant with respect to a specific compound. However, since the system's physical properties are not fully described in our simplified model, different values for the mass diffusivity ought to be tested. From the results of previous cases, a closer value to the diffusivity of the tracer gas would yield a better result near the injection locations, while the concentration error would increase at the far side. We can also tune the turbulent Schmidt number Sc_t within a range of $[0.2, 0.9]$ to better fit the wind tunnel experiment, because Sc_t reflects the ratio of momentum diffusion (viscosity) and mass diffusion. This equates to finding a better eddy diffusivity value.

Chapter 5

Conclusion

Initial validation indicates that both software packages are able to make relatively accurate predictions for the velocity field, with Fluent moderately outperforming OpenFOAM. However, both software packages failed to give reliable results for the pollutant concentration. Most of the errors originate from the area behind the building, where turbulence is the dominant factor affecting the velocity distribution, which in turn, has a significant effect on the pollutant concentration. In order to obtain more reliable simulation results, further adjustments ought to be made by applying different boundary conditions, fine-tuning parameters, refining the mesh, implementing other turbulence models, and using different solvers.

Once OpenFOAM results reach a satisfactory degree of accuracy, the software may be used as a predictive tool to simulate real-life situations. Figure 5.1 shows an example geometry, generated by ANSYS software, which could later be tested on OpenFOAM.

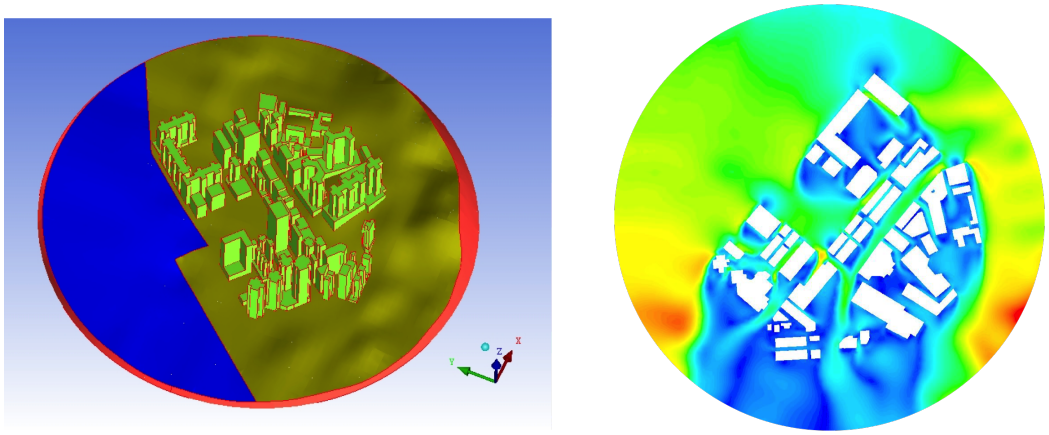

Figure 5.1: Example Simulation of Wind Velocity for Real-life Urban Environment

Appendix A

Constants

Air Constants:

$Density = 1.184\ kg \cdot m^{-3}$

$Specific\ Heat\ (C_p) = 1006.43\ J \cdot kg^{-1}K^{-1}$

$Thermal\ Conductivity = 0.0242\ W \cdot m^{-1}K^{-1}$

$Viscosity = 1.86 \times 10^{-5}\ kg \cdot m^{-1}s^{-1}$

$Molecular\ Weight = 28.966\ kg \cdot kmol^{-1}$

C_2H_4 Constants:

$Density = 1.178\ kg \cdot m^{-3}$

$Specific\ Heat\ (C_p) = 1548\ J \cdot kg^{-1}K^{-1}$

$Thermal\ Conductivity = 0.0174\ W \cdot m^{-1}K^{-1}$

$Viscosity = 9.47 \times 10^{-6}\ kg \cdot m^{-1}s^{-1}$

$Molecular\ Weight = 28.05\ kg \cdot kmol^{-1}$

$Diffusivity\ (D_T) = 1.28 \times 10^{-5}\ m^2s^{-1}$

Miscellaneous Constants:

$\alpha = 0.2624$

$\beta = 3.0733$

$c_1 = 1.6697$

$c_{gas} = 1 \times 10^6\ ppm$

$C_\mu = 0.09$

$H = 0.2\ m$

$\kappa = 0.4187$

$q = 5.83 \times 10^{-6}\ m^3s^{-1}$

$Temperature = 298\ K$

$Turbulent\ Schmidt\ Number\ (Sc_t) = 0.7$

$u_H = 4.2\ ms^{-1}$

$u_* = 0.3823$

$z_0 = 0.0025$

Appendix B

Formulae

Wind Inlet Profiles:

$$\epsilon(z) = \frac{u_*^3}{\kappa(z + z_0)}$$

$$\epsilon(k, z) = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{\kappa(z + z_0)}$$

$$k(z) = c_1 z^\alpha e^{-\beta z}$$

$$\omega(k) = C_\mu^{-\frac{1}{4}} k^{\frac{1}{2}}$$

$$\omega(k, z) = C_\mu^{-\frac{1}{4}} \frac{k^{\frac{1}{2}}}{\kappa(z + z_0)}$$

$$u(z) = u_H \left(\frac{z}{H}\right)^\alpha$$

Other Formulae:

$$c_0 = \frac{c_{gas} q}{u(H) H^2}$$

$$y^+ = \frac{u_* y}{\nu}$$

Appendix C

Additional Figures

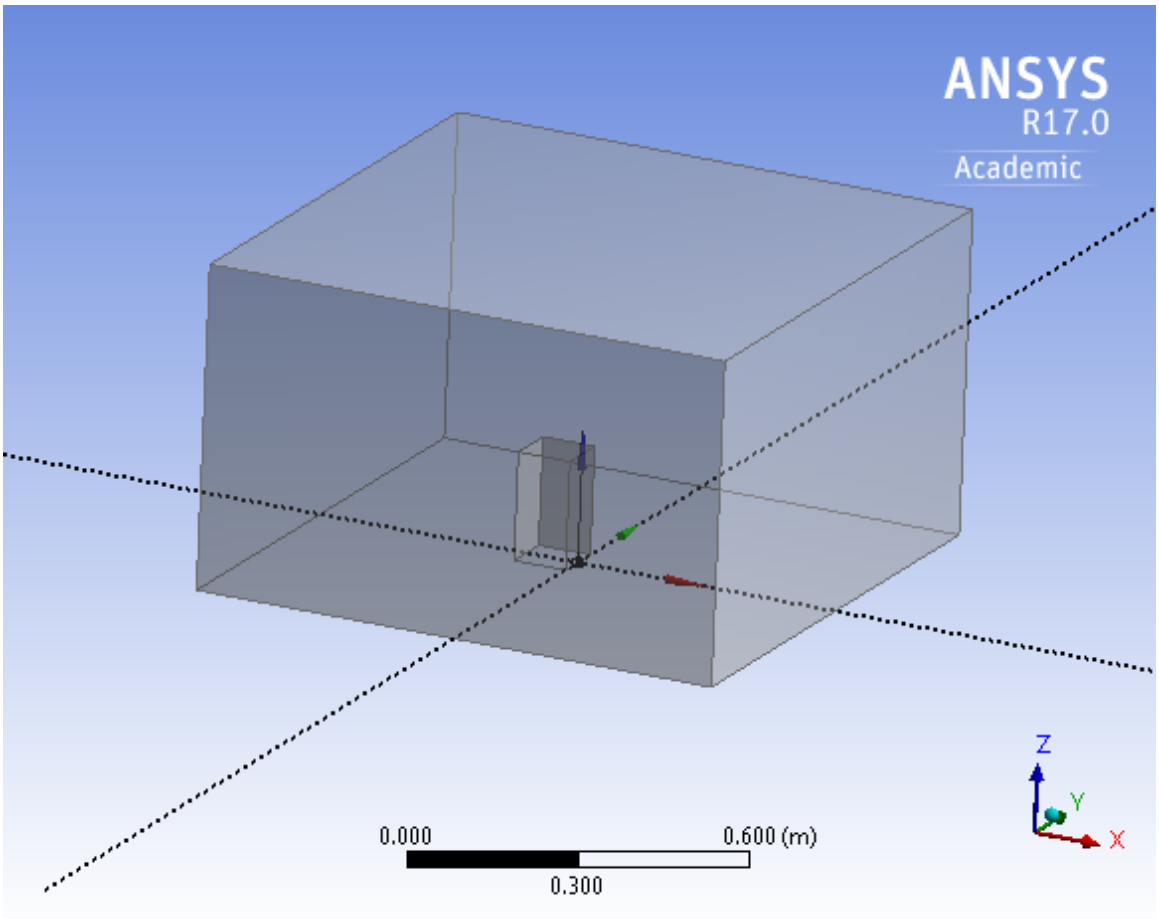

Figure C.1: CFD Geometric Model

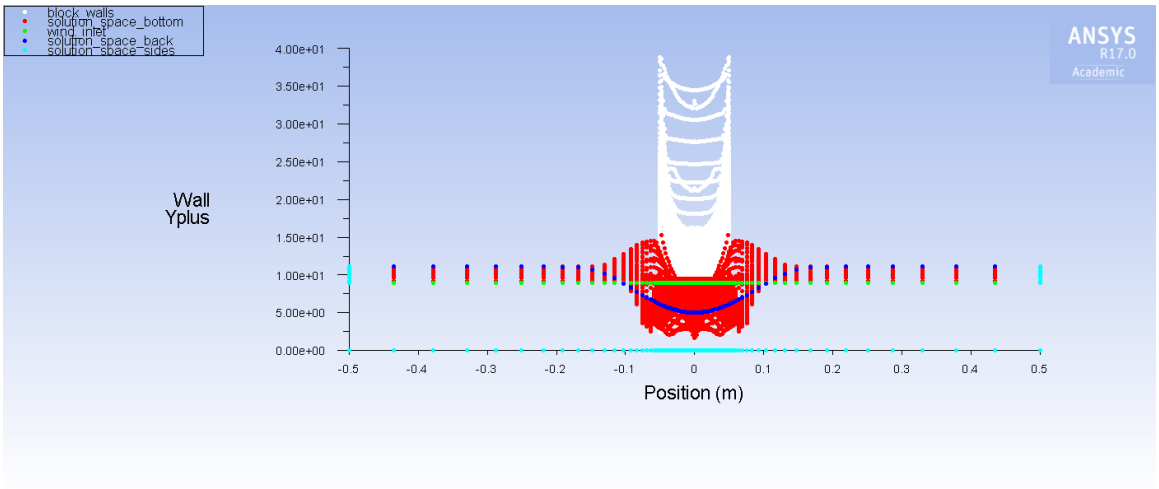

Figure C.2: Wall y^+ Values (Position denotes the y-axis)

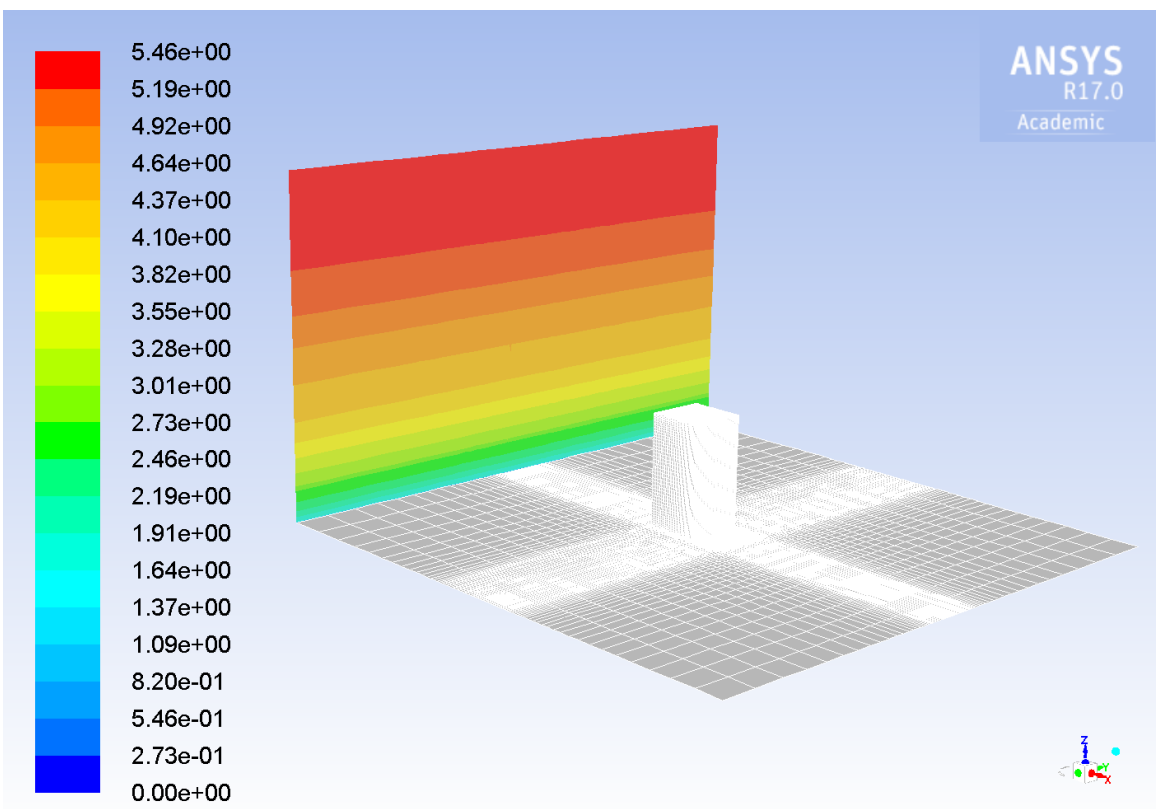

Figure C.3: Inlet Velocity Profile

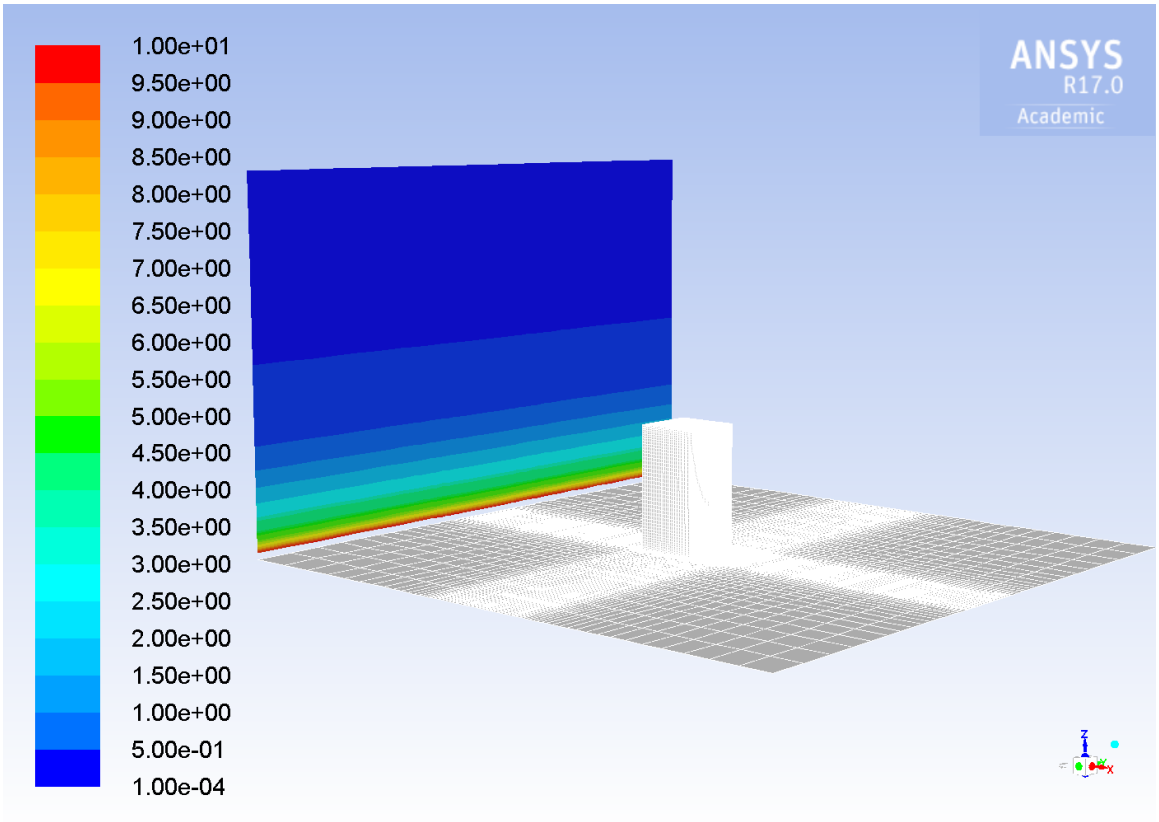

Figure C.4: Inlet TKE (k) Profile

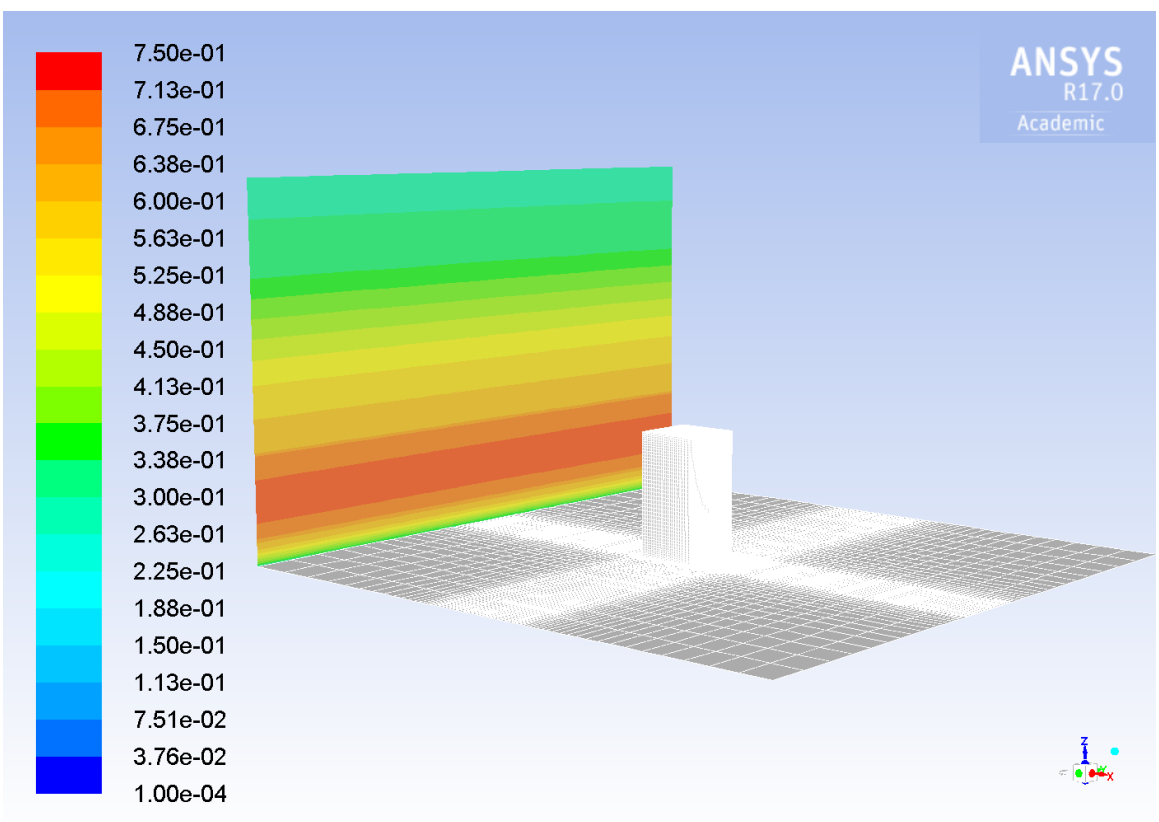

Figure C.5: Inlet TKE Dissipation Rate (ϵ) Profile

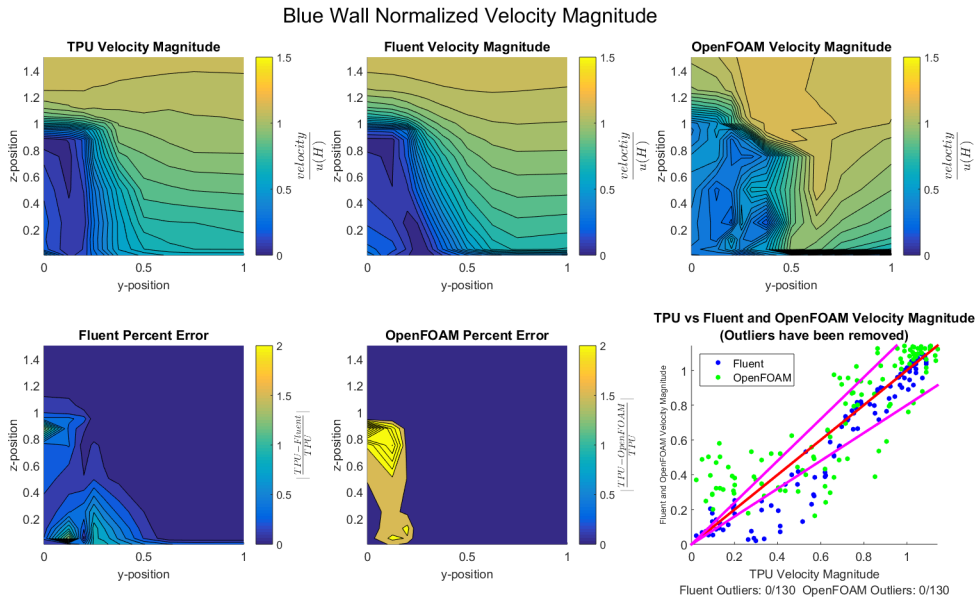

Figure C.6: Blue Wall Normalized Velocity Magnitude

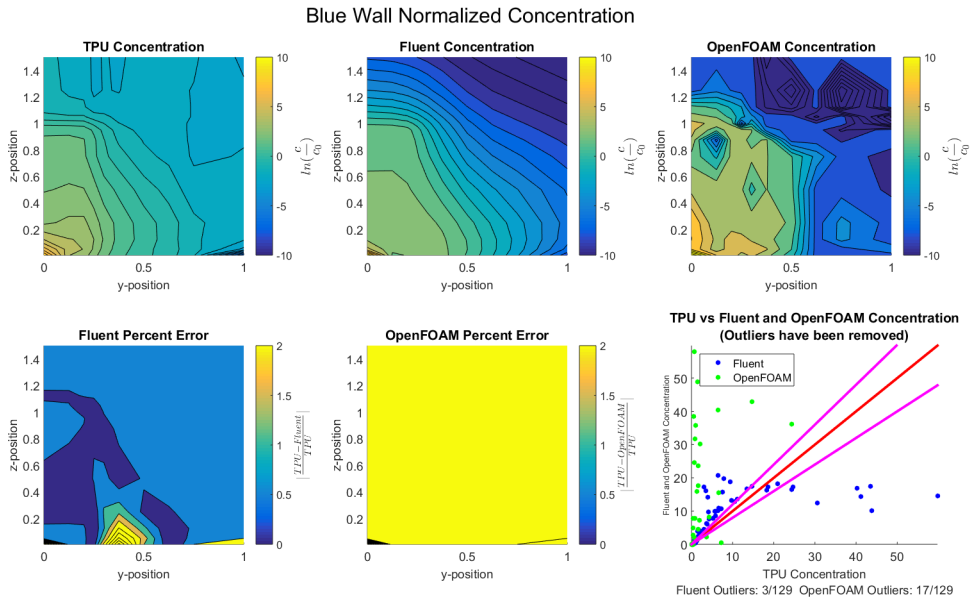

Figure C.7: Blue Wall Normalized Concentration

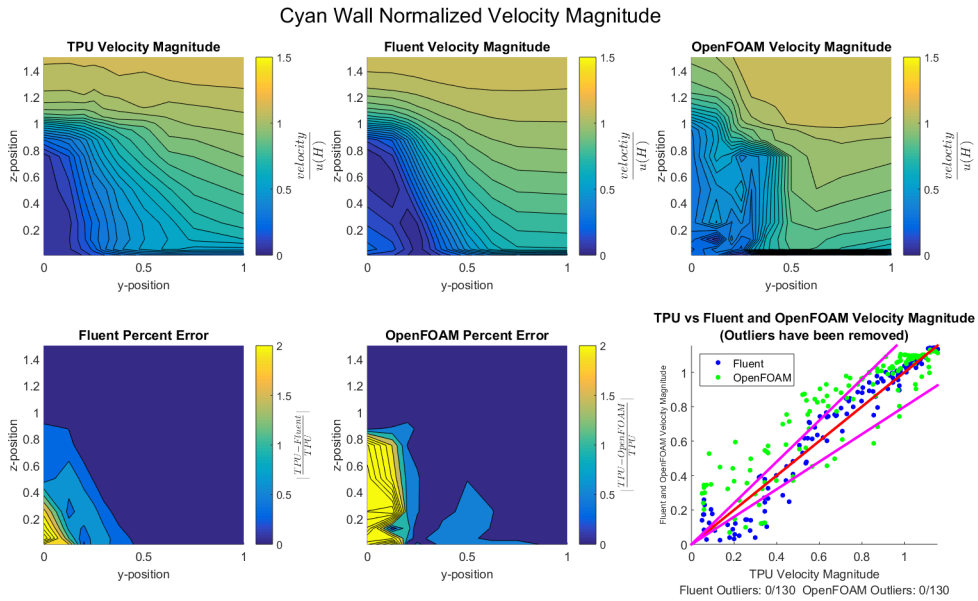

Figure C.8: Cyan Wall Normalized Velocity Magnitude

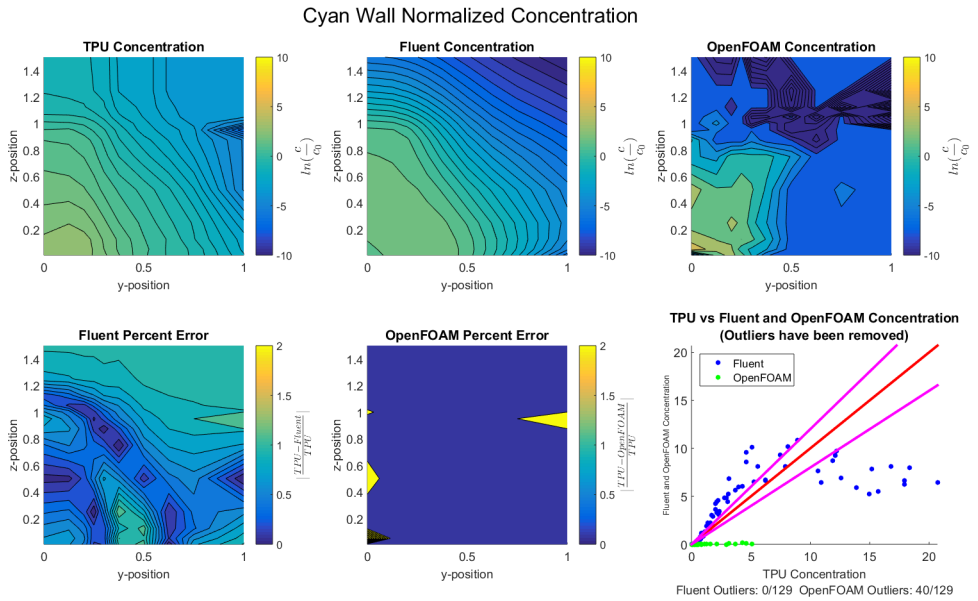

Figure C.9: Cyan Wall Normalized Concentration

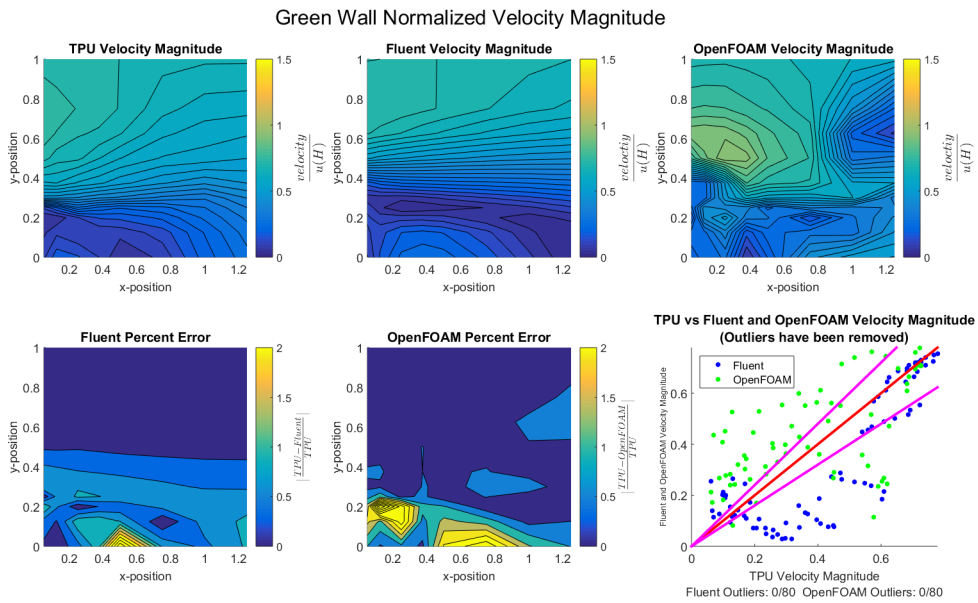

Figure C.10: Green Wall Normalized Velocity Magnitude

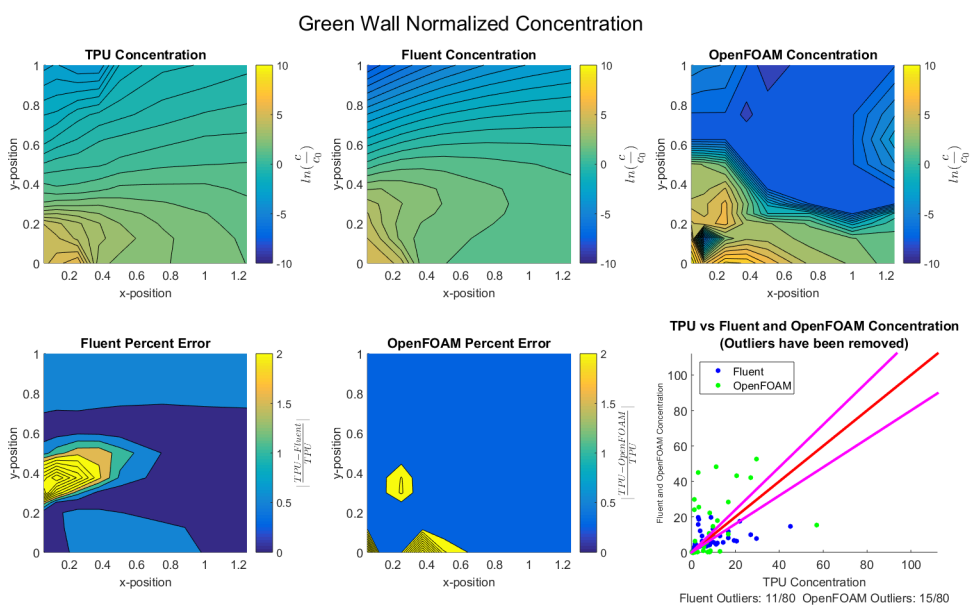

Figure C.11: Green Wall Normalized Concentration

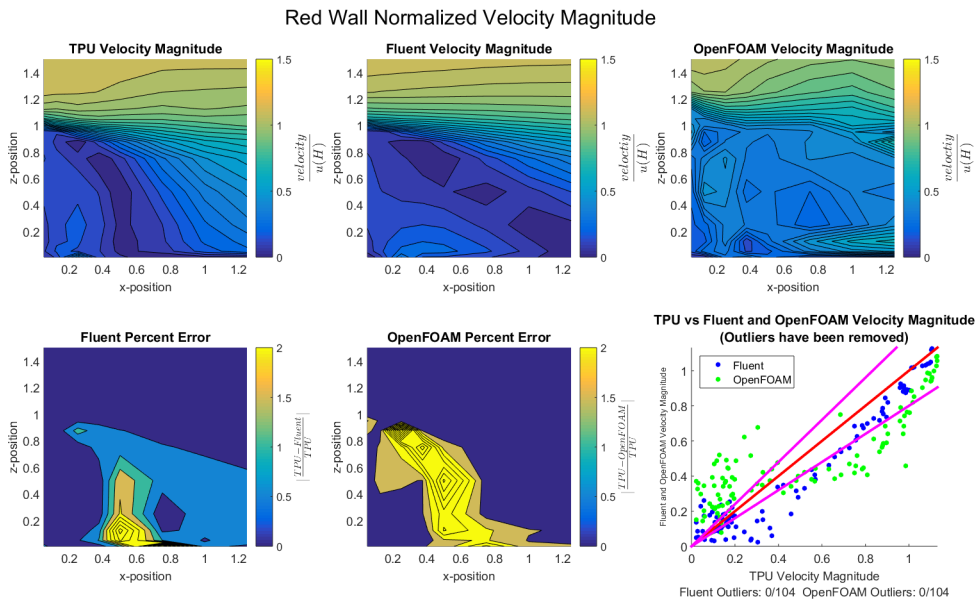

Figure C.12: Red Wall Normalized Velocity Magnitude

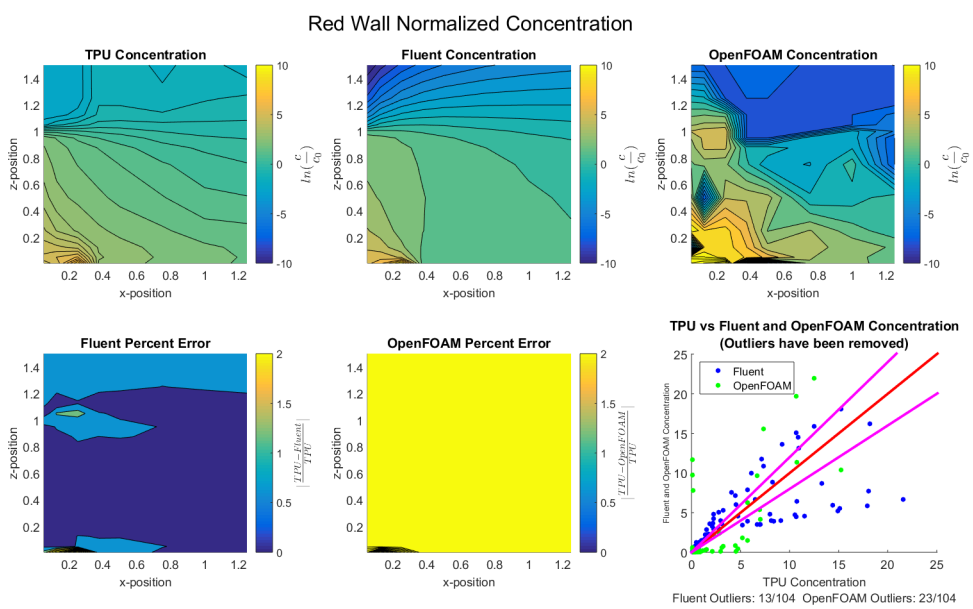

Figure C.13: Red Wall Normalized Concentration

Bibliography

[1] ***Contrib simplescalarfoam.*** "https://openfoamwiki.net/index.php/Contrib_simpleScalarFoam", **2009.**

[2] J. e. a. Bardina, ***Turbulence modeling validation, testing, and development, nasa technical memorandom 110446.*** "http://www.ewp.rpi.edu/hartford/~ferraj7/ET/Other/References/nasa_techmemo_110446.pdf", **1997.**

[3] K. Becker, ***Numerical flow simulation for the airbus a380.*** "https://www.researchgate.net/profile/Hans_Josef_Pesch/publication/268609764/figure/fig7/AS:295370262368273@1447433166910/Figure-9-Numerical-flow-simulation-for-the-Airbus-A380-picture-credit-Airbus.png", **2012.**

[4] R. e. a. Buccolieri, ***Analysis of local scale tree-atmosphere interaction on pollutant concentration in idealized street canyons and application to a real urban junction***, **Atmospheric Environment, 45 (2011), pp. 1702–1713.**

[5] J. BUDDENBERG and C. WILKE, ***Calculation of gas mixture viscosities.*** "http://pubs.acs.org/doi/pdf/10.1021/ie50475a011", **1949.**

[6] I. Celik, ***Introductory turbulence modeling, west virginia university class notes, western university press.*** "http://www.fem.unicamp.br/~im450/palestras%26artigos/ASME_Tubulence/cds13workbook.pdf", **1999.**

[7] H. Chanson, ***Applied Hydrodynamics: An Introduction to Ideal and Real Fluid Flows***, **CRC Press, Taylor & Francis Group, 2009.**

[8] C. Gromke, ***A vegetation modeling concept for building and environmental aerodynamics wind tunnel tests and its application in pollutant dispersion studies***, **Environmental Pollution, 159 (2011), p. 20942099.**

[9] C. e. a. Gromke, ***Dispersion study in a street canyon with tree planting by means of wind tunnel and numerical investigations evaluation of cfd data with experimental data***, **Atmospheric Environment, 42 (2008), p. 86408650.**

[10] R. Hora, ***Q&a: Hong kongs air-pollution problem.*** "http://graphics8.nytimes.com/images/blogs/greeninc/hongpollution.jpg", **jan 2010.**

[11] LAEROMEC. "static.wixstatic.com/media/b8ae24_c36de2b7d1c64ef4a7b71a8299d1ef0a.png_srz_1903_1516_85_22_0.50_1.20_0.00_png_srz", **2015.**

[12] D. Lafforgue, ***Sails: from experimental to numerical.*** "http://www.finot.com/ecrits/Damien%20Lafforgue/article_voiles_english_fichiers/averaging.bmp", **2007.**

[13] F. R. Menter, ***Improved two-equation k-ω turbulence models for aerodynamic flows, nasa technical memorandum 103975.*** "http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930013620.pdf", **1992.**

[14] B. Rohdin, P.; Moshfegh, ***Numerical predictions of indoor climate in large industrial premises. a comparison between different $k-\epsilon$ models supported by field measurements***, **Building and Environment, 42 (2007), pp. 3872–3882.**

[15] C. Ruck, B.; Gromke, ***Codasc laboratory of building- and environmental aerodynamics karlsruhe institute of technology kit.*** "http://www.windforschung.de/CODASC.htm", **apr 2016.**

[16] T. e. a. Shih, ***A new $k-\epsilon$ eddy viscosity model for high reynolds number turbulent flows***, **Computers and Fluids, 24 (1995), pp. 227–238.**

[17] R. Yoshie, ***Wind effects on buildings and urban environment, tokyo polytechnic university.*** "http://www.wind.arch.t-kougei.ac.jp/info_center/pollution/Isothermal_Flow.html", **jun 2006.**