**National Technical University of Athens**
**School of Mechanical Engineering**
**Fluids Section**
**Laboratory for Innovative Environmental Technologies**

# Application of Data Assimilation Methods in Computational Fluid Dynamics

Diploma Thesis

**Konstantinos Kellaris**

Supervisor: Demetri Bouris, Professor NTUA

Athens, October 2022

# Acknowledgements

Firstly, I would like to thank my Thesis supervisor, Professor Demetri Bouris of the School of Mechanical Engineering, Fluids Section at the National Technical University of Athens, for assigning me this topic, providing insightful remarks, and allowing this work to be my own but steering me back in the right direction when needed.

Furthermore, I would like to thank Ph.D. candidate Nikos Pallas for his remarks about the Data Assimilation methods used in this Thesis and fruitful discussions about the pressure calculation from PIV data. Additionally, I would like to thank Ph.D. candidate Maria Kotsiopoulou for her remarks and the great atmosphere in the lab.

Last but not least, I would like to thank my family for always being there and supporting me with their love, as well as my friends for always being there for me.

# Περίληψη

Η Αφομοίωση Δεδομένων (ΑΔ) είναι το επιστημονικό πεδίο που αφορά τον βέλτιστο συνδυασμό δυναμικών μοντέλων και πειραματικών δεδομένων, με στόχο την βελτίωση των αποτελεσμάτων των πρώτων, λαμβάνοντας υπόψη τις αβεβαιότητες και των δύο. Στην παρούσα Διπλωματική Εργασία, η χρήση μεθόδων ΑΔ έγινε για την βελτίωση των αποτελεσμάτων σε προσομοιώσεις Υπολογιστικής Ρευστοδυναμικής (ΥΡ), σε δύο διαφορετικές εφαρμογές.

Αρχικά, δημιουργήθηκε σε Fortran 95 παράλληλος (μέσω του OpenMP) κώδικας επίλυσης των μη-μόνιμων 1-Δ εξισώσεων Euler για συμπιεστό ρευστό και μελετήθηκε η ροή μέσα σε κρουστικό σωλήνα, ενώ το λογισμικό πιστοποιήθηκε μέσω της αναλυτικής λύσης του προβλήματος. Στη συνέχεια, εφαρμόστηκε η μέθοδος nudging, για την βελτίωση των αριθμητικών αποτελεσμάτων μέσω της αναλυτικής λύσης. Έπειτα, έγινε χρήση της μεθόδου nudging για τον συνδυασμό αριθμητικών λύσεων σε αραιό και πυκνό πλέγμα, οδηγώντας σε σημαντική βελτίωση των αποτελεσμάτων σε σχέση με ακόμη πυκνότερα πλέγματα. Τέλος, το μητρώο κερδών της μεθόδου nudging βελτιστοποιήθηκε ως προς την ελαχιστοποίηση του σφάλματος υπολογισμού για ένα χρονικό διάστημα, επιλύοντας τις συζυγείς μη-μόνιμες 1-Δ εξισώσεις Euler για τον προσδιορισμό των παραγώγων ευαισθησίας.

Στο δεύτερο σκέλος, υλοποιήθηκε στο λογισμικό ΥΡ OpenFOAM, μια μέθοδος υπολογισμού πεδίων πίεσης από πεδία ταχύτητας, τα οποία είναι γνωστά από πειράματα μέσω PIV για τη μόνιμη τυρβώδη 2-Δ ροή γύρω από κύλινδρο τετραγωνικής διατομής που βρίσκεται εντός ατμοσφαιρικού οριακού στρώματος. Η μέθοδος κάνει χρήση του αλγορίθμου SIMPLE για τη διόρθωση των πεδίων πίεσης και ταχύτητας ενώ έγινε σύγκριση της και με την επίλυση μια εξίσωσης Poisson για τα πεδία πίεσης. Αρχικά, έγινε πιστοποίηση της μεθόδου ως προς την ικανότητα υπολογισμού πεδίων πίεσης και συμβατότητας με την εξίσωση συνέχειας. Έπειτα, μελετήθηκε η επίδραση της πύκνωσης του υπολογιστικού πλέγματος. Τέλος, υλοποιήθηκαν διαφορετικές μεθοδολογίες ΑΔ, βασισμένες στη μέθοδο nudging και στη μέθοδο του φίλτρου Kalman. Εξετάστηκαν διαφορετικές διατυπώσεις των μεθόδων και έγινε σύγκριση ως προς την ακρίβεια και βελτίωση που επιφέρουν στις τελικές λυσείς, με τη μέθοδο του φίλτρου Kalman να εμφανίζει την καλύτερη συμπεριφορά.

### Λέξεις Κλειδιά

Υπολογιστική Ρευστοδυναμική, Αφομοίωση Δεδομένων, Τυρβώδης Ροή, Φίλτρο Kalman, OpenFOAM

# Abstract

Data Assimilation (DA) is a mathematical discipline that seeks to optimally combine dynamic models with experimental observations in order to improve the results of the former while accounting for uncertainties in both the model and observation data. In this Diploma Thesis, DA methods were applied to improve the results obtained from Computational Fluid Dynamics (CFD) simulations in two applications.

Firstly, a Fortran 95 parallel (using OpenMP) computational code for the solution of the transient 1-D compressible Euler equations was developed, and a Shock Tube flow was investigated. The code was verified using the analytical solution of the problem. Subsequently, the nudging method was implemented to improve the numerical results taking into consideration the analytical solution. The nudging method was then used to combine the numerical solution from a coarse and a fine computational grid, leading to a considerable improvement compared to numerical results from even finer grids. Finally, the nudging gain matrix was optimized to minimize the computational error over a time interval. The adjoint transient 1-D Euler equations were solved to determine the sensitivity derivatives during the optimization loop.

In the next part of the Thesis, an iterative method was implemented in the OpenFOAM CFD toolbox to calculate pressure fields using velocity fields provided by PIV measurements. The method was also applied to the steady-state 2-D turbulent flow around a wall-mounted cube inside the atmospheric boundary layer. The method utilizes the SIMPLE algorithm to correct the pressure and velocity fields and was compared with the solution of a Poisson equation for the determination of the pressure fields. Initially, the method's ability to calculate pressure and velocity fields that obey the continuity equation was verified. Then, the effect of mesh refinement on the results was investigated. Finally, various DA methods were implemented based on nudging and the Kalman filter. Different implementations were analyzed and compared in terms of their accuracy and improvement to the final solutions, with the Kalman filter method prevailing.

### Keywords

Computational Fluid Dynamics, Data Assimilation, Turbulent Flow, Kalman Filter, OpenFOAM

**National Technical University of Athens**
**School of Mechanical Engineering**
**Fluids Section**
**Laboratory for Innovative Environmental Technologies**

Υπευθυνη Δηλωση Για Λογοκλοπη Και Για Κλοπη Πνευματικης Ιδιοκτησιας

Έχω διαβάσει και κατανοήσει τους κανόνες για τη λογοκλοπή και τον τρόπο σωστής αναφοράς των πηγών που περιέχονται στον οδηγό συγγραφής Διπλωματικών Εργασιών. Δηλώνω ότι, από όσα γνωρίζω, το περιεχόμενο της παρούσας Διπλωματικής Εργασίας είναι προϊόν δικής μου εργασίας και υπάρχουν αναφορές σε όλες τις πηγές που χρησιμοποίησα.
Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή τη Διπλωματική εργασία είναι του συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις της Σχολής Μηχανολόγων Μηχανικών ή του Εθνικού Μετσόβιου Πολυτεχνείου.

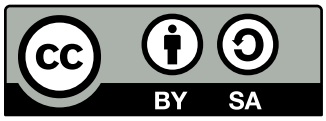
**Κωνσταντίνος Κελλάρης, Αθήνα 2022**

Statement About Plagiarism And Intellectual Property Theft

I have read and understood the rules about plagiarism and the proper way of refrencing constained in the Diploma Thesis writing guide. I declare that, to the best of my knowldege, the content of this Diploma Thesis is the product of my own work and there are refrences to all sources that I have used.
The opinions and conclusions contained in this Diploma Thesis are of the author and should not be interpeted as representing the official views of the School of Mechanical Engineering or the National Technical University of Athens.

**Konstantinos Kellaris, Athens 2022**





# List of Abbreviations

**API** Application Programming Interface

**BLUE** Best Linear Unbiased Estimator

**CFD** Computational Fluid Dynamics

**CFL** Courant–Friedrichs–Lewy

**DA** Data Assimilation

**DNS** Direct Numerical Simulation

**EFD** Experimental Fluid Dynamics

**EnKF** Ensemble Kalman Filter

**FVM** Finite Volume Method

**KF** Kalman Filter

**LES** Large Eddy Simulation

**MPI** Message Passing Interface

**MUSCL** Monotonic Upstream-centered Scheme for Conservation Laws

**NS** Navier-Stokes

**PDEs** Partial Differential Equations

**PIV** Particle Image Velocimetry

**RANS** Reynolds Averaged Navier-Stokes

**RMSE** Root Mean Square Error

**SIMPLE** Semi-Implicit Method for Pressure-Linked Equations

**TVD** Total Variation Diminishing

# List of Symbols

| | |
|---|---|
| $\gamma$ | Heat capacity ratio |
| $\mu$ | Dynamic viscocity |
| $\nu$ | Kinematic viscocity |
| $\rho$ | Density |
| E | Specific energy |
| H | Specific enthalpy |
| $c$ | Speed of sound |
| $\mathbf{U}$ | Conservative variables vector |
| $\mathbf{F}$ | Fluxes vector |
| $\mathbf{W}$ | Primitive variables vector |
| $\Phi$ | Flux limiter function |
| $\mathbf{\Psi}$ | Adjoint variables vector |
| $\mathbf{u}$ | Velocity |
| $\mathbf{u}_{\mathrm{PIV}}$ | Velocity from PIV measurements |
| $p$ | Pressure |
| $\mathbf{K}$ | Kalman Filter gain matrix |

# Contents

# Chapter 1

# Introduction

In this chapter, the motivation behind the use of Data Assimilation methods in Computational Fluid Dynamics is presented. Then, a small literature review of the application of these methods to Computational Fluid Dynamics problems is performed. Finally, the contents of the present Thesis are outlined.

## 1.1 Motivation - Problem Statement

DA (*Data Assimilation*) is a mathematical discipline that seeks to optimally combine dynamic models with experimental observations in order to improve the results of the former while accounting for the uncertainties of both the model and observation data. DA is classically approached in two ways, as variational DA and as statistical DA. In both approaches, an optimal solution is sought; variationally, a solution that minimizes a suitable cost (or error) function is sought, whereas statistically, a solution with minimum variance is sought. Although, in certain special cases, the two approaches are identical and provide the same solution. However, the statistical approach, though often more complex and time-consuming, can offer an enriched information structure that contains an average solution and some characteristics of its variability (probability distribution). Nowadays, DA approaches have been extensively utilized in geophysics, meteorology, oceanography, and other fields, and several key techniques have been developed.

It is probably fair to say that EFD (*Experimental Fluid Dynamics*) and CFD (*Computational Fluid Dynamics*) have been independently developed for a long time. Some interactive efforts have been made, for example, to design EFD with aid of CFD, to correct EFD results based on CFD or to validate/calibrate CFD models using EFD. However, more attention has been paid to coupling EFD and CFD to overcome their weaknesses in the last several years.

Measurements, and measurement devices, can characterize real-world flows, but they are often inadequate due to their limitation in time and space, their inaccuracy, and the fact that a measurement instrument can actually perturb the flow. Unlike existing EFD approaches that rely on high-end measurement devices and high-order post-processing schemes to deal with these shortcomings, the dynamical models are augmented in DA. Compared with conventional CFD approaches, uncertainties are reduced when using DA by directly employing measured data instead of pursuing the completeness of modeling and initial and/or boundary conditions.

The scope of this Thesis is to apply DA techniques in CFD problems. Two applications are studied and validated using a variety of methods and incorporating synthetic (analytical solution) and experimental data.

## 1.2 Literature Review

In this section, a brief description of DA methods is presented and references to recent applications of them in CFD problems are provided. Three main categories of DA methods exist [1]:

- Variational methods
- Statistical methods
- Hybrid methods

and all of them have been applied to CFD problems [2].

Variational DA emanates from the optimal control theory and seeks to determine the state of a dynamical system that optimizes a cost function that combines the state as computed by the dynamical model, the observations, and both the model and observation errors and uncertainties. These methods can be applied to both transient and steady-state problems, and when applied to the former, they are usually coupled with the adjoint approach for the computation of the sensitivity variables of the corresponding function and a simple optimization method, e.g., the steepest descent method, for updating the state during the optimization loop.

For example, in [3], a variational formulation and a Lagrange multipliers approach were applied to reconstruct the mean flow field of the 2-D flow around an infinite cylinder at a supercritical Reynolds number of Re = 150 using DNS (*Direct Numerical Simulation*) data as observations. Similarly, in [4] data assimilation of the 2-D mean flow around an idealized airfoil Re = 13, 500 obtained from time-averaged PIV (*Particle Image Velocimetry*) data is performed by solving the NS (*Navier-Stokes*) equations with an additional forcing term calculated from a direct-adjoint optimization procedure.

On the other hand, statistical or sequential DA methods emanate from statistical estimation theory and Bayes' law. They usually consist of a prediction step, solving the dynamical model, and a correction step, correcting the state considering the observation data. They seek to determine an optimal a posteriori estimate of the system state, that is, a linear combination of an a priori estimate and a weighted difference between the observation data and the observation prediction from the dynamical model.

The most widespread sequential DA method is the KF (*Kalman Filter*) [5], which is based on the minimization of the error variance. In [6] a Kalman filter-based sequential estimator is implemented in a way that is integrated into the segregated solver's structure to analyze incompressible flows. This technique provides an augmented flow state integrating available observation in the CFD model, naturally preserving a zero-divergence condition for the velocity field. Using synthetic data, the method is applied in 2-D/3-D laminar flows and 2-D turbulent flows. Additionally, in [7] the Kalman filter is applied to the RANS (*Reynolds Averaged Navier-Stokes*) equations for turbulent flow. The integration of the Kalman estimator is extended to an implicit segregated method and the thermodynamic analysis of turbulent flow, adding a sub-stepping procedure that ensures mass conservation at each time step. The accuracy of the algorithm is verified using the 2-D heated lid-driven cavity benchmark, using synthetic data.

Additionally, an improvement to the KF is the EnKF (*Ensemble Kalman Filter*) [8] which requires the creation of ensembles of states, in the spirit of Monte Carlo methods, that allows for a better calculation of the model error covariance matrix. In [9], the EnKF is applied to 2-D transonic flow around the RAE 2822 airfoil and the 3-D transonic flow around the ONERA M6 wing in order to determine the angles of attack, Mach numbers, and turbulent viscosities using experimental data. Furthermore, in [10] the EnKF coupled with a fourth-order CFD code is investigated for solving convection-diffusion-reaction problems and applied to the 1-D

convection-diffusion-reaction problem and the 2-D flame propagation. Similarly, in [11], the EnKF method is employed to incorporate the observation of velocity and different sources of wall quantities (e.g., wall shear stress, wall pressure distribution, lift and drag force) and applied to 2-D turbulent flows using synthetic data.

Another family of methods that can be formulated using either the variational or statistical DA approaches is the state observers or nudging methods. They emanate from control theory, and their function is to add a forcing term in the model equations proportional to the difference between numerical and observation data, "nudging" the model to the observation data. In [12], an observer method for the pressure differences between CFD and experimental data is applied to the turbulent flow through a duct of a square cross-section. In [13], an observer method for the pressure differences between CFD and experimental data is applied to the 2-D flow around a square cylinder with von Kármán vortex shedding. Similarly, in [14] a state observer-based DA method that is constructed based on a proportional-integral-derivative control law and acts on the pressure equation is applied to the flow around a square cylinder at $\mathrm{Re} = 100$ using synthetic data. Additionally, in [15] two state observer approaches, proportional and integral-proportional, for the velocity differences between CFD, using the unsteady RANS equations and experimental data are proposed and demonstrated using an experimental database that describes the time-averaged 3-D flow behind a generic car-mirror model. Finally, in [16] a nudging method for the velocity fields from the momentum equation is applied to the estimation of the turbulent flow around a canonical square cylinder at $\mathrm{Re} = 22,000$ solving and using DNS data.

Finally, hybrid methods exist. For example, in [17] variational, EnKF-based and ensemble-based variational DA approaches are investigated for the reconstruction of the unsteady 2-D flow past a cylinder in the presence of incident coherent gusts using synthetic data. Additionally, in [18] a HNEnKF (*Hybrid Nudging-Ensemble Kalman Filter*) approach is used in the shallow water equations. The HNEnKF effectively combines the advantages of the EnKF and the observation nudging to achieve more gradual and continuous DA by computing the nudging coefficients from the flow-dependent, time-varying error covariances of the EnKF. In [19] three methods are investigated for the combination of DNS and PTV (*Particle Tracking Velocimetry*) data. The proper orthogonal decomposition–Galerkin-projection approach with proportional feedback of PTV data, the DNS solver with similar feedback and the DNS solver with the extended KF are applied to a planar-jet problem at $\mathrm{Re} \approx 2,000$.

## 1.3 Thesis Outline

Following the introductory chapter, the Thesis is structured as follows:

- **Chapter 2**
  A quick introduction to Computational Fluid Dynamics is presented, containing the main concepts that will be used for the numerical solution of the flows throughout the Thesis. The governing equations of fluid flows are presented and then discretized using the Finite Volume Method. Additionally, the SIMPLE algorithm for the segregated solution of the Navier-Stokes equations is explained. Finally, the treatment of turbulence in numerical simulations and the RANS approach are presented.

- **Chapter 3**
  The main theory behind the field of Data Assimilation is presented. Both variational and sequential DA methods are described. Additionally, the necessary approximations that need to be made, so the implementation of DA methods in a numerical code is feasible are discussed. Finally, the nudging method is presented.

taken into

- **Chapter 4**
  The Sod Shock tube test case is investigated. It is one transient 1-D compressible flow problem of an ideal gas with a discontinuous initial condition. Both the numerical and the analytical solutions are presented. Then, the nudging method is implemented in order to improve the numerical results using the analytical solution. Finally, a parameter estimation technique utilizing the adjoint equations is used in order to optimize the gain matrix (Optimal nudging).

- **Chapter 5**
  The calculation of pressure fields using velocity field measurements calculated by the PIV method is investigated using the OpenFOAM Computational Fluid Dynamics toolbox. Initially, the pressure fields are calculated using two different methods, the solution of a Poisson equation and the pressure-velocity correction using an iterative algorithm (SIMPLE). The results of the two methods are compared. The implementation of the SIMPLE method is based on work previously conducted in the Laboratory for Innovative Environmental Technologies at the School of Mechanical Engineering at the National Technical University of Athens. Then, a grid independence study is performed to validate the results of the two methods. Finally, Data Assimilation methods, namely the nudging method and the Kalman Filter method, are implemented in the OpenFOAM toolbox in order to improve the calculation of the velocity fields using the SIMPLE algorithm. Different formulations of these methods are devised and implemented, leading to an investigation of the effect on the final solution.

- **Chapter 6**
  The Thesis summary and conclusions are presented, and suggestions for future work in the studied applications and developed codes are made.

# Chapter 2

# Computational Fluid Dynamics

This chapter lays out the theoretical background of CFD methods and techniques used in this Thesis. Firstly, the governing equations of fluid flows, the Euler and NS equations, are presented. Then, the Finite Volume discretization method is described and applied to the governing equations. Additionally, the numerical schemes used in the Applications described in this Thesis are derived. Finally, the solution of turbulent flows using the RANS equations is described.

## 2.1 Governing Equations

The Euler and NS equations are formal ways to express conservation laws in fluid flows. In this Thesis, two applications of these equations were studied. In the first application, the 1-D compressible Euler equations were used; in the second application, the 2-D incompressible NS were used. Thus the equations regarding the relevant space dimensions and compressibility behavior are described in the following subsections.

### 2.1.1 Euler Equations

The Euler equations are a system of quasilinear hyperbolic PDEs (*Partial Differential Equations*) governing the adiabatic and inviscid flow of fluid [20, 21]. They describe the conservation of mass, momentum, and energy throughout the flow of a fluid in space. Depending on the space dimensions, the system of equations can have 3 (1-D) up to 5 (3-D) equations. The 1-D compressible Euler equations in the conservative form using vector notation are presented below

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{0} \tag{2.1}$$

By applying the chain rule, eq. (2.1) transforms to

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x} = \mathbf{0} \tag{2.2}$$

The Jacobian matrix of the system is defined as $\frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \mathbf{A}$, with $\mathbf{A} = \mathbf{A}(\mathbf{U})$, then eq. (2.2) becomes

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{U}}{\partial x} = \mathbf{0} \tag{2.3}$$

where $\mathbf{U}$ is the vector of conservative variables

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix} \tag{2.4}$$

and **F** is the vector of their respective fluxes

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u H \end{bmatrix} \tag{2.5}$$

where $\rho$ is the density, $u$ is the velocity, and $p$ is the static pressure. The specific energy $E$ and enthalpy $H$ are given by

$$E = \frac{1}{\gamma - 1}\frac{p}{\rho} + \frac{u^2}{2} \tag{2.6}$$

$$H = \frac{\gamma}{\gamma - 1}\frac{p}{\rho} + \frac{u^2}{2} \tag{2.7}$$

$$H = \frac{E + p}{\rho} \tag{2.8}$$

where $\gamma$ is the heat capacity ratio of the fluid. Using eqs. (2.6) and (2.7), eq. (2.5) can be rewritten to

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u\left(E + p\right) \end{bmatrix} \tag{2.9}$$

Finally, the primitive variables vector is given by

$$\mathbf{W} = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix} \tag{2.10}$$

The conservative form is preferred here because in a compressible flow, where shocks and contact discontinuities might be present; only the conservative variables are appropriate for calculating numerical fluxes.

### 2.1.2 Navier-Stokes Equations

The NS (*Navier-Stokes*) equations are a system of parabolic PDEs governing viscous flow. They express the conservation of mass and conservation of momentum for Newtonian fluids. For compressible flow, they are coupled with an equation of state relating pressure, temperature, and density. Depending on the space dimensions, the system of equations can have two (1-D) up to 4 (3-D) equations. The transient incompressible NS equations using tensor (or Einstein) notation are presented below

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.11}$$

$$\rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \tag{2.12}$$

where $i = 1, 2$ for a 2-D flow. Additionally, $f_i$ is the $i$ component of external forces, e.g., gravity or thermal forces, and $\tau_{ij}$ is the viscous stress tensor

$$\tau_{ij} = \rho\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.13}$$

where $\nu$ is the kinematic viscosity of the fluid.

## 2.2 The Finite Volume Method

The FVM (*Finite Volume Method*) [22] is a discretization method for PDEs. It adopts the idea of control volumes used to create models of physical systems. A control volume represents a region of space, usually fixed, enclosed by a surface through which fluid flows. It applies conservation equations, e.g., of mass, momentum and energy, by balancing fluxes, due to inflow and outflow at the bounding surface, with additional sources within the volume. Rather than using a single control volume to describe an entire physical system, the FVM splits the system domain into multiple connected finite volumes.

In the FVM, the governing PDEs are integrated throughout the finite volumes and using the Green-Gauss (or divergence) theorem, these volume integrals are converted to surface integrals. These terms are then evaluated as fluxes at the finite volume's surface. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative.

Additionally, due to their nature, FVM methods can be easily formulated to allow for unstructured meshes in conjunction with, e.g., finite differences methods even though, in this Thesis, only structured and orthogonal meshes were used.

### 2.2.1 Euler Equations

The finite volume of a 1-D uniform grid is presented in fig. 2.1. The volume surrounding the $i-$ th cell is described, and the blue vertical lines represent the bounding surfaces of the volume, denoted by $i \pm 1/2$. The $L$ and $R$ indices refer to the different sides of the cells' interfaces, where the Riemann problems are solved.

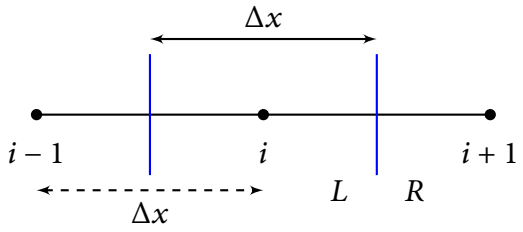



**Figure 2.1:** Finite volume for 1-D problem.

Firstly, eq. (2.1) is integrated throughout the finite volume described in fig. 2.1

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \mathbf{U}}{\partial t}\, \mathrm{d}x + \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}\, \mathrm{d}x = \mathbf{0} \tag{2.14}$$

Then, approximating the average value over the $i-$ th cell at the $n-$ th time step by

$$\mathbf{U}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}\, \mathrm{d}x \tag{2.15}$$

where $\Delta x = x_{i+1/2} - x_{i=1/2}$ is the length of the cell. This approximation corresponds to the value at the cell center with 2nd-order accuracy. Using the Leibniz theorem for the temporal term, it becomes

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \mathbf{U}}{\partial t}\, \mathrm{d}x = \frac{\partial}{\partial t} \left( \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}\, \mathrm{d}x \right) \tag{2.16}$$

Thus, using a 1st order Euler time integration scheme, eq. (2.16) becomes

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \mathbf{U}}{\partial t}\, \mathrm{d}x = \frac{\Delta x}{\Delta t}\left(\mathbf{U}_i^{n+1} - \mathbf{U}_i^{n}\right) \tag{2.17}$$

Then, the spatial term becomes

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}\, \mathrm{d}x = \mathbf{F}\left(\mathbf{U}\right)_{i+1/2} - \mathbf{F}\left(\mathbf{U}\right)_{i-1/2} \tag{2.18}$$

Combining eqs. (2.17) and (2.18), a Godunov-like scheme [21, 23] is devised

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^{n} - \frac{\Delta x}{\Delta t}\left[\mathbf{F}(\mathbf{U})_{i+1/2} - \mathbf{F}(\mathbf{U})_{i-1/2}\right] \tag{2.19}$$

#### 2.2.1.1 Roe Method

Because a discretization method is used, local discontinuities between neighboring cells exist. An initial value problem that is described by a conservation governing equation and has piecewise constant initial data with a local discontinuity is called a Riemann problem. As noted in the previous section, the flow variables are computed at the center of each cell and are the averaged values across the finite volume, with their values changing due to the fluxes in the cell surfaces. To calculate these inter-cell fluxes, local Riemann problems must be solved on every cell surface. As the Riemann problem is non-linear, costly iterative solvers need to be used to obtain an exact solution [21, 23]

In order to reduce the high computational cost of an exact Riemann solver, approximate Riemann solvers are used. One of the most common approximate Riemann solvers is the one developed by Roe [24]. A linearization of the Jacobian matrix of the system is performed, converting the system of equations into a system of linear hyperbolic PDEs that can be easily solved at each cell interface. In Roe's method, the linearization of the Jacobian is denoted by the Roe matrix $\tilde{\mathbf{A}}$. The following properties must hold:

- $\tilde{\mathbf{A}}$ is diagonalizable with real eigenvalues, ensuring that the new linear system is hyperbolic
- $\tilde{\mathbf{A}}$ is consistent with the exact Jacobian: $\tilde{\mathbf{A}}\left(\mathbf{U}, \mathbf{U}\right) = \mathbf{A}\left(\mathbf{U}\right)$
- $\tilde{\mathbf{A}}$ ensures conservation across discontinuities: $\mathbf{F}_R - \mathbf{F}_L = \tilde{\mathbf{A}}(\mathbf{U}_R - \mathbf{U}_L)$

A brief investigation of the 1-D Euler equations will be performed. The exact Jacobian of the system is given by [21]:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-1)H - u^2 - c^2 & (3-\gamma)u & \gamma - 1 \\ \frac{1}{2}u\left[(\gamma-3)H - c^2\right] & H - \gamma u^2 & \gamma u \end{bmatrix} \tag{2.20}$$

where $u$ is the velocity, $H$ is the specific enthalpy, $c$ is the speed of sound and $\gamma$ is the heat capacity ratio of the fluid. Solving the eigenvalue problem

$$|\mathbf{A} - \lambda_i \mathbf{I}| = \mathbf{0} \tag{2.21}$$

the system eigenvalues are obtained:

$$\begin{aligned}\lambda_1 &= u \\ \lambda_2 &= u + c \\ \lambda_3 &= u - c\end{aligned} \tag{2.22}$$

Thus, the eigenvalue matrix is given by $\mathbf{\Lambda} = \text{diag}\,(\lambda_1, \lambda_2, \lambda_3)$. Then, the left $\mathbf{L}$ and right $\mathbf{R}$ eigenvectors are computed

$$(\mathbf{A} - \lambda_i \mathbf{I})\,\mathbf{R}_i = \mathbf{0} \tag{2.23}$$

$$\mathbf{L}_i^{\mathsf{T}}\,(\mathbf{A} - \lambda_i \mathbf{I}) = \mathbf{0} \tag{2.24}$$

yielding

$$\mathbf{R} = \begin{bmatrix}\mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3\end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & 1 \\ u & u+c & u-c \\ \frac{1}{2}u^2 & H+uc & H-uc \end{bmatrix} \tag{2.25}$$

$$\mathbf{L} = \begin{bmatrix}\mathbf{L}_1 \\ \mathbf{L}_2 \\ \mathbf{L}_3\end{bmatrix} = \begin{bmatrix} \frac{c^2 - \frac{\gamma-1}{2}u^2}{c^2} & \frac{(\gamma-1)u}{c^2} & \frac{1-\gamma}{c^2} \\ \frac{\frac{\gamma-1}{2}u^2 + c}{2c^2} & \frac{(1-\gamma)u+c}{2c^2} & \frac{\gamma-1}{2c^2} \\ \frac{\frac{\gamma-1}{2}u^2 - c}{2c^2} & \frac{(1-\gamma)u-c}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix} \tag{2.26}$$

The matrices containing the left and right eigenvectors of the system and the matrix containing system eigenvalues are used in order to produce a diagonalizable matrix. Roe's method proposes the usage of the diagonalizable $\tilde{\mathbf{A}}$ matrix, which is given by

$$\left|\tilde{\mathbf{A}}\right| = \tilde{\mathbf{R}}\left|\tilde{\mathbf{\Lambda}}\right|\tilde{\mathbf{L}} \tag{2.27}$$

where $\tilde{r}$ denotes the Roe average for a variable $r$. This averaging is performed by taking the geometric mean of the density values alongside the cell interface

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \tag{2.28}$$

Additionally, the Roe averaged velocity and specific enthalpy are given by

$$\tilde{u} = \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{2.29}$$

$$\tilde{H} = \frac{H_L\sqrt{\rho_L} + H_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{2.30}$$

and the Roe averaged speed of sound is calculated as

$$\tilde{c} = \sqrt{(\gamma - 1)\left(\tilde{H} - \tilde{u}^2/2\right)} \tag{2.31}$$

Thus, the Roe averaged system eigenvalues are

$$\begin{aligned}\tilde{\lambda}_1 &= \tilde{u} \\ \tilde{\lambda}_2 &= \tilde{u} + \tilde{c} \\ \tilde{\lambda}_3 &= \tilde{u} - \tilde{c}\end{aligned} \tag{2.32}$$

and equivalently, the Roe averaged right eigenvectors matrix is given by

$$\tilde{\mathbf{R}} = \begin{bmatrix} 1 & 1 & 1 \\ \tilde{u} & \tilde{u} + \tilde{c} & \tilde{u} - \tilde{c} \\ \frac{1}{2}\tilde{u}^2 & \tilde{H} + \tilde{u}\tilde{c} & \tilde{H} - \tilde{u}\tilde{c} \end{bmatrix} \tag{2.33}$$

Finally, according to Roe [24], the intercell flux is calculated by

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}_L + \mathbf{F}_R\right) - \frac{1}{2}\left|\tilde{\mathbf{A}}\right|\left(\mathbf{U}_L - \mathbf{U}_R\right) \tag{2.34}$$

#### 2.2.1.2 Face Reconstruction Scheme

From eq. (2.34), it is evident that a numerical scheme is needed for the reconstruction of the cell values into the cell interfaces so that the intercell fluxes can be calculated. The most obvious and straight-forward scheme is an Upwind-like scheme, which 1st order accurate in space and is defined as:

$$\begin{aligned} \mathbf{U}_L &= \mathbf{U}_i \\ \mathbf{U}_R &= \mathbf{U}_{i+1} \end{aligned} \tag{2.35}$$

So the values at cell centers are mapped to the cell interfaces "as-is". Even though it is numerically stable, this scheme presents a more diffusive behavior than higher-order schemes, which is evident in problems that contain discontinuities. Then, using piecewise linear approximations of each cell, a second order scheme is formulated, using Taylor expansion around the neighborhood of the cell center and central difference scheme for the calculation of the gradients, yielding

$$\begin{aligned} \mathbf{U}_L &= \mathbf{U}_i + \frac{1}{4}\left(\mathbf{U}_{i+1} - \mathbf{U}_{i-1}\right) \\ \mathbf{U}_R &= \mathbf{U}_{i+1} - \frac{1}{4}\left(\mathbf{U}_{i+2} - \mathbf{U}_i\right) \end{aligned} \tag{2.36}$$

Although the above second-order scheme provides greater accuracy for smooth solutions, it is not a TVD (*Total Variation Diminishing*) [25] scheme and introduces spurious oscillations into the solution where discontinuities or shocks are present. Thus, the MUSCL (*Monotonic Upstream-centered Scheme for Conservation Laws*) scheme was introduced by van-Leer [26]. MUSCL based numerical schemes extend the idea of using a linear piecewise approximation to each cell by using gradient limited left and right extrapolated states. This results in the following high resolution, TVD discretization scheme

$$\begin{aligned} \mathbf{U}_L &= \mathbf{U}_i + \frac{1}{2}\Phi\left(r_L\right)\left(\mathbf{U}_i - \mathbf{U}_{i-1}\right) \\ \mathbf{U}_R &= \mathbf{U}_{i+1} - \frac{1}{2}\Phi\left(r_R\right)\left(\mathbf{U}_{i+2} - \mathbf{U}_{i+1}\right) \end{aligned} \tag{2.37}$$

where $r_*$ is the slope ratio in each cell interface, given by

$$\begin{aligned} r_L &= \frac{\mathbf{U}_{i+1} - \mathbf{U}_i}{\mathbf{U}_i - \mathbf{U}_{i-1}} \\ r_R &= \frac{\mathbf{U}_{i+1} - \mathbf{U}_i}{\mathbf{U}_{i+2} - \mathbf{U}_{i+1}} \end{aligned} \tag{2.38}$$

The function $\Phi(r)$ is a limiter function that limits the slope of the piecewise approximations to ensure the solution is TVD, thereby avoiding the spurious oscillations that would otherwise

occur around discontinuities or shocks. The limiter is equal to zero when $r \leq 0$ and is equal to unity when $r = 1$. Thus, the MUSCL scheme using limiter functions degrades to 1st order at local extrema but tends to 2nd order over smooth parts of the domain. Here, a common limiter function is described, introduced by van-Leer [27]

$$\Phi(r) = \frac{r + |r|}{1 + |r|} \tag{2.39}$$

Combining eqs. (2.35) to (2.37) into a more compact form, the following expression is devised

$$\begin{aligned} \mathbf{U}_L &= \mathbf{U}_i + \frac{\varepsilon}{2}\Phi\left(r_L\right)\left(\mathbf{U}_i - \mathbf{U}_{i-1}\right) \\ \mathbf{U}_R &= \mathbf{U}_{i+1} - \frac{\varepsilon}{2}\Phi\left(r_R\right)\left(\mathbf{U}_{i+2} - \mathbf{U}_{i+1}\right) \end{aligned} \tag{2.40}$$

which correspond to different schemes for different values of $\epsilon$:

- $\varepsilon = 0$, 1st order scheme
- $\varepsilon = 1$ and $\Phi(r) = 0.5$, 2nd order scheme
- $\varepsilon = 1$ and $\Phi(r) = \frac{r+|r|}{1+|r|}$, 2nd order MUSCL scheme with van-Leer limiter

### 2.2.2 Navier-Stokes Equations

In this section, the FVM discretization of the NS equations is presented using the *cell-centered* formulation, using *collocated* structured regular meshes. In the cell-centered formulation, which is the most widely used FVM formulation, the state variables, e.g., velocity and pressure, are stored in each finite volume center in comparison to the vertex-centered formulation where they are stored in the center of each finite volume vertex. Additionally, the use of collocated meshes means that both pressure and velocity variables are stored in the cell centers, compared to the staggered mesh approach, where the pressure, or scalar variables, in general, are stored in cell centers, and velocity is stored at the cell faces (or vertexes).

The use of the cell-centered formulation using collocated meshes is selected, as it is the most common approach in incompressible NS solvers and is implemented in the OpenFOAM open-source CFD toolbox [28–30], which is utilized in the second application presented in this Thesis. Additionally, the method is formulated using a structured, regular mesh, as this family of meshes is used in the second application.

The finite volume of a 2-D structured, regular grid is shown in fig. 2.2. The cell center is denoted by $P$, and the rest of the indices refer to the neighboring cells using the relative position to $P$ ($S$ : South, $N$ : North, $W$ : West, $E$ : East). The continuous lines represent the surfaces surrounding the finite volumes, while the dashed lines represent the grid where the cell centers lie.

Firstly, the NS equations for incompressible flow, eqs. (2.11) and (2.12), are rewritten in vector form and reformulated for steady state $\left(\frac{\partial \mathbf{u}}{\partial t} = 0\right)$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.41}$$

$$\rho \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \rho \nabla \cdot (\nu \nabla \mathbf{u}) + \rho \mathbf{f} \tag{2.42}$$

Next, the momentum equation, eq. (2.42), is integrated throughout the finite volume $V_P$

$$\rho \int_{V_P} \nabla \cdot (\mathbf{u}\mathbf{u})\, \mathrm{d}V = -\int_{V_P} \nabla p\, \mathrm{d}V + \rho \int_{V_P} \nabla \cdot (\nu \nabla \mathbf{u})\, \mathrm{d}V + \rho \int_{V_P} \mathbf{f}\, \mathrm{d}V \tag{2.43}$$

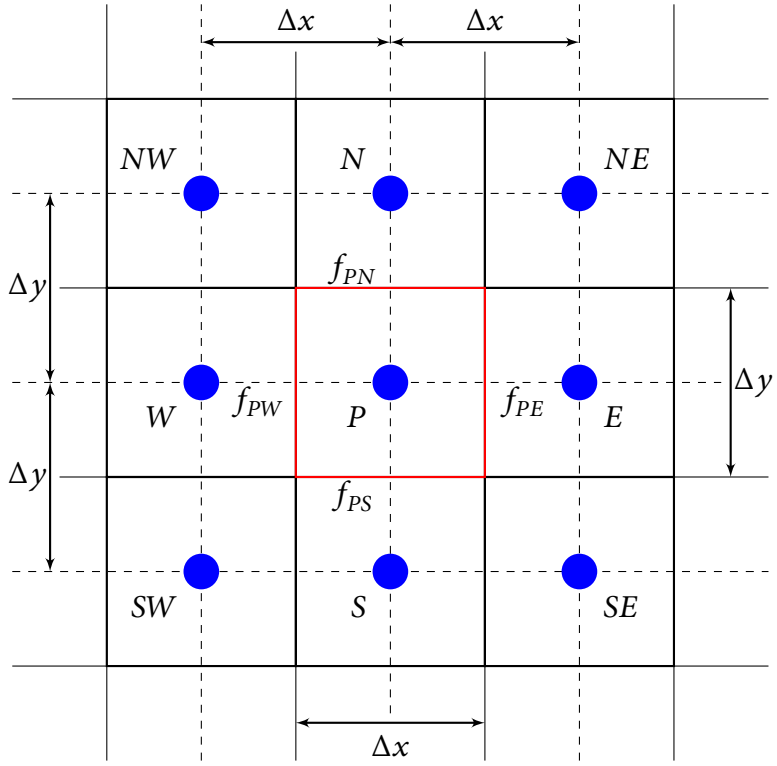



**Figure 2.2:** Finite volume for 2-D regular grid.

Finally, the volume integrals are transformed into surface integrals using the divergence theorem, and then using numerical schemes to reconstruct the cell center variables into the cell surfaces, the final discretized form of the NS equations is reached [28–32]. In the following sections, the terms presented in eq. (2.43) are discretized using a notation relevant to the one used in the OpenFOAM toolbox [28, 29].

#### 2.2.2.1 Source Term

The source term $\rho\mathbf{f}$ is assumed to be constant inside the finite volume and independent of $\mathbf{u}$. Additionally, assuming that the depth of the computational domain is unity and the grid is regular, $V_P = \Delta x \Delta y$. The integral of the source term becomes

$$\int_{V_P} \mathbf{f}\, \mathrm{d}V = \mathbf{f}_P V_P = \mathbf{f}_P \Delta x \Delta y \tag{2.44}$$

#### 2.2.2.2 Gradient Term

The pressure gradient term is computed using the divergence theorem

$$\int_{V_P} \nabla p\, \mathrm{d}V = \frac{1}{V_P} \oint_{\partial V_P} p\, \mathrm{d}\mathbf{S} = \frac{1}{V_P} \sum_f p_f \mathbf{S}_f \tag{2.45}$$

where the $f$ index means the surfaces or faces that the cell shares with its neighboring cells. Additionally, $\mathbf{S}_f = \mathbf{n}_f S_f$, where $\mathbf{n}_f$ is the normal vector to the $f$ face and $S_f$ is the surface area. For a regular grid, $S_f = \Delta x$ or $S_f = \Delta y$, depending on the surface location. The required reconstruction of $p$ at the cell faces is performed by a simple linear interpolation, using central

differencing (Gauss linear). For example, $p_{f_{PE}}$ is given by

$$p_{f_{PE}} = f_{PE} p_P + (1 - f_{PE})\, p_E \tag{2.46}$$

where $f_{PE} = \frac{|fE|}{|PE|}$, or for fig. 2.2 $f_{PE} = \frac{\Delta x/2}{\Delta x}$. Consequently, for a regular and orthogonal mesh, this scheme is a 2nd order scheme. The formulation is the same for a face in the vertical direction, using $\Delta y$ instead of $\Delta x$

#### 2.2.2.3 Advection Term

The advection term is computed using the divergence theorem

$$\int_{V_P} \nabla \cdot (\mathbf{uu})\, \mathrm{d}V = \frac{1}{V_P} \oint_{\partial V_P} \mathbf{u} \left(\mathbf{n}_f \cdot \mathbf{u}\right) \mathrm{d}\mathbf{S} = \frac{1}{V_P} \sum_f \mathbf{u}_f \left(\mathbf{S}_f \cdot \mathbf{u}_f\right) = \frac{1}{V_P} \sum_f \mathbf{u}_f \mathbf{Flux}_f \tag{2.47}$$

where $\mathbf{Flux}_f = \mathbf{S}_f \cdot \mathbf{u}_f$ is the volumetric flux across the $f$ face. In order to capture the physical behavior of advection, the Upwind scheme [33] is used (Gauss upwind), which simply uses the value of the cell in the upwind direction of the flow. For example, $\mathbf{u}_{f_{PE}}$ is given by

$$\mathbf{u}_{f_{PE}} = \max\left(\mathbf{Flux}_f, 0\right)\mathbf{u}_P + \min\left(\mathbf{Flux}_f, 0\right)\mathbf{u}_E \tag{2.48}$$

The Upwind scheme is 1st order accurate in space and is bounded. Additionally, a central differencing scheme can be used, as formulated in eq. (2.46) (Gauss linear).

#### 2.2.2.4 Diffusion Term

The diffusion term is computed using the divergence theorem

$$\int_{V_P} \nabla \cdot (\nu \nabla \mathbf{u})\, \mathrm{d}V = \frac{1}{V_P} \oint_{\partial V_P} \nu \left(\mathbf{n}_f \cdot \nabla \mathbf{u}\right) \mathrm{d}\mathbf{S} = \frac{1}{V_P} \sum_f |\mathbf{S}_f| \left(\nu \nabla_{\mathbf{n}} \mathbf{u}\right)_f = \frac{1}{V_P} \sum_f \nu |\mathbf{S}_f| \cdot \left(\nabla_{\mathbf{n}} \mathbf{u}\right)_f \tag{2.49}$$

where $\left(\nabla_{\mathbf{n}} \mathbf{u}\right)_f$ is the surface normal velocity gradient at $f$ face. For an orthogonal grid, the surface normal gradient is calculated using central differencing (Gauss linear orthogonal). For example, $\left(\nabla_{\mathbf{n}} \mathbf{u}\right)_{f_{PE}}$ is given by

$$\left(\nabla_{\mathbf{n}} \mathbf{u}\right)_{f_{PE}} = \frac{\varphi_E - \varphi_P}{|PE|} \tag{2.50}$$

which is 2nd order accurate in space.

#### 2.2.2.5 Pressure-Velocity Coupling

An interesting behavior of the NS equations is that for a given pressure field, the momentum equations can be solved, resulting in a velocity field that does not necessarily obey the mass conservation constraint. The most common approach is constructing an iterative prediction correction algorithm to ensure that the resulting fields conform to the mass and momentum conservation constraints. In this Thesis, the SIMPLE (*Semi-Implicit Method for Pressure-Linked Equations*) algorithm [34] is used. Firstly, the momentum equation is written in a semi-discretized form:

$$a_P \mathbf{u}_P = \mathbf{H}(\mathbf{u}) - \nabla p_P \tag{2.51}$$

where $\mathbf{u}_P$ is the velocity vector at point P, $a_P$ the coefficient of $\mathbf{u}$ at the discretized form and $\mathbf{H}(\mathbf{u})$ is defined by

$$\mathbf{H}(\mathbf{u}) = \sum_{nb} a_{nb}\mathbf{u}_{nb} + \mathbf{b}_P \tag{2.52}$$

Consequently, the $\mathbf{H}$ matrix contains the contribution of the neighboring cells of the P cell to the discretized momentum equation for the calculation of $\mathbf{u}_P$, depending on the used discretization schemes and the momentum source terms $\mathbf{b}_P$ (e.g., gravity force or control forcing). Combining eqs. (2.51) and (2.52) yields

$$\mathbf{u}_P = \frac{1}{a_P}\left[\mathbf{H}(\mathbf{U}) - \nabla p_P\right] \tag{2.53}$$

The SIMPLE algorithm is initialized by making an initial guess for the pressure field $p^*$ and the corresponding velocity field $\mathbf{u}^*$. Then, eq. (2.51) becomes

$$a_P\mathbf{u}_P^* = \mathbf{H}(\mathbf{u}^*) - \nabla p_P^* \tag{2.54}$$

where the terms $\mathbf{H}(\mathbf{u}^*)$ is given by

$$\mathbf{H}(\mathbf{u}^*) = \sum_{nb} a_{nb}\mathbf{u}_{nb}^* + \mathbf{b}_P \tag{2.55}$$

and similarly to eq. (2.53), $\mathbf{u}_P^*$ is given by

$$\mathbf{u}_P^* = \frac{1}{a_P}\left[\mathbf{H}(\mathbf{u}^*) - \nabla p_P^*\right] \tag{2.56}$$

Then, an assumption about the correction field to the pressure and velocity fields is defined by

$$\mathbf{u} = \mathbf{u}^* + \mathbf{u}' \tag{2.57}$$

$$p = p^* + p' \tag{2.58}$$

Subtracting eq. (2.56) from eq. (2.53) yields

$$\mathbf{u}_P - \mathbf{u}_P^* = \frac{1}{a_P}\left\{\mathbf{H}(\mathbf{u}) - \mathbf{H}(\mathbf{u}^*) - \left[\nabla p_P - \nabla p_P^*\right]\right\} \tag{2.59}$$

Next, using the definitions from eqs. (2.57) and (2.58) to transform eq. (2.59) and taking into account the definition of $\mathbf{H}(\mathbf{u})$ in eq. (2.52), the equation that connects the pressure and velocity corrections is devised

$$\mathbf{u}_P' = \frac{1}{a_P}\left[\sum_{nb} a_{nb}\mathbf{U}_{nb}' - \nabla p_P'\right] \tag{2.60}$$

Then, assuming that the contribution of the off-diagonal terms (the neighboring cells) in the correlation between the pressure and velocity, corrections is negligible, or $\sum_{nb} a_{nb}\mathbf{u}_{nb}' = 0$, eq. (2.60) becomes

$$\mathbf{u}_P' = -\frac{\nabla p_P'}{a_P} \tag{2.61}$$

Consequently, $\mathbf{u}_P$ is given by

$$\mathbf{u}_P = \mathbf{u}_P^* - \frac{\nabla p_P'}{a_P} \tag{2.62}$$

Substituting eq. (2.62) into the mass equation ($\nabla \cdot \mathbf{u}_P = 0$), a Poisson equation for the pressure correction is devised

$$\nabla \cdot \left( \frac{1}{a_P} \nabla p'_P \right) = \nabla \cdot \mathbf{u}^*_P \tag{2.63}$$

Then, eq. (2.63) can be solved, leading to the pressure correction field $p'$, and then the pressure and velocity fields are corrected using eqs. (2.57) and (2.58) and used as the initial guesses in the next iteration. Thus an iterative algorithm is devised, at whose convergence the pressure and velocity corrections will be zero. Consequently, the assumption that $\sum_{nb} a_{nb} \mathbf{u}'_{nb} = 0$ does not influence the final result, as in the converged state, the correction terms are zero, but this assumption influences the convergence speed of the algorithm. Additionally, the usage of under relaxation is mandatory to achieve convergence. In relaxation, a fraction of the new solution is used, while the rest of the new solution comes from the values at the previous iteration. So after the $k-$ th iteration and the calculation of the corrected fields, the initial values at the $k+1-$ th are given by

$$p^{k+1} = \beta_p p^{\text{corr.}} + \left(1 - \beta_p\right) p^k \tag{2.64}$$

$$\mathbf{u}^{k+1} = \beta_u \mathbf{u}^{\text{corr.}} + \left(1 - \beta_u\right) \mathbf{u}^k \tag{2.65}$$

where $p^{\text{corr.}} = p^* + p'$ is the corrected pressure field and $\mathbf{u}^{\text{corr.}} = \mathbf{u}^* + \mathbf{u}'$ is the corrected velocity field. Taking into account that $\mathbf{u}^* \equiv \mathbf{u}^k$ and $p^* \equiv p^k$, eqs. (2.64) and (2.65) become

$$p^{k+1} = p^k + \beta_p p' \tag{2.66}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \beta_u \mathbf{u}' \tag{2.67}$$

typical values for the relaxation factors are 0.3 for the pressure relaxation factor $\beta_p$ and 0.7 for the velocity relaxation factor $\beta_u$. Thus, the SIMPLE algorithm can be summarized as follows:

- Initial guess of $p^*$ and $\mathbf{u}^*$
- Until convergence is reached:
    1. Solve discretized momentum equation, eq. (2.56) $\rightarrow \mathbf{u}^*$
    2. Solve pressure correction equation, eq. (2.63) $\rightarrow p'$
    3. Correct pressure and velocities using eqs. (2.57) and (2.58)
    4. Set $p^{k+1}$ and $\mathbf{u}^{k+1}$ using under relaxation from eqs. (2.66) and (2.67)
- Finalize procedure

Each system of equations is solved using iterative linear solvers. In this Thesis, the convergence criteria are defined by the initial normalized residuals of the iterative solvers. Specifically, for momentum convergence, they must be under $10^{-2}$, and for pressure convergence, they must be under $10^{-5}$.

## 2.3 Turbulence and Turbulence Modeling

The flow was considered laminar in all of the sections above. But most engineering problems require the resolution of turbulent flows. In this Thesis, turbulence is present in the second examined application. Chaotic changes in pressure and velocity characterize turbulent flow, and

the correct resolution of turbulence requires the solution of the NS equations in substantially small time and space scales, requiring lots of computational power.

Thus, many methods have been developed to reduce computational requirements using turbulence modeling. The most used are arranged by increasing computational cost:

- RANS (*Reynolds Averaged Navier-Stokes*)
- LES (*Large Eddy Simulation*)
- DNS

The RANS method uses Reynolds averaging (temporal averaging) to decompose the NS equations into a mean and a fluctuating field, solving the equations of the mean field, making some assumptions about the fluctuating field, and using model equations. RANS is the oldest of these methods and is the industry standard for turbulent flows. The LES method uses spatial and temporal filtering (averaging) of the NS equations. The smaller temporal and spatial scales are solved using modeling just as RANS while the larger scales are resolved by direct computation. Finally, the DNS method, as its name implies, does not take advantage of any turbulence model and solves the NS equations as described in the previous section in all of the turbulent spatial and temporal scales, leading to extreme computational costs. DNS is used only for academic purposes.

### 2.3.1 Reynolds Averaged Navier-Stokes Equations

In this Thesis, the RANS method is used to simulate steady-state turbulent flows. Firstly, the pressure and velocity fields are decomposed into a mean and a fluctuating component. For example, the velocity field is decomposed into

$$\mathbf{u}(t) = \bar{\mathbf{u}} + \mathbf{u}'(t) \tag{2.68}$$

where $\bar{\mathbf{u}}$ is the mean field and $\mathbf{u}'(t)$ is the fluctuating field. The components are defined by integrating $\mathbf{u}(t)$ through a time interval $T$ big enough so the following stand:

$$\bar{\mathbf{u}} = \frac{1}{T}\int_{t_0}^{t_0+T} \mathbf{u}(t)\,\mathrm{d}t \tag{2.69}$$

$$\mathbf{0} = \frac{1}{T}\int_{t_0}^{t_0+T} \mathbf{u}'(t)\,\mathrm{d}t \tag{2.70}$$

Additionally, the mean value of the fluctuating field is equal to zero $\overline{\mathbf{u}'} = \mathbf{0}$, but $\overline{\mathbf{u}'^2} \neq \mathbf{0}$. Using the Reynolds averaged pressure and velocity fields, the steady-state, incompressible NS equations are transformed into the RANS equations given in tensor notation by

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2.71}$$

$$\rho\frac{\partial \bar{u}_i}{\partial t} + \rho\bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\bar{\tau}_{ij} + \tau_{ij}^R\right) + \rho\bar{f}_i \tag{2.72}$$

where $\bar{f}_i$ is the averaged external forces, e.g., gravity, while $\bar{\tau}_{ij}$ is the stress tensor of the mean field

$$\bar{\tau}_{ij} = \rho\nu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \tag{2.73}$$

where $\tau_{ij}^R$ is the Reynolds stress tensor given by

$$\tau_{ij}^R = -\rho\overline{u_i' u_j'} \tag{2.74}$$

The Reynolds stresses in the RANS method essentially express the momentum flux through a finite volume due to the turbulent fluctuations of the velocity field in comparison with the mean field. From eqs. (2.71) and (2.72), it is evident that a way to model the Reynolds stress tensor is needed.

### 2.3.2 Boussinesq Hypothesis

Boussinesq proposed [35] to model the Reynolds stress tensor, so it resembles the strain rate tensor given by

$$-\rho\overline{u_i' u_j'} = \rho\nu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\rho k\delta_{ij} \tag{2.75}$$

where $\delta_{ij}$ is the Kronecker delta, $\nu_t$ is the eddy viscosity and $k = \frac{1}{2}\rho\overline{u_i' u_i'}$ is the turbulent kinetic energy. Thus, eq. (2.72) becomes

$$\rho\frac{\partial \bar{u}_i}{\partial t} + \rho\bar{u}_j\frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \rho\frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\right] - \frac{2}{3}\rho\frac{\partial k}{\partial x_i} + \rho\bar{f}_i \tag{2.76}$$

Based on the Boussinesq hypothesis, a plethora of turbulence models have been developed whose aim is to either directly calculate $\nu_t$ using algebraic equations and experimental data (algebraic models - zero order models), or the calculation of other fields that describe turbulent behavior, e.g., turbulent kinetic energy $k$ and dissipation rate of turbulent kinetic energy $\varepsilon$, by solving advection-diffusion PDEs. Some example of these models are the $k-\varepsilon$ [36], $k-\omega$ [37] and $k-\omega$ SST [38] models. Additionally, other models require the solution of up to 6 advection-diffusion PDEs for the direct calculation of the Reynolds stress tensor components, these models are called Reynolds Stress Models and some examples of them are the LRR (*Launder-Reece-Rodi*) [39] and the SSG (*Speziale-Sarkar-Gatski*) [40] models. Although, in this Thesis, due to the availability of experimental data of successfully resolved turbulent flow for the Application that will be studied, no turbulence closure models will be used due to the availability of the Reynolds stress terms, calculated by a previously conducted experiment in the Laboratory for Innovative Environmental Technologies at the School of Mechanical Engineering at the National Technical University of Athens [41–44].

# Chapter 3

# Data Assimilation Methods

In this Chapter, the motivation for the use of DA (*Data Assimilation*) methods is described. Firstly, the DA problem of a dynamical model is mathematically defined. Next, the primary methods in the sequential DA field are formulated. Then, variational DA methods are described. Finally, a simpler DA method called nudging is presented.

## 3.1 Problem Definition

Initially, a discrete model for the evolution of a physical system from time $t_k$ to time $t_{k+1}$, is described by a dynamic state equation

$$\mathbf{x}_{k+1}^f = \mathbf{M}_{k+1}\mathbf{x}_k^f + \mathbf{B}_{k+1}\mathbf{c}_{k+1} + \mathbf{w}_k \tag{3.1}$$

where $\mathbf{x}$ is the model's state vector with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{M}$ is the corresponding dynamics operator that can be time-dependent. The $f$ index refers to the forecast state, the state that is predicted using the model equations. Additionally, $\mathbf{B}_k$ is the control-input model which is applied to the control vector $\mathbf{c}_k$ and $\mathbf{w}_k$ represents the modeling error with an associated error covariance matrix $\mathbf{Q}_k$. Then, observations or measurements $\mathbf{y} \in \mathbb{R}^m$ are defined by

$$\mathbf{y}_k^o = \mathbf{H}_k\mathbf{x}_k^t + \mathbf{v}_k \tag{3.2}$$

where $\mathbf{H}$ is an observation operator that maps model space to observation space (usually $m \ll n$) and can be time-dependent, and $\mathbf{v}_k$ represent the measurement/observation errors that can be considered a statistical process with a zero mean and an associated covariance matrix $\mathbf{R}_k$. The $t$ index refers to the unknown *true* state of the system.

Considering there is no control using model input, eq. (3.1) reduces to

$$\mathbf{x}_{k+1}^f = \mathbf{M}_{k+1}\mathbf{x}_k^f + \mathbf{w}_k \tag{3.3}$$

The modeling of the $\mathbf{w}_k$ and $\mathbf{v}_k$ is an important aspect of DA and in the methods investigated in this Thesis they are considered to be independent and with Gaussian/normal probability distributions

$$\mathbf{w}_k \sim N(0,\, \mathbf{Q}_k) \tag{3.4}$$

$$\mathbf{v}_k \sim N(0,\, \mathbf{R}_k) \tag{3.5}$$

## 3.2 Statistical/Sequential Methods

In this section, the main ideas behind the statistical approach to DA are presented. Firstly, the BLUE (*Best Linear Unbiased Estimator*) estimator is defined, and then the Kalman filter algorithm is described. Finally, some caveats and terms that require special treatment in the KF approach are discussed.

### 3.2.1 Optimal Interpolation - BLUE

A steady-state problem with scalar states is initially considered (so the $k$ index can be dropped). Assuming that the observation operator is scalar and equal to one, $\mathbf{H} = 1$ the observation becomes

$$y^o = x^o \tag{3.6}$$

An approximation for the *analysis* state $x^a$; the state that is produced by the DA method, is assumed as a simple linear combination of the model state, or *forecast* $x^f$ and the measured state $x^o$, yielding

$$x^a = x^b + w\left(x^o - x^b\right) \tag{3.7}$$

where $w$ is a weight factor with $0 \leq w \leq 1$. The focus of this analysis is to find a $w$ that provides an approximation of $x^a$ that minimizes the error variance. The analysis, forecast and observation errors are defined

$$e^a = x^a - x^t \ , \ e^f = x^f - x^t \ , \ e^o = x^o - x^t \tag{3.8}$$

where $x^t$ is the unknown true state. Subtracting $x^t$ from eq. (3.7) and using the definition from above yields

$$e^a = e^f + w\left(e^o - e^f\right) \tag{3.9}$$

Then, the expected value of eq. (3.9) is given by

$$\langle e^a \rangle = \left\langle e^f \right\rangle + w\left(\langle e^o \rangle - \left\langle e^f \right\rangle\right) \tag{3.10}$$

In order to have the optimal prediction of the state, taking into account the uncertainties/errors in both the model and observations, the expected value of the analysis errors should be zero.

$$\langle e^a \rangle = 0 \tag{3.11}$$

This means that the estimate of the true state is *unbiased*. Then, the *variance* of eq. (3.7) is calculated using the notation $\sigma^2 = \left\langle (e - \langle e \rangle)^2 \right\rangle$ and the condition of eq. (3.11) is taken into account:

$$\sigma_a^2 = \sigma_f^2 + w^2 \left\langle \left(e^o - e^f\right)^2 \right\rangle + 2w \left\langle e^f \left(e^o - e^f\right) \right\rangle \tag{3.12}$$

If $e^o$ and $e^b$ are uncorrelated, another basic assumption in this framework, eq. (3.12) reduces to

$$\sigma_a^2 = \sigma_f^2 + w^2 \left(\sigma_o^2 + \sigma_f^2\right) - 2w\sigma_f^2 \tag{3.13}$$

By taking the derivative of eq. (3.13) with respect to $w$ and demanding it to be equal to zero, in order for $\sigma_a$ to be minimum, an equation for the optimal weight $w_*$ is determined

$$w_* = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_o^2} = \frac{1}{1 + \sigma_o^2/\sigma_f^2} \tag{3.14}$$

thus, substituting eq. (3.14) into eq. (3.13), $\sigma_a$ is given by

$$\sigma_a^2 = w_* \sigma_o^2 + (1 - w_*)^2 \sigma_f^2 \tag{3.15}$$

or equivalently

$$\sigma_a^2 = \frac{\sigma_f^2 \sigma_o^2}{\sigma_o^2 + \sigma_f^2} = (1 - w_*)^2 \sigma_f^2 \tag{3.16}$$

thus $\sigma_a^2 \leq \sigma_f^2$, and it can be shown [1] that $\sigma_a^2 \leq \min\left(\sigma_f^2, \sigma_o^2\right)$. Substituting, eq. (3.14) into eq. (3.7), the BLUE estimator for $x^a$ is given by

$$x^a = x^f + \frac{1}{1 + \sigma_o^2/\sigma_f^2}\left(x^o - x^f\right) \tag{3.17}$$

Three special cases can be considered:

- If the observation is very accurate, $\sigma_o^2/\sigma_f^2 \ll 1$, thus $x^a \approx x^o$
- If the forecast is very accurate, $\sigma_o^2/\sigma_f^2 \gg 1$, thus $x^a \approx x^f$
- If both variances are close to each other, $\sigma_o^2/\sigma_f^2 \approx 1$, thus $x^a \approx \frac{x^a + x^f}{2}$

This approach can be extended to a vector state $\mathbf{x}^a$, using the Gauss-Markov theorem [1] for the system defined in eq. (3.3), again dropping the $k$ index for simplicity. The analysis state vector $\mathbf{x}^a$ is given by

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{K}(\mathbf{y}^o - \mathbf{H}\mathbf{x}^f) \tag{3.18}$$

where the optimal gain matrix $\mathbf{K}$ gain is given by the form

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^\mathsf{T} (\mathbf{H}\mathbf{P}^f \mathbf{H}^\mathsf{T} + \mathbf{R})^{-1} \tag{3.19}$$

where $\mathbf{P}^f$ and $\mathbf{R}$ are the forecast error covariance matrix and observation error covariance matrix, respectively.

### 3.2.2 Kalman Filter - KF

Next, the most well-known sequential DA scheme is described, the KF (*Kalman Filter*) [5]. It is applied to transient problems, and it aims to the minimization of the a posteriori error covariance.

The KF consists of a prediction/forecast step and a correction/analysis step. At the time $t_k$, the result of a previous forecast is available, $\mathbf{x}_k^f$, and the result of an ensemble of observations in $\mathbf{y}_k^o$. Based on these two vectors, we perform an analysis that produces $\mathbf{x}_k^a$. Then, the evolution model is used to obtain a prediction of the state at time $t_{k+1}$. Finally, the result of the forecast is denoted $\mathbf{x}_{k+1}^f$ and becomes the background, or initial guess, for the next time step. This process is summarized in fig. 3.1.

The goal is to compute an optimal a posteriori estimate, $\mathbf{x}_k^a$, that is a linear combination of an a priori estimate, $\mathbf{x}_k^f$, and a weighted difference between the actual measurement, $\mathbf{y}_k^o$, and the measurement prediction, $\mathbf{H}_k\mathbf{x}_k^f$. The weight used is the same as the BLUE estimator defined in the section above. Thus, the filter is given by

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k\left(\mathbf{y}_k^o - \mathbf{H}_k\mathbf{x}_k^f\right) \tag{3.20}$$

the difference $\mathbf{y}_k^o - \mathbf{H}_k\mathbf{x}_k^f$ is called the innovation and reflects the discrepancy between the actual and the predicted measurements at time $t$. Additionally, the Kalman gain matrix is given by

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^\mathsf{T} \left(\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^\mathsf{T} + \mathbf{R}_k\right)^{-1} \tag{3.21}$$

The behavior is similar to that observed for the BLUE estimator. When the measurement error covariance, $\mathbf{R}_k$, approaches zero, the gain, $\mathbf{K}_k$, weights the innovation more heavily. On the other hand, as the a priori error estimate covariance, $\mathbf{P}_k^f$, approaches zero, the gain, $\mathbf{K}_k$, weights the innovation less heavily. The forecast and analysis steps are described below:

**Prediction step** The process starts from a previously analyzed state $\mathbf{x}_k^a$ or from the initial state $k = 0$, a Gaussian distribution characterizes the state with a mean of $\mathbf{x}_k^a$ and a covariance matrix $\mathbf{P}_k^a$.

$$\mathbf{x}_{k+1}^f = \mathbf{M}_{k+1}\mathbf{x}_k^a \tag{3.22}$$

$$\mathbf{P}_{k+1}^f = \mathbf{M}_{k+1}\mathbf{P}_k^a\mathbf{M}_{k+1}^\mathsf{T} + \mathbf{Q}_{k+1} \tag{3.23}$$

**Update step** At time $t_{k+1}$, the mean $\mathbf{x}_{k+1}^f$ is known, and covariance matrix, $\mathbf{P}_{k+1}^f$ is just calculated, as well as the assumption of a Gaussian distribution. In the analysis step, the state is corrected using the observation available at time $t_{k+1}$. This comes from the BLUE as discussed earlier in the dynamical context and gives

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}_k^\mathsf{T} \left(\mathbf{H}_k \mathbf{P}_{k+1}^f \mathbf{H}_k^\mathsf{T} + \mathbf{R}_{k+1}\right)^{-1} \tag{3.24}$$

$$\mathbf{x}_{k+1}^a = \mathbf{x}_{k+1}^f + \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1}^o - \mathbf{H}_k\mathbf{x}_{k+1}^f\right) \tag{3.25}$$

$$\mathbf{P}_{k+1}^a = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_k\right)\mathbf{P}_{k+1}^f \tag{3.26}$$

The predictor-corrector loop is described below:

- Initial estimates for $\mathbf{x}_k^a$ and $\mathbf{P}_k^a$
- While $k \leq k_{\text{total}}$
  - Time Update ("Predict")
    1. Project the state ahead $\mathbf{x}_{k+1}^f = \mathbf{M}\mathbf{x}_k^a$
    2. Project the error covariance ahead $\mathbf{P}_{k+1}^f = \mathbf{M}\mathbf{P}_k^a\mathbf{M}^\mathsf{T} + \mathbf{Q}$
  - Measurement Update ("Correct")
    1. Compute the Kalman gain $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^f \mathbf{H}_k^\mathsf{T} \left(\mathbf{H}_k \mathbf{P}_{k+1}^f \mathbf{H}_k^\mathsf{T} + \mathbf{R}_{k+1}\right)^{-1}$
    2. Update estimate with measurement $\mathbf{x}_{k+1}^a = \mathbf{x}_{k+1}^f + \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1}^o - \mathbf{H}_k\mathbf{x}_{k+1}^f\right)$
    3. Update the error covariance $\mathbf{P}_{k+1}^a = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_k\right)\mathbf{P}_{k+1}^f$

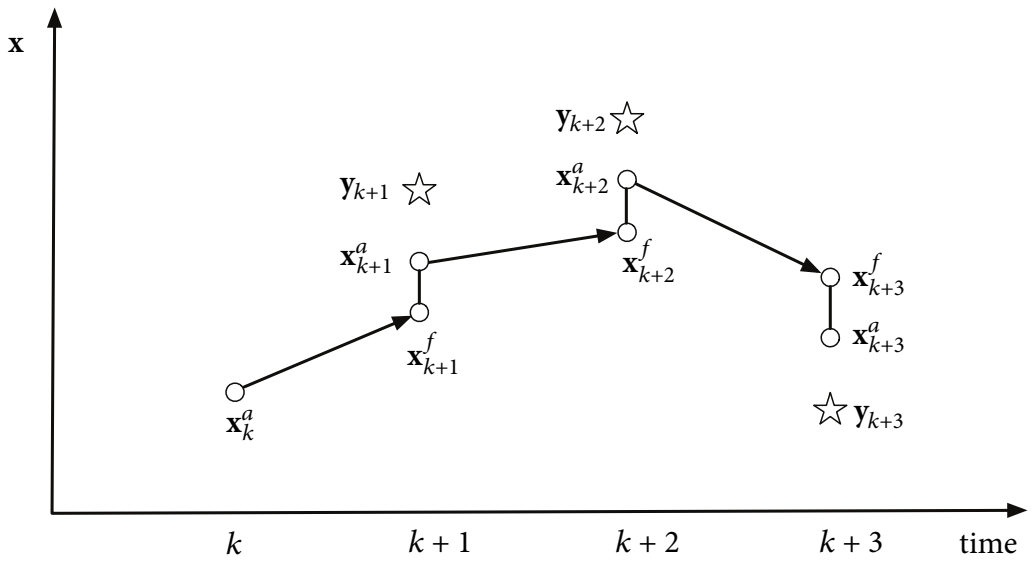



**Figure 3.1:** Sequential assimilation scheme for the KF. The x-axis denotes time; the y-axis denotes the values of the state and observation vectors (adapted from [1]).

### 3.2.3 Notes on the Forecast Error Covariance Matrix

The modeling of the covariance matrices of the forecast error $\mathbf{P}^f$ and the observation error $\mathbf{R}$ is a delicate subject. One of the problems is the huge dimensions of these matrices, $n \times n$ and $m \times m$ respectively, for some particular problems, for example, weather forecasting or turbulent flow control problems. The other main problem is the need to have a good approximation quality, depending on the test case.

Thus, in forecast error covariance modeling [45], compromises have to be made to produce a computationally viable model. Since the true background state is, by definition, unknown, two approaches are used. On the one hand, the information about forecast error statistics can be separated from the innovation statistics, or, on the other hand, statistics can be derived for a surrogate quantity. Both approaches require assumptions, for example, about the observation error's statistical properties.

The "separation" approach can be addressed by running an ensemble of randomly perturbed predictions drawn from relevant distributions, essentially a Monte Carlo method. This method of generating surrogate fields of background error is strongly related to the EnKF [8] where the forecast error covariance matrix is approximated by the sample covariance matrix of the ensemble.

Other approaches for modeling the $\mathbf{P}^f$ matrix by reduced bases, factorization, spectral methods [1, 46] or simple assumptions about the matrix structure based on the physics of the problem [6, 47]. In this Thesis, the latter approach will be used when necessary.

## 3.3 Variational Methods

Unlike sequential assimilation, which emanates from statistical estimation theory, variational assimilation is based on optimal control theory which itself emanates from the calculus of variations. The analyzed state is not defined as the one that maximizes a certain probability density function, or equivalently minimizes the error covariances, but as the one that minimizes a cost function. Thus, it requires numerical optimization techniques, which rely on the gradient of this cost function.

### 3.3.1 3D-Var

The 3D variational analysis, or 3D-Var, is performed for a steady-state problem. A scalar quadratic cost function is defined

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^f)^\mathsf{T}\mathbf{P}_f^{-1}(\mathbf{x} - \mathbf{x}^f) + \frac{1}{2}(\mathbf{H}\mathbf{x} - \mathbf{y}^o)^\mathsf{T}\mathbf{R}^{-1}(\mathbf{H}\mathbf{x} - \mathbf{y}^o) \tag{3.27}$$

where $\mathbf{x}$ is the state vector, $\mathbf{x}^f$ is the forecast state vector and $\mathbf{y}^o$ is the measured state.

This quadratic cost function attempts to strike a balance between some a priori knowledge about a forecast (using a model or a state from a previous time) and the actual measured or observed state. Additionally, it can be decomposed into the forecast and observation errors

$$J(\mathbf{x}) = J^f(\mathbf{x}) + J^o(\mathbf{x}) \tag{3.28}$$

$$J^f(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^f)^\mathsf{T}\mathbf{P}_f^{-1}(\mathbf{x} - \mathbf{x}^f) \tag{3.29}$$

$$J^o(\mathbf{x}) = \frac{1}{2}(\mathbf{H}\mathbf{x} - \mathbf{y}^o)^\mathsf{T}\mathbf{R}^{-1}(\mathbf{H}\mathbf{x} - \mathbf{y}^o) \tag{3.30}$$

thus, it represents the sum of the weighted forecast and observation deviations. Then, the analysis state $\mathbf{x}^a$ is obtained by minimizing $J(\mathbf{x})$, or

$$\mathbf{x}^a = \underset{\mathbf{x}}{\operatorname{argmin}}\, J(\mathbf{x}) \tag{3.31}$$

An expression for $\mathbf{x}^a$ can be devised by taking the gradient of the cost function in eq. (3.27), assuming that $\mathbf{H}$ is linear and equating it to zero

$$\nabla J(\mathbf{x}^a) = \mathbf{P}_f^{-1}(\mathbf{x}^a - \mathbf{x}^f) - \mathbf{H}^\mathsf{T}\mathbf{R}^{-1}(\mathbf{y}^o - \mathbf{H}\mathbf{x}^a) = 0 \tag{3.32}$$

It can be shown that if the definition of the analysis vector in eq. (3.18) is multiplied by $\mathbf{P}_f^{-1}$ and then eq. (3.32) is substituted, the gain matrix is given by

$$\mathbf{K} = \mathbf{P}^f\mathbf{H}^\mathsf{T}(\mathbf{H}\mathbf{P}^f\mathbf{H}^\mathsf{T} + \mathbf{R})^{-1} \tag{3.33}$$

which is the same as the gain matrix used in BLUE. Thus, the statistical analysis performed in the previous sections is reproduced in the 3D-Var framework.

In general, the 3D-Var approach requires the solution of the minimization problem using a gradient-based optimization, e.g., the gradient descent method in order to find $\mathbf{x}^a$, and can be extended easily in the case where $\mathbf{H}$ is non-linear but requires the calculation of the Jacobian matrix $\frac{\partial H_i}{\partial \mathbf{x}_j^a}$.

### 3.3.2 4D-Var

In time-dependent systems, the influence of model dynamics must be taken into account. Thus the 4D-Var approach is followed. The cost function is given by

$$J(\mathbf{x}) = J^f(\mathbf{x}) + J^o(\mathbf{x}) \tag{3.34}$$

where the forecast term $J^f(\mathbf{x})$ is calculated at $k = 0$

$$J^f(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^f)^\mathsf{T}\mathbf{P}_f^{-1}(\mathbf{x}_0 - \mathbf{x}_0^f) \tag{3.35}$$

while the observation term $J^o(\mathbf{x})$ is given by

$$J^o(\mathbf{x}) = \frac{1}{2}\sum_{k=0}^{K}\mathbf{d}_k^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{d}_k \tag{3.36}$$

where $\mathbf{d}_k$ is the innovation vector

$$\mathbf{d}_k = \mathbf{y}_k^o - \mathbf{H}_k\mathbf{M}_k\mathbf{M}_{k-1}\ldots\mathbf{M}_2\mathbf{M}_1\mathbf{x} \tag{3.37}$$

Then, the gradient of the cost function can be calculated

$$\nabla J(\mathbf{x}) = \mathbf{P}_f^{-1}(\mathbf{x}_0 - \mathbf{x}_0^f) - \sum_{k=0}^{K}\mathbf{M}_1^\mathsf{T}\mathbf{M}_2^\mathsf{T}\ldots\mathbf{M}_{k-1}^\mathsf{T}\mathbf{M}_k^\mathsf{T}\mathbf{H}^\mathsf{T}\mathbf{R}_i^{-1}\mathbf{d}_i \tag{3.38}$$

The calculation of the $\mathbf{M}_k^\mathsf{T}$ and $\mathbf{H}_k^\mathsf{T}$ is performed using the adjoint approach. The algorithm can be summarised as follows:

1. For a given initial condition, $\mathbf{x}_0$, the (nonlinear) state equation is integrated forward in time

2. From the final condition, the adjoint equations are integrated backward in time

3. The cost function gradient is calculated

4. The gradient is used in an optimization algorithm to find the $\mathbf{x}^a = \text{argmin}_\mathbf{x} J(\mathbf{x})$

Both the 3D-Var and 4D-Var procedures are described in fig. 3.2.

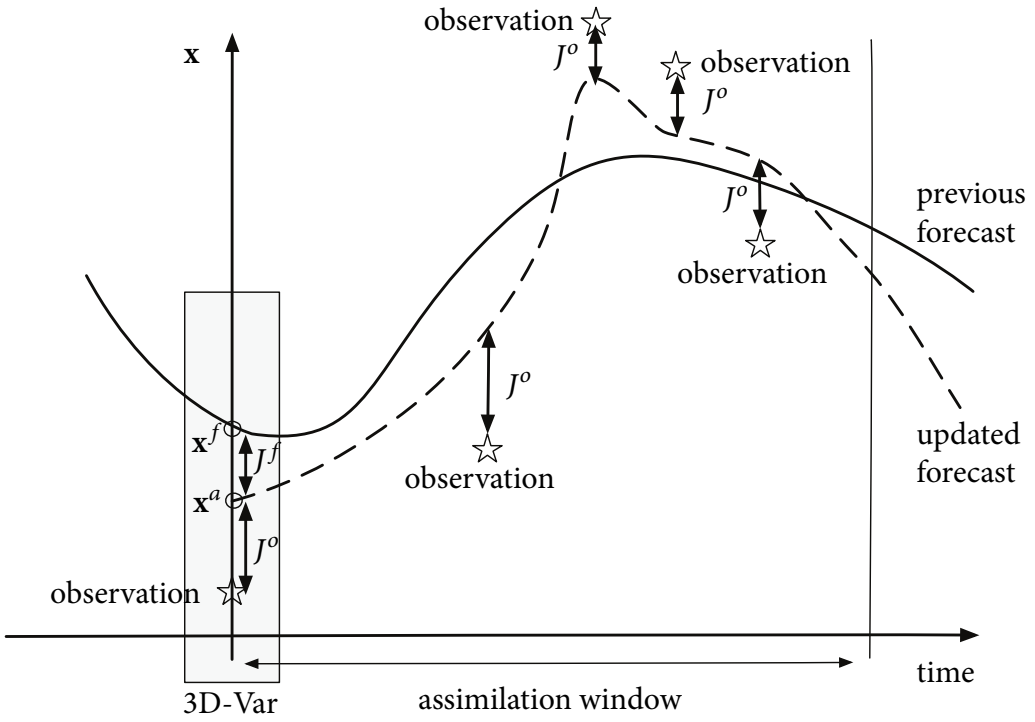



**Figure 3.2:** The 3D-Var and 4D-Var algorithms (adapted from [1]).

## 3.4 Nudging

A simpler method called nudging, or Newtonian relaxation, aims to dynamically adjust the model toward the observations by simply inserting a feedback term into the model equation that is proportional to the observation and model misfit (similar to the innovation term) and nudges the model state toward the observations [48], as shown in fig. 3.3. The dynamic model, assuming linear operators, becomes

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x} + \mathbf{K}\left(\mathbf{y}^o - \mathbf{H}\mathbf{x}\right) \tag{3.39}$$

with an initial condition of $\mathbf{x}\left(t = 0\right) = \mathbf{x}_0$. Additionally, the nudging method can be considered a state observer, a well-established method in the control theory field, if both $\mathbf{M}$ and $\mathbf{H}$ operators are matrices.

The nudging method can be derived both from variational and sequential approaches. In the former, the nudging method can be considered a compromise between the minimization of the energy of the system and the quadratic distance to the observations [49]. In the latter, its formulation in a discrete time system, if the nudging gain matrix is selected using BLUE with $\mathbf{R} = \mathbf{0}$ for every time-step, is equivalent to the KF.

The main advantage of nudging is the simplicity of its implementation since there is no need for adjoint development, nor for a complex BLUE computation step. There is only the need to implement an observation operator and a slight modification of the model equation to add a feedback term.

Of course, the improvement that nudging provides compared to the base model is case-dependent; for instance, for an observation error that is small, with no correlation in space, and constant over time, the nudging method, can produce sufficient results [1, 50]. On the contrary, the nudging method with a diagonal $\mathbf{K}$ can produce poor results if, for example, the observation errors are correlated in space, or if the errors vary much in time [50].

In this kind of framework, the specification of $\mathbf{K}$ is crucial; it can be selected diagonal or even scalar, but the effect of it depends on the structure of observation errors, their correlation in space, or their variance in time. But its tuning is not automated and relies on numerical experimentation. The BLUE approach can be used if information about observation errors is available, or an optimal nudging approach [51, 52] can be used to minimize with respect to $\mathbf{K}$, a cost function that represents the error between model and observations as follows,

$$J(\mathbf{K}) = J^o(\mathbf{x}(\mathbf{K})) + J^f(\mathbf{K}) \tag{3.40}$$

where $J^o(\mathbf{x}(\mathbf{K}))$ measures the observation misfit $\mathbf{x} - \mathbf{x}^o$ and $J^f(\mathbf{K})$ is a regularization term for $\mathbf{K}$, e.g., $\mathbf{K} - \mathbf{K}_0$ where $\mathbf{K}_0$ is an initial guess for $\mathbf{K}$.

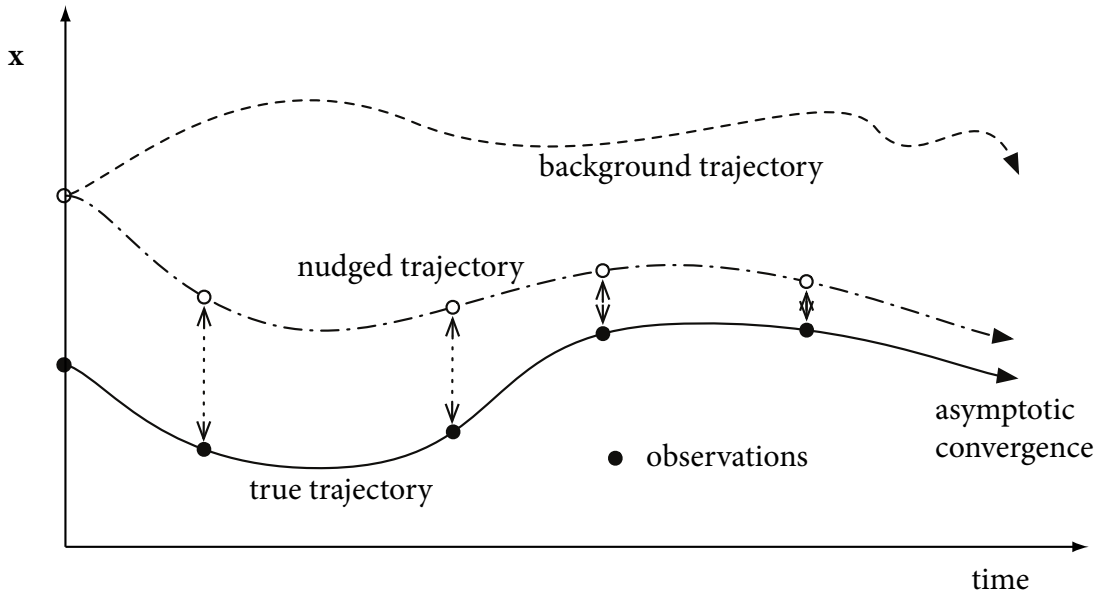



**Figure 3.3:** Schematic representation of the nudging method (adapted from [1]).

# Chapter 4

# Application 1: 1-D Shock Tube Problem

In this Chapter, Data Assimilation methods are applied to a very common test case, the Sod Shock Tube. This test case has been extensively used to verify CFD codes [53, 54] as an analytical solution to this problem exists. Firstly, a computational code was developed using Fortran 95, based on the code provided in the Computational Fluid Dynamics course in the 7th semester at the School of Mechanical Engineering of the National Technical University of Athens. The developed code is modular and can run in parallel on shared memory computers, e.g., commercial laptops or desktop computers, using the OpenMP API (*Application Programming Interface*) [55].

After the verification of the solver and the investigation of different orders of the flux reconstruction schemes and/or the usage of gradient limiters, the nudging method is implemented. Initially, the analytical solution of the problem is used as input for the nudging forcing term. Next, the results of simulations with finer spatial discretization are used to improve the results of coarser grids. Finally, the optimal nudging method is used to optimize the gain matrix and utilize the adjoint fields to calculate the sensitivity derivatives.

## 4.1 Problem Description

The Sod Shock tube problem describes the 1-D propagation of density and pressure discontinuities, a Riemann problem, inside an tube of infinite length that contains an initially stationary, inviscid compressible ideal gas as shown in fig. 4.1. The time evolution of the problem is governed by the 1-D Euler equations in conservative form as described in eqs. (2.1) and (2.3).

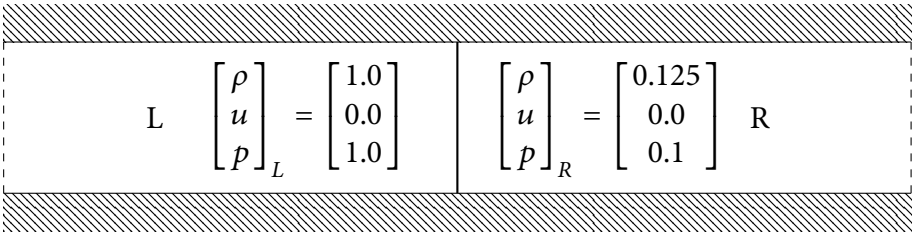



**Figure 4.1:** Problem definition.

The system defined in eq. (2.3) is a system of hyperbolic quasilinear PDEs and can be solved using the method of characteristics, even though, for computational reasons, an approximate Riemann solver is used for the numerical solution. The information propagation is described by three characteristics, the rarefaction wave, the contact discontinuity, and the shock wave, as

shown in fig. 4.2. The time evolution of the three characteristics splits the fluid domain into five discrete regions, where the fluid states can be calculated analytically [21, 23, 53].

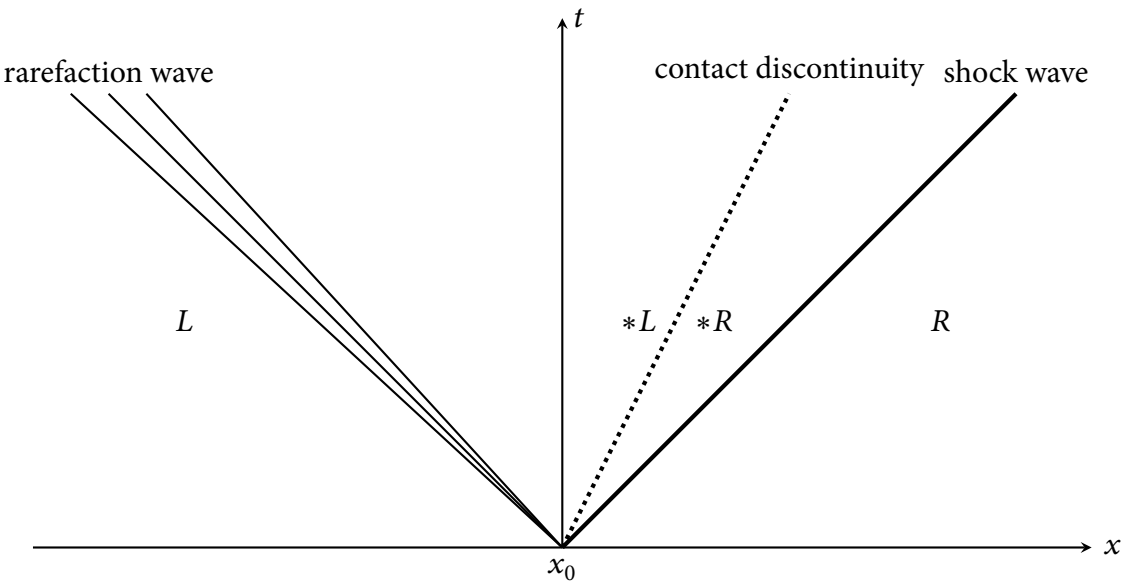



**Figure 4.2:** The 3 system characteristics (adapted from [21]).

### 4.1.1 Analytical Solution

The fluid domain is separated into five regions with discrete states fig. 4.3. In regions 1 & 5, the ideal gas states are known and equal to the initial left and right conditions, respectively. Then, shock jump conditions are used along the discontinuities to account for the information propagation in the direction of the system characteristics [21].

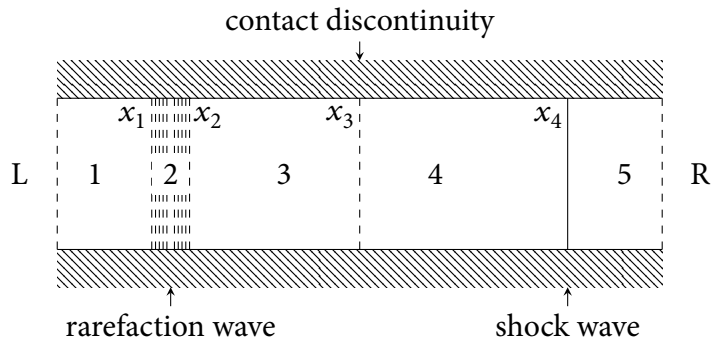



**Figure 4.3:** The five regions of the fluid domain with discrete states.

Before the analytical solution is laid out, two auxiliary variables are defined, using the heat capacity ratio $\gamma$.

$$\Gamma = \frac{\gamma - 1}{\gamma + 1}$$
$$\beta = \frac{\gamma - 1}{2\gamma}$$

Then, each discontinuity is investigated separately, from left to right:

- Regions 1-3: The rarefaction wave is present, and the relation between the states at the wave head and tail can be described as an isentropic process [21]:

$$\rho_3 = \rho_1 \left( \frac{p_3}{p_1} \right)^{1/\gamma} \tag{4.1}$$

$$u_3 = (p_1^\beta - p_3^\beta) \sqrt{\frac{(1-\Gamma^2) p_1^{1/\gamma}}{\Gamma^2 \rho_1}} \tag{4.2}$$

- Regions 3-4: The contact discontinuity is present, and only a density discontinuity is observed:

$$u_3 = u_4 \tag{4.3}$$

$$p_3 = p_4 \tag{4.4}$$

- Regions 4-5: The shock wave is present, and it is described by Rankine-Huganiot relations [56]:

$$\rho_4 = \rho_5 \frac{p_4 + \Gamma p_4}{p_5 + \Gamma p_5} \tag{4.5}$$

$$u_4 = (p_4 - p_5) \sqrt{\frac{1-\Gamma}{\rho_5 (p_4 + \Gamma p_5)}} \tag{4.6}$$

Combining eqs. (4.2) to (4.4) and (4.6), the following non-linear equation for the calculation of $p_3$ is devised:

$$p_3 = (p_1^\beta - p_3^\beta) \sqrt{\frac{(1-\Gamma^2) p_1^{1/\gamma}}{\Gamma^2 \rho_1}} + p_5 \sqrt{\frac{1-\Gamma}{\rho_5 (p_3 + \Gamma p_5)}} \tag{4.7}$$

Then, eq. (4.7) is solved iteratively using the secant method. Finally, inside region 2, where the rarefaction wave is located, the state is described analytically [21]:

$$\rho_2(x) = \rho_1 \left( 1 - \frac{\gamma - 1}{2} \frac{u_2(x)}{c_1} \right)^{2/\gamma - 1} \tag{4.8}$$

$$u_2(x) = \frac{2}{\gamma + 1} \left( c_1 + \frac{x - x_0}{t} \right) \tag{4.9}$$

$$p_2(x) = p_1 \left( 1 - \frac{\gamma - 1}{2} \frac{u_2(x)}{c_1} \right)^{2/\gamma - 1} \tag{4.10}$$

where $c_1 = \sqrt{\gamma \frac{p_1}{\rho_1}}$ is the sound speed at region 1 and $x_0$ the initial position of the contact discontinuity.

At this point, the methodology for determining the five states is developed. The only necessity is determining the position of the different regions in time, described by the waves' heads and tails. This can easily be achieved using the three system characteristics, assuming $t_0 = 0$. Firstly, the head of the rarefaction wave is moving with the speed of sound in the left non-disturbed state:

$$x_1 = x_0 - c_1 t \tag{4.11}$$

Then, the position of the tail of the rarefaction wave is moving in the characteristic described moving with a speed of $u_3 - c_3$, thus:

$$x_2 = x_0 + (u_3 - c_3)t \tag{4.12}$$

Next, the position of the contact discontinuity is moving with the speed of the state after the rarefaction wave to the right:

$$x_3 = x_0 + u_3 t \tag{4.13}$$

Finally, the position of the shock front is calculated using the shock jump conditions:

$$w = c_5 \sqrt{1 + \frac{\gamma + 1}{2}\left(\frac{p_4}{p_5} - 1\right)} \tag{4.14}$$

$$x_4 = x_0 + wt \tag{4.15}$$

The above methodology can be easily implemented in a computational code. In fig. 4.4 the distribution of the primitive variables $\mathbf{W} = \begin{bmatrix}\rho & u & p\end{bmatrix}^\mathsf{T}$ is shown at $t = 0.42\ s$ for $x \in [-5, 5]$ and $x_0 = 0$.

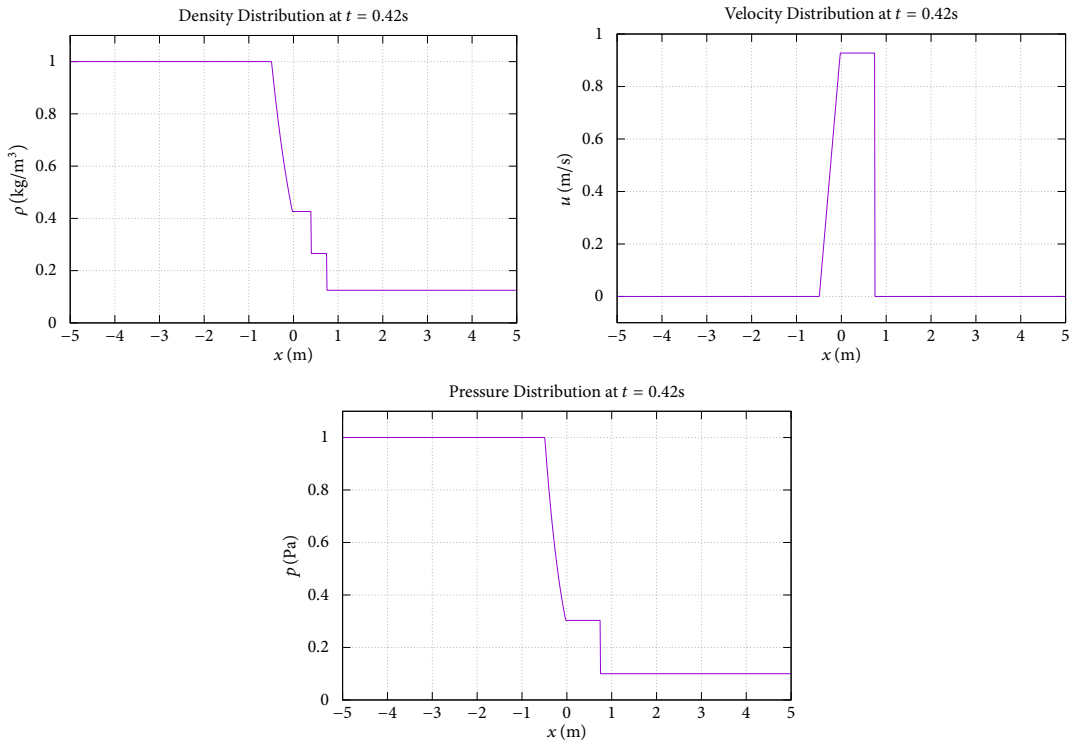



**Figure 4.4:** Analytical solution of Sod Shock problem at $t = 0.42$s.

### 4.1.2 Numerical Solution using the Finite Volume Method

After developing a Fortran 95 code that uses the analytical solution, a numerical code was developed based on the code provided in the Computational Fluid Dynamics course in the 7th semester at the School of Mechanical Engineering of the National Technical University of Athens. The existing code uses the Finite Volume method with the Godunov scheme to solve the conservative 1-D Euler equations, using the Roe approximate Riemann solver with both 1st and 2nd order MUSCL discretization scheme [26], with the addition of the van Leer [27] flux limiter.

The code is cell-centered and runs in parallel on shared memory computing systems (laptops and desktop computers) using the OpenMP API.

The simulation is performed in a computational grid like the one shown in fig. 4.5, using the ghost cell to represent the domain boundaries. In order to take into account the infinite dimensions of the shock tube, symmetry boundary conditions are used, and the total simulation duration is selected so that neither the rarefaction nor the shock wave reach the boundaries during the simulation. The total domain length is $L = 10$m, with $x \in [-5, 5]$ m and the initial discontinuity placed at $x_0 = 0$.

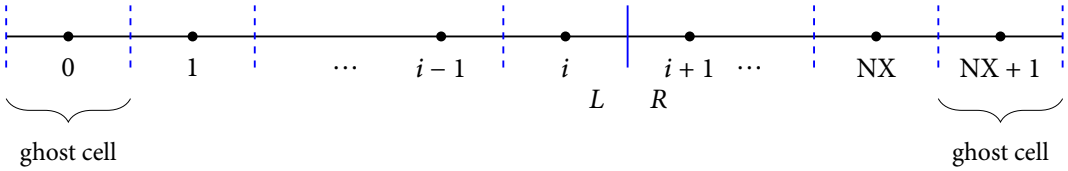



**Figure 4.5:** Computational grid.

The Godunov scheme uses an explicit 1st order time integration scheme (Euler method), thus a stability condition must be taken into account. According to literature [21, 23], the CFL (*Courant–Friedrichs–Lewy*) convergence condition for the Godunov scheme is:

$$\mathrm{CFL} = \frac{\Delta t}{2\Delta x} \max |\lambda| \leq 1 \tag{4.16}$$

where $\max |\lambda|$ is the max of the absolute values of the system eigenvalues.

The space distribution of the primitive system variables is presented in fig. 4.6 at $t = 0.42$s, showing the effect on the accuracy of the different implemented schemes. The domain was discretized into 1000 points, leading to $\Delta x = 10^{-2}$m and a time step of $\Delta t = 10^{-4}$s was selected. Additionally, the selected heat capacity ratio is $\gamma = 1.4$. Thus, the maximum system eigenvalue is $u_* + c_L$, with $u_* \approx 0.9$m/s and $c_L = \sqrt{\gamma \frac{p_L}{\rho_L}} \approx 1.18$m/s. The resulting CFL is satisfied, as $\mathrm{CFL} \approx 10^{-2} \ll 1$.

It is evident that although the code generally can follow the fluid behavior, it fails to capture the shock behaviors accurately, as shown in fig. 4.7. In general, the 1st order scheme shows a more diffusive behavior, while the 2nd order scheme has similar behavior, a little more accurate near the shock but fails near the right boundary, which is corrected using the van Leer flux limiter.

Next, the effect of spatial resolution is investigated. The problem is solved using a 1st order reconstruction scheme for 500, 1000, 2000, 4000 και 10000 points in space, with the same time step, and the results are presented in figs. 4.8 and 4.9. As expected, the increase in spatial resolution increases the achieved accuracy and allows for better capture of the shock behaviors as shown in fig. 4.9.

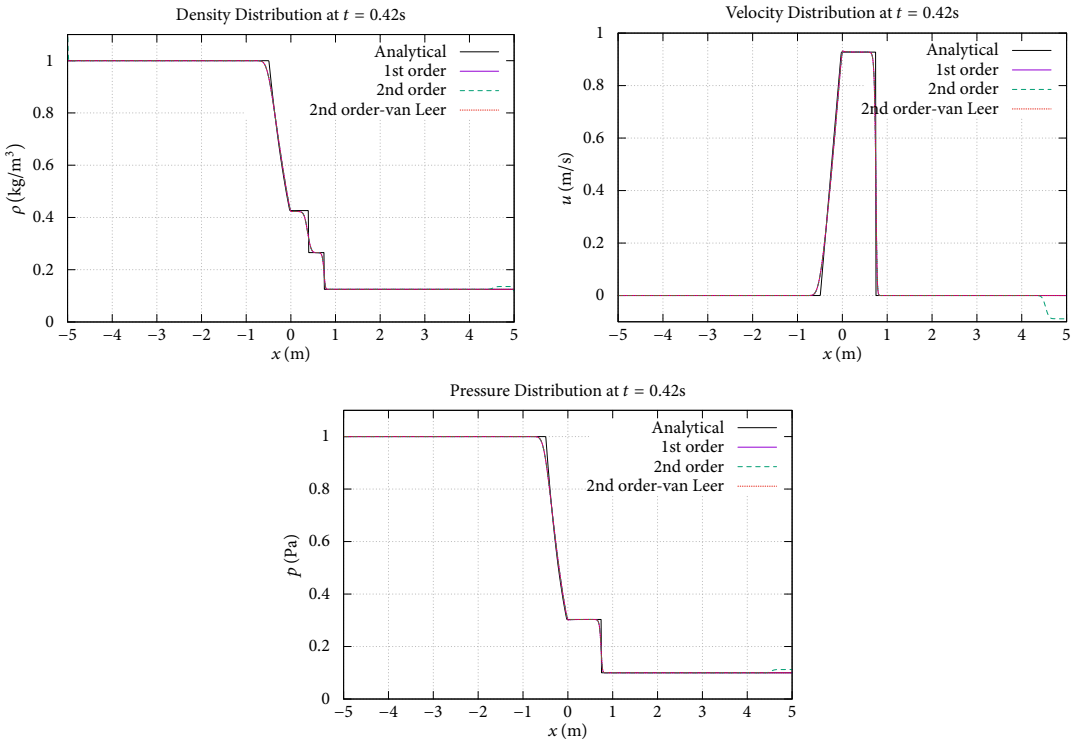



**Figure 4.6:** Numerical solution of Sod Shock problem at $t = 0.42$s using the Finite Volume method with 1000 grid volumes.

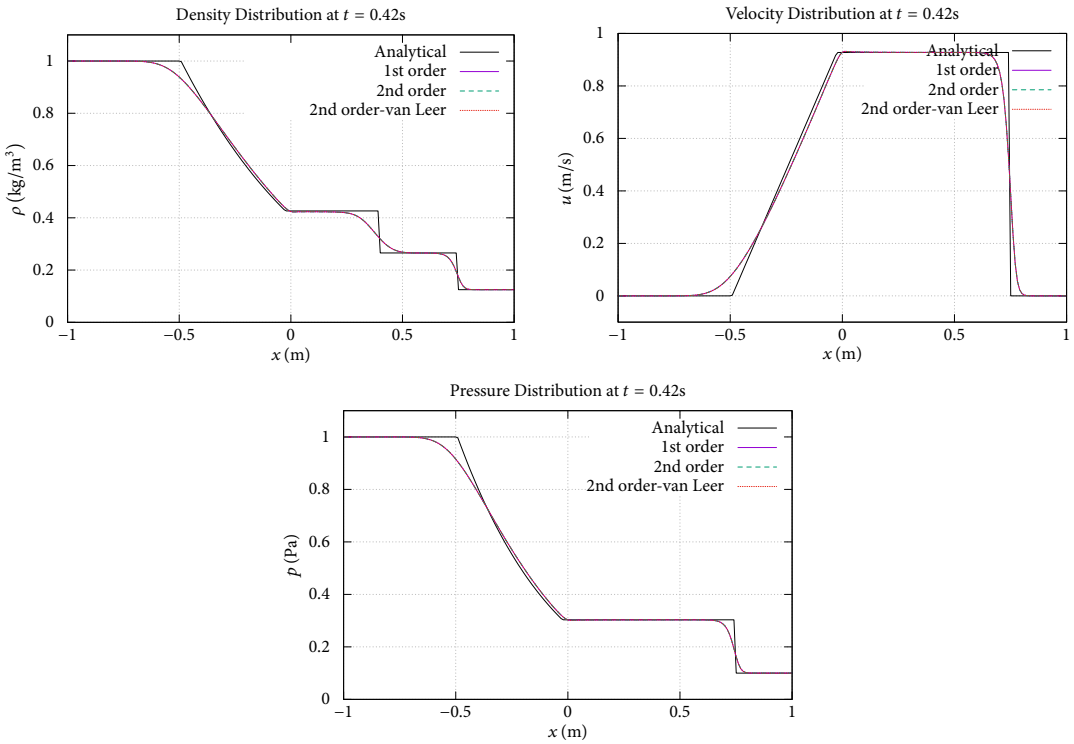



**Figure 4.7:** Numerical solution of Sod Shock problem for $x \in [-1, 1]$ at $t = 0.42$s using the Finite Volume method with 1000 grid volumes.

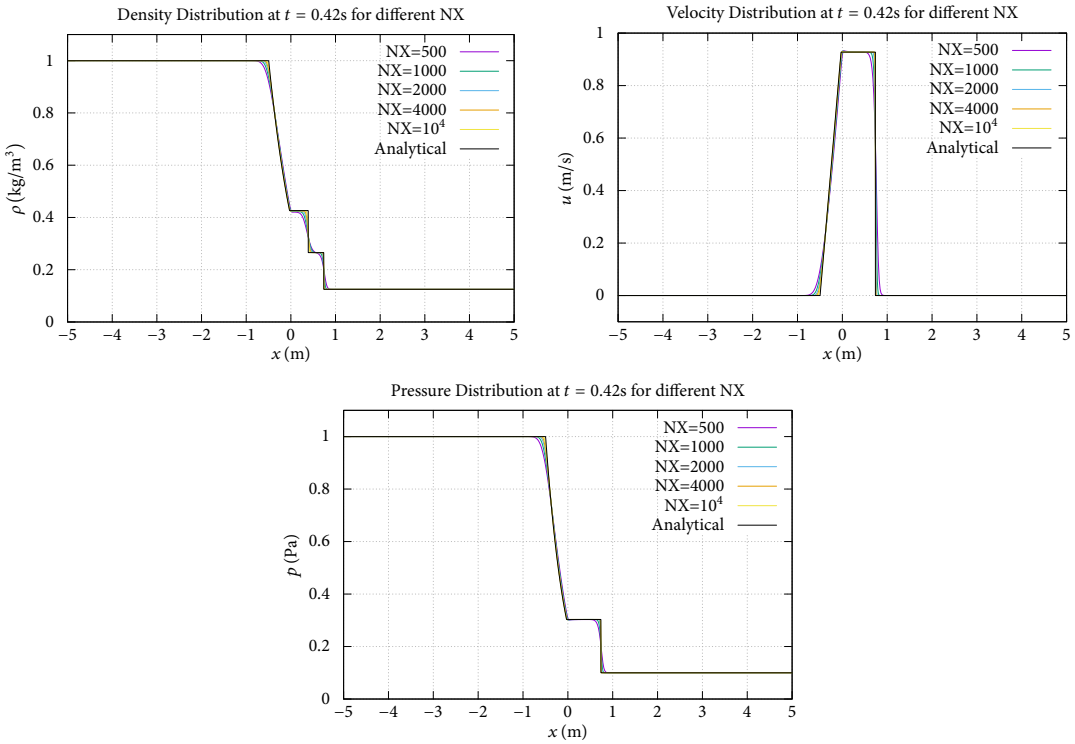



**Figure 4.8:** Numerical solution of Sod Shock problem for varying spatial resolution at $t = 0.42$s.

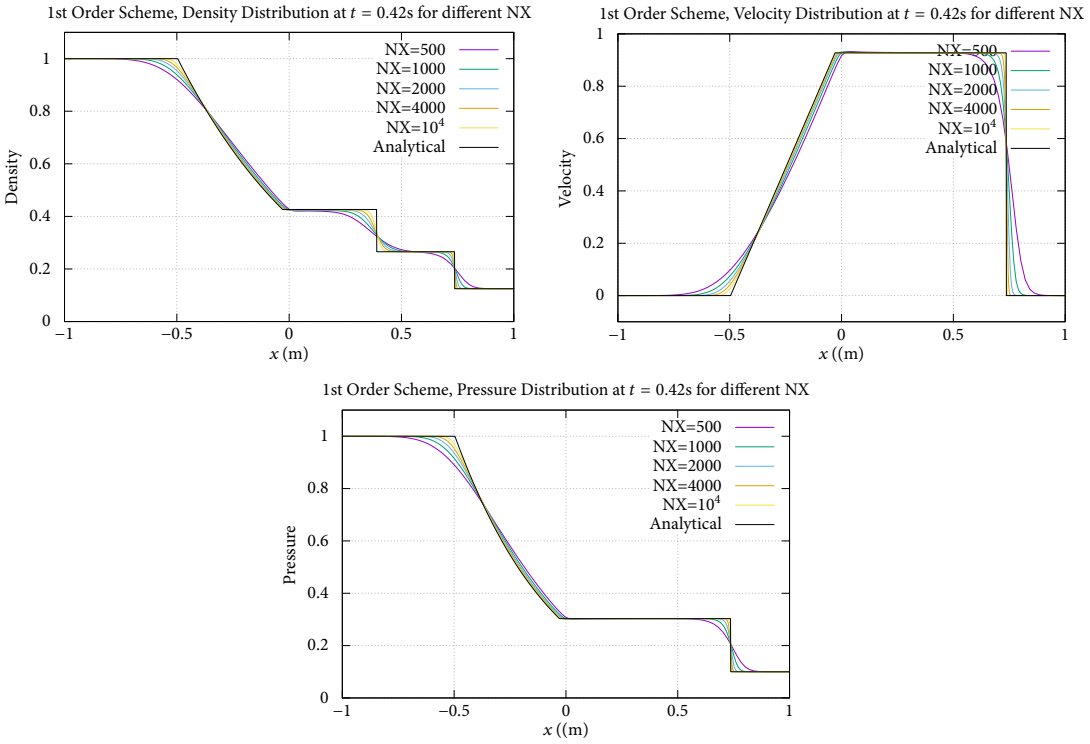



**Figure 4.9:** Numerical solution of Sod Shock problem for varying spatial resolution for $x \in [-1, 1]$ at $t = 0.42$s.

In order to quantify the accuracy improvement, an accuracy metric is defined in eq. (4.17), namely the RMSE (*Root Mean Square Error*) between the numerical and analytical solution. The

time evolution of the RMSE of the primitive variables for varying spatial resolution is shown in fig. 4.10. After a while, the RMSE tends to converge to oscillating around some fixed value. It is expected that by increasing the spatial resolution, the accuracy is improved, and it is verified in the figure above, but interestingly the improvement in accuracy doesn't scale precisely as expected. For instance, by increasing the spatial resolution by an order of 10, the RMSE does not decrease by an order of 30 as expected (first order scheme means that the difference between the numerical and analytical solution is proportional to $1/\mathrm{NX}$ and thus the RMSE is proportional to $1/\sqrt{\mathrm{NX}^3}$).

$$\mathbf{RMSE} = \sqrt{\frac{\sum_{i=1}^{\mathrm{NX}} \left(\mathbf{U}_{analytical} - \mathbf{U}\right)^2}{\mathrm{NX}}} \tag{4.17}$$

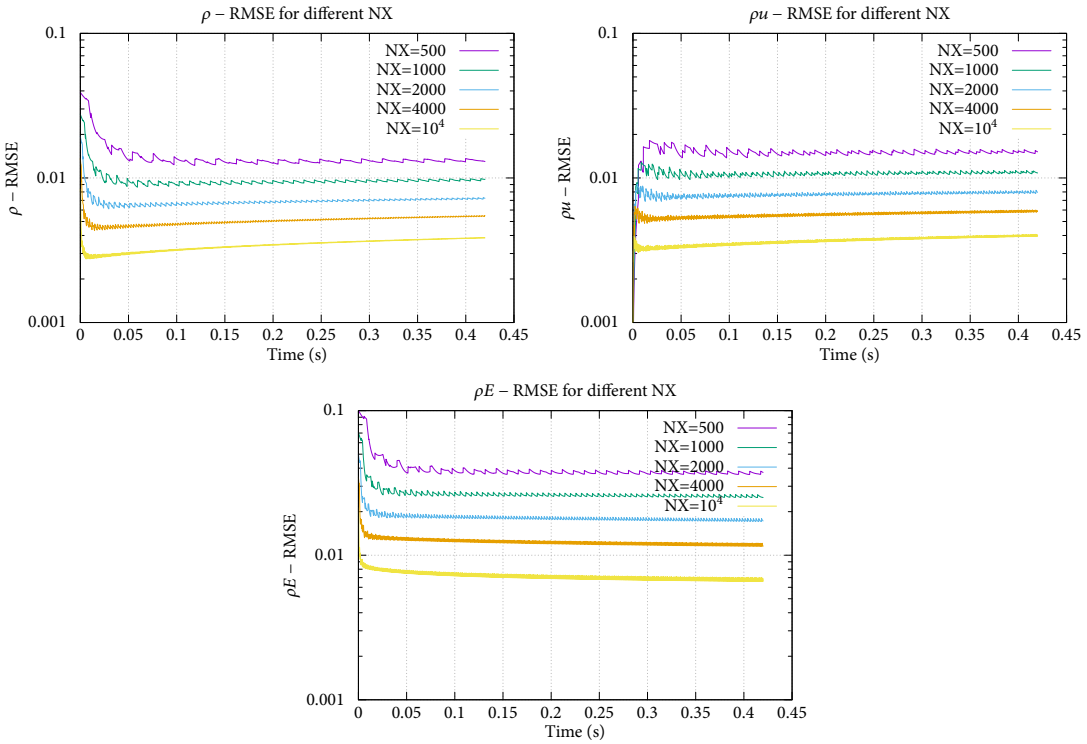



**Figure 4.10:** Time evolution of RMSE of conservative variables for varying spatial resolution in the solution of the Sod shock problem.

## 4.2 Nudging

Next, the developed code was improved by implementing a nudging method into the Euler equations in conservative form. The implementation of such a method is pretty straightforward and was done using two approaches. In the first approach, the analytical solution was used in the source term added by the nudging method. In the second approach, a numerical solution from a more refined grid was used as the additional source term, in the spirit of a multigrid method.

### 4.2.1 Nudging using the Analytical Solution

Initially, the numerical solution is improved using the nudging method by utilizing the analytical solution at the cell centers. The 1-D Euler equations in conservative form from eq. (2.1), are modified by adding a nudging source term, as shown in eq. (3.39), yielding:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = \mathbf{K}\left(\mathbf{U}_{\text{analytical}} - \mathbf{U}\right) \tag{4.18}$$

where $\mathbf{K}$ is the gain matrix. Assuming that $\mathbf{U}$ is constant inside each cell and equal to the values at the cell center, and integrating eq. (4.18), the new explicit time integration scheme is devised:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}(\mathbf{U})_{i+1/2} - \mathbf{F}(\mathbf{U})_{i-1/2}\right) + \frac{\Delta t}{\Delta x}\mathbf{K}_i\left(\mathbf{U}^n_{\text{analytical,i}} - \mathbf{U}_i^n\right) \tag{4.19}$$

Then, using the scheme of eq. (4.19), the test case is solved using a 1st order reconstruction scheme for varying gain matrices $\mathbf{K}$. The selection of the gain matrix structure and values is an interesting problem. Firstly, due to the usage of the Gudonov-like scheme, the gain matrix is $3 \times 3$ and corresponds to each face of the computational grid. This leads to the first assumption, that the gain matrix is constant for each face ($\mathbf{K}_i = \mathbf{K}$). Then, an assumption about the structure of the gain matrix was made, considering that the primary influence in the data assimilation of each variable is the difference between the respective variable and the analytical solution; the gain matrix is assumed to be diagonal. Because all primitive variables have the same order of magnitude, the diagonal elements are considered to be equal $\mathbf{K} = K\mathbf{I_3}$. The $K$ order of magnitude was selected empirically after trying different combinations. In figs. 4.11 and 4.12, the numerical solution at $t = 0.42$s is shown for different $K$ values. It is worth noting that even using a 1st order scheme, the method nudges the numerical solution to follow the analytical solution accurately. Additionally, as shown in fig. 4.12, even for $K = 1$, the numerical results are significantly closer to the analytical solution than the numerical results without using nudging.

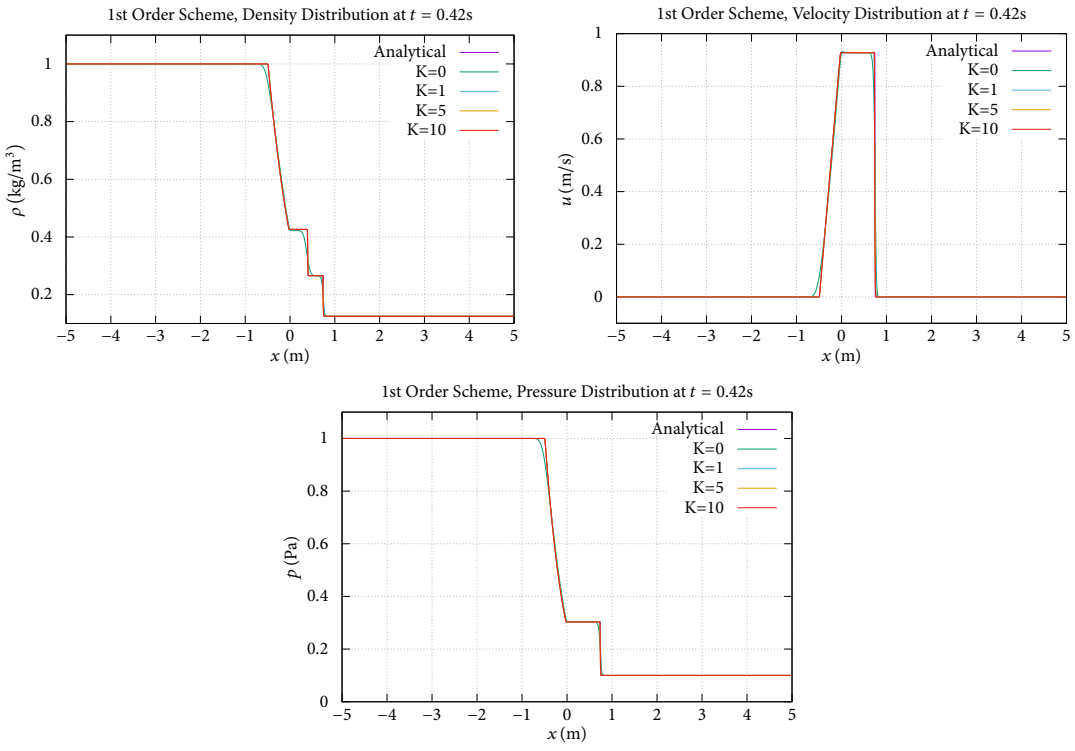



**Figure 4.11:** Numerical results using the nudging method with varying $K$ values at $t = 0.42$s.

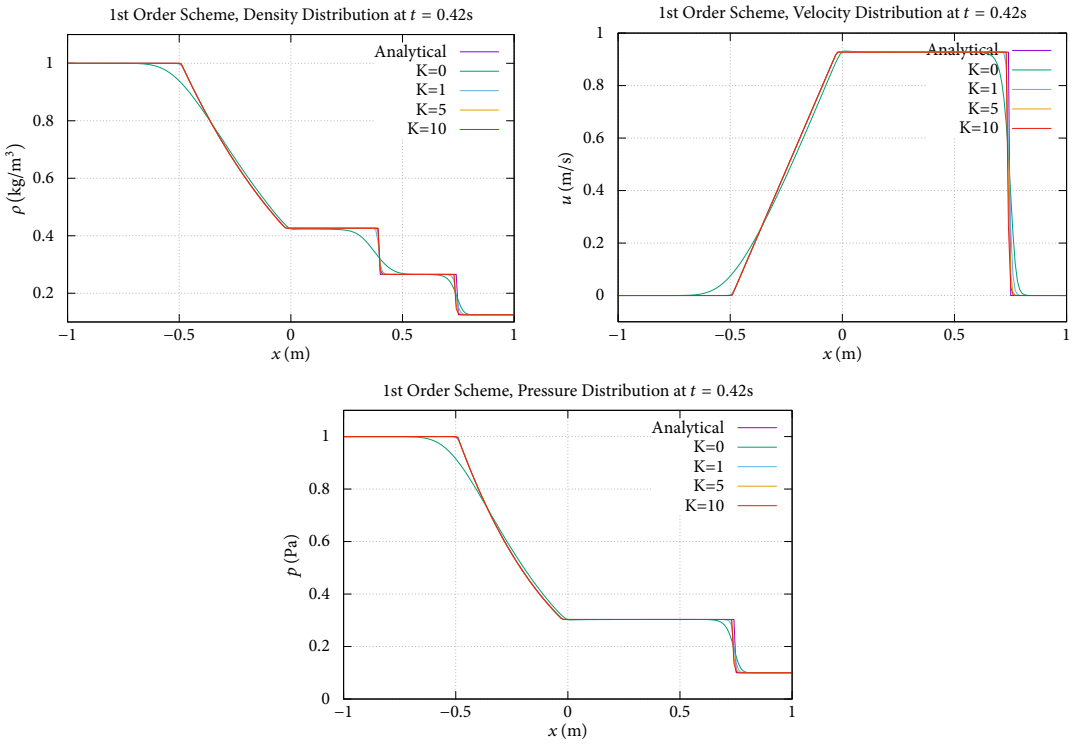



**Figure 4.12:** Numerical results using the nudging method with varying $K$ values for $x \in [-1, 1]$ at $t = 0.42$s.

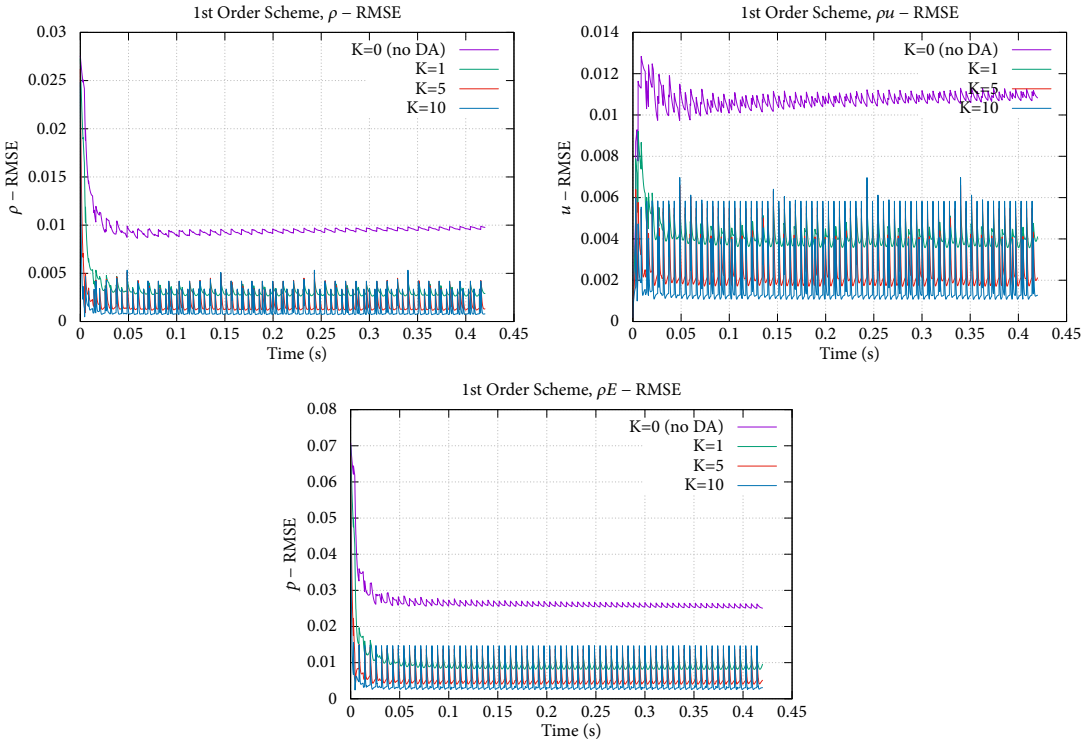



**Figure 4.13:** Time evolution of RMSE for varying $K$ values.

In fig. 4.13 the time evolution of the RMSE of the conservative values with varying $K$ values is shown. Firstly, the RMSE values converge to an oscillatory behavior around a fixed mean value

for each variable due to the 1st order time integration scheme and the usage of an approximate Riemann solver. These mean values are significantly smaller (2 to 5 times) when using the nudging method, but the amplitudes of these oscillations increase as the $K$ value increases. Also, it is worth noting that the minimum error values decrease by increasing $K$, but the max values remain the same, leading to the observation that the mean error values decrease by increasing $K$. Thus, the need to fine-tune the gain matrix arises and is further investigated in section 4.3

### 4.2.2 Nudging using Different Grids

Next, the computational code was modified to solve the problem with two different grid resolutions in parallel. Firstly, the problem is solved in a fine grid with NXF grid points, and then the problem is solved in a coarse grid with NXC grid points (NXC < NXF), using the solution of the fine grid as the ground truth. The same time step was used for both solutions and ensuring the CFL condition holds for both grids. It is worth noting that the computational cost is proportional to the grid resolution for the same time step.

The object of this method is to use the more computationally expensive solution from the fine grid in order to improve the computationally cheaper solution in the coarse grid and observe the improvement in the final solution. Initially, the 1-D Euler equations, as presented in eq. (2.1) are solved in the grid with the fine resolution:

$$\frac{\partial \mathbf{U}_{\text{fine}}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U}_{\text{fine}})}{\partial x} = \mathbf{0} \tag{4.20}$$

Then, the 1-D Euler equations with a nudging source term, as presented in eq. (4.18) are solved in the grid with the coarse resolution, using the numerical solution of the fine grid. Both grids are selected in a way that all of the coarse grid cell centers coincide with a subset of the fine grid cell centers. Thus, no explicit mapping is used, and the solution from the fine grid is used "as-is" in the cell centers common to the coarse and fine grids. Thus, eq. (4.18) becomes

$$\frac{\partial \mathbf{U}_{\text{coarse}}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U}_{\text{coarse}})}{\partial x} = \mathbf{K}\,(\mathbf{U}_{\text{fine}} - \mathbf{U}_{\text{coarse}}) \tag{4.21}$$

Finally, the analytical solution and the RMSE are calculated for both the coarse and fine grids.

The procedure described above was used to investigate the effect of using different (NXC, NXF) pairs and compare the results with the numerical solution without nudging, using a grid resolution that is one order of magnitude $(\text{NX} = 10^4)$ larger than the pair and, as shown in fig. 4.10 is more accurate.

The time evolution of the RMSE of the primitive variables for different (NXC, NXF) pairs is presented in figs. 4.14 to 4.16. Firstly, for all the (NXC, NXF) pairs, the resulting RMSE is smaller than the one achieved using an extremely finer grid $(\text{NX} = 10^4)$ throughout the whole duration of the simulation. Additionally, as evident by the RMSE values for each component at the end of the simulation ($t = 0.42$s), the maximum RMSE decreases as the fine grid resolution of the pair is increased, meaning that the achieved accuracy is increased as the fine grid resolution is increased. Finally, an oscillatory behavior is observed, similar to fig. 4.13, whose amplitude is increased as the $K$ value increases.

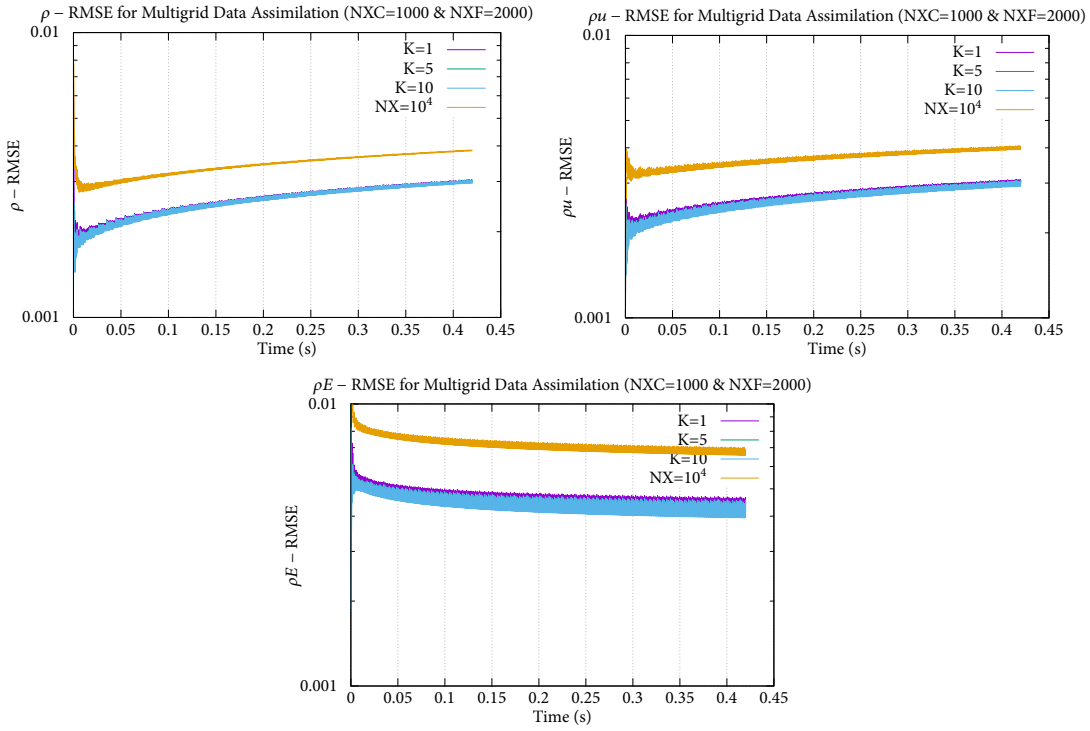



**Figure 4.14:** Time evolution of RMSE of primitive variables for nudging with NXC=1000 & NXF=2000.

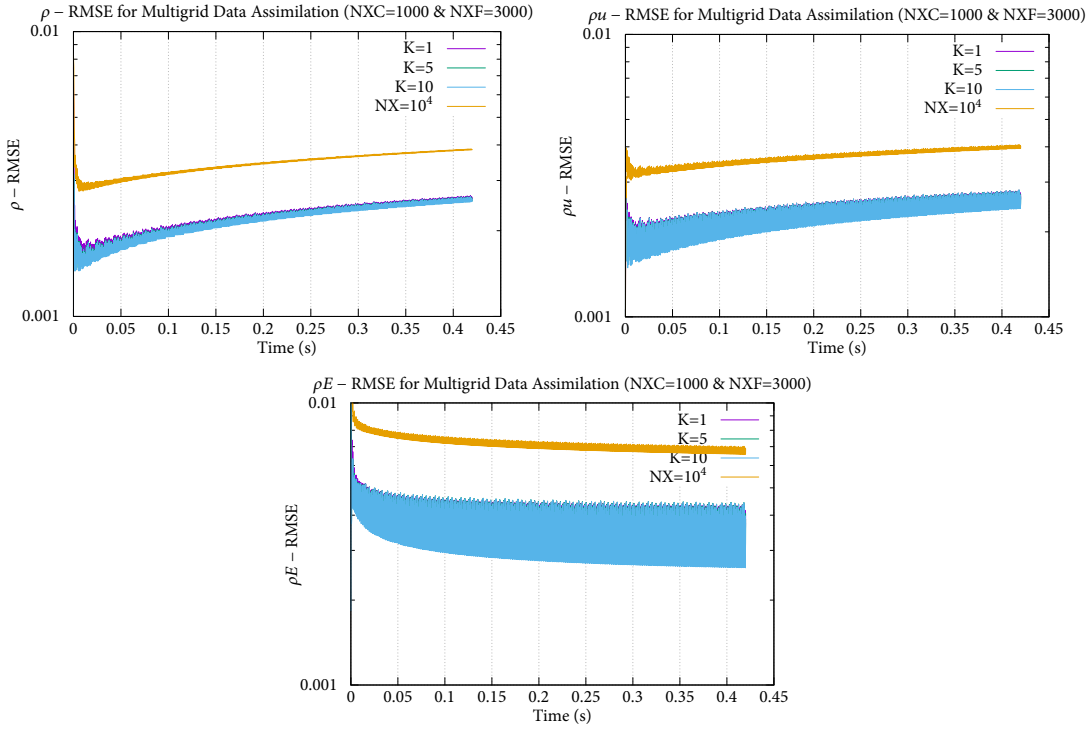



**Figure 4.15:** Time evolution of RMSE of primitive variables for nudging with NXC=1000 & NXF=3000.

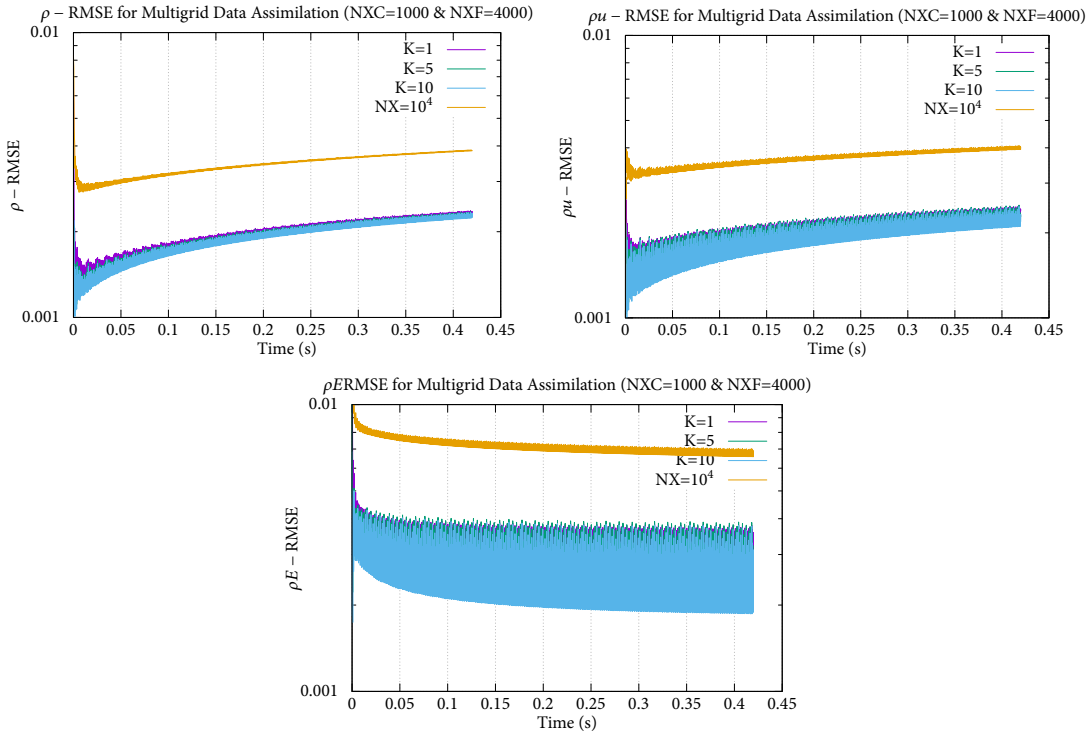



**Figure 4.16:** Time evolution of RMSE of primitive variables for nudging with NXC=1000 & NXF=4000.

## 4.3 Optimal Nudging

In the section above, the selection of the **K** gain matrix was made empirically, and several assumptions were made about the matrix structure and values. An optimization problem was devised to accommodate these shortcomings, and the final gain matrix minimizes the calculation error between the numerical and analytical solutions. This method is called optimal nudging [1] and has been used successfully in the literature [51, 52]

The main concept is the computation of an optimal gain matrix $\mathbf{K}_{\text{optimal}}$ that minimizes the computational error throughout a time interval $[t_1, t_2]$ called assimilation window and then continue the simulation using the $\mathbf{K}_{\text{optimal}}$. The following cost function **J** is used:

$$\mathbf{J}(\mathbf{U}, \mathbf{K}) = \int_{t_1}^{t_2} \int_{x} \frac{1}{2} \left( \mathbf{U} - \mathbf{U}_{\text{analytical}} \right)^2 \mathrm{d}x \, \mathrm{d}t \tag{4.22}$$

The above cost function is a $3 \times 1$ vector, and each component refers to the relevant conservative variable. It represents the error between the numerical and analytical solution for each conservative variable. Thus, each row of **K** influence the nudging of the relevant conservative variable. In order to find the **K** that minimizes **J**, the steepest descent method is used as shown below:

$$\mathbf{K}_{\text{new}} = \mathbf{K}_{\text{old}} - \eta \frac{\delta \mathbf{J}}{\delta \mathbf{K}} \tag{4.23}$$

where $\eta$ is the step size or learning rate.

### 4.3.1 Adjoint Problem Formulation

As shown in eq. (4.23), the calculation of the sensitivity derivatives $\frac{\delta \mathbf{J}}{\delta \mathbf{K}}$ is required to solve the optimization problem. There are various methods to calculate the sensitivity derivatives, e.g., various formulations of the Finite Difference method, but the Continuous Adjoint method [57, 58] was used.

The main principle is to augment the cost function with an integral of the adjoint fields $\mathbf{\Psi}$ multiplied by the primal problem equation, which is equal to zero. In this case, the 1-D Euler equations using nudging are used as the equation describing the primal problem:

$$\mathbf{J}_{\text{augmented}} = \mathbf{J} + \underbrace{\int_{t_1}^{t_2} \int_x \mathbf{\Psi} \left[ \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} - \mathbf{K}\left(\mathbf{U}_{\text{analytical}} - \mathbf{U}\right) \right] \mathrm{d}x \, \mathrm{d}t}_{0} \tag{4.24}$$

or using eq. (4.22) and Fubini's theorem:

$$\begin{aligned} \mathbf{J}_{\text{augmented}} = & \int_x \int_{t_1}^{t_2} \frac{1}{2} \left(\mathbf{U} - \mathbf{U}_{\text{analytical}}\right)^2 \mathrm{d}t \, \mathrm{d}x \\ & + \int_x \int_{t_1}^{t_2} \mathbf{\Psi} \left[ \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} - \mathbf{K}\left(\mathbf{U}_{\text{analytical}} - \mathbf{U}\right) \right] \mathrm{d}t \, \mathrm{d}x \end{aligned} \tag{4.25}$$

Then, using the calculus of variations, the sensitivity derivatives are calculated. Obviously, $\frac{\delta \mathbf{J}_{\text{augmented}}}{\delta K} = \frac{\delta \mathbf{J}}{\delta K}$, thus:

$$\begin{aligned} \frac{\delta \mathbf{J}_{\text{augmented}}}{\delta \mathbf{K}} = & \int_x \int_{t_1}^{t_2} \left(\mathbf{U} - \mathbf{U}_{\text{analytical}}\right) \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \mathrm{d}t \, \mathrm{d}x \\ & + \int_x \int_{t_1}^{t_2} \frac{\delta}{\delta \mathbf{K}} \left( \mathbf{\Psi} \left[ \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} - \mathbf{K}\left(\mathbf{U}_{\text{analytical}} - \mathbf{U}\right) \right] \right) \mathrm{d}t \, \mathrm{d}x \end{aligned} \tag{4.26}$$

The terms of the first integral are well-defined, but some manipulations are required for the terms of the second integral. From left to right:

- For the first term, integration by parts is performed:

$$\begin{aligned} \int_x \int_{t_1}^{t_2} \frac{\delta}{\delta \mathbf{K}} \left( \mathbf{\Psi} \left[ \frac{\partial \mathbf{U}}{\partial t} \right] \right) \mathrm{d}t \, \mathrm{d}x &= \int_x \int_{t_1}^{t_2} \mathbf{\Psi} \frac{\partial}{\partial t} \left( \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \right) \mathrm{d}t \, \mathrm{d}x \\ &= \int_x \left[ \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \mathbf{\Psi} \right]_{t_1}^{t_2} \mathrm{d}x - \int_x \int_{t_1}^{t_2} \frac{\partial \mathbf{\Psi}}{\partial t} \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \mathrm{d}t \, \mathrm{d}x \end{aligned} \tag{4.27}$$

- For the second term, the chain rule is applied:

$$\int_x \int_{t_1}^{t_2} \frac{\delta}{\delta \mathbf{K}} \left( \mathbf{\Psi} \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} \right) \mathrm{d}t \, \mathrm{d}x = \int_x \int_{t_1}^{t_2} \mathbf{\Psi} \frac{\delta}{\delta \mathbf{K}} \left( \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x} \right) \mathrm{d}t \, \mathrm{d}x = \int_x \int_{t_1}^{t_2} \mathbf{\Psi} \frac{\partial^2 \mathbf{F}(\mathbf{U})}{\partial x \partial \mathbf{U}} \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \mathrm{d}t \, \mathrm{d}x \tag{4.28}$$

- For the third term, the product rule is applied:

$$\int_x \int_{t_1}^{t_2} \mathbf{\Psi} \frac{\delta}{\delta \mathbf{K}} \left[ \mathbf{K}\left(\mathbf{U}_{\text{analytical}} - \mathbf{U}\right) \right] \mathrm{d}t \, \mathrm{d}x = \int_x \int_{t_1}^{t_2} \mathbf{\Psi} \left[ \left(\mathbf{U}_{\text{analytical}} - \mathbf{U}\right) - \mathbf{K} \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \right] \mathrm{d}t \, \mathrm{d}x \tag{4.29}$$

Combining all of the above, eq. (4.26) becomes:

$$\begin{aligned}\frac{\delta \mathbf{J}}{\delta \mathbf{K}} = &- \int_x \int_{t_1}^{t_2} \left[ \frac{\partial \boldsymbol{\Psi}}{\partial t} - \boldsymbol{\Psi} \frac{\partial^2 \mathbf{F}(\mathbf{U})}{\partial x \partial \mathbf{U}} - \mathbf{K}\boldsymbol{\Psi} - \left( \mathbf{U} - \mathbf{U}_{\text{analytical}} \right) \right] \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \, \mathrm{d}t \, \mathrm{d}x \\ &- \int_x \int_{t_1}^{t_2} \boldsymbol{\Psi} \left( \mathbf{U}_{\text{analytical}} - \mathbf{U} \right) \mathrm{d}t \, \mathrm{d}x + \int_x \left[ \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \boldsymbol{\Psi} \right]_{t_1}^{t_2} \mathrm{d}x\end{aligned} \tag{4.30}$$

#### 4.3.1.1 Adjoint Field Equations

The main objective is to avoid the direct calculation of $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}$ due to the high computational cost. Specifically, as $\mathbf{K}$ has 9 elements, the lowest accuracy approximation would require solving the primal problem 9 additional times per iteration of the optimization loop, while the adjoint approach requires the solution of 1 additional problem per iteration. In order to factor out the contribution of $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}$ to eq. (4.30), the adjoint equations are devised:

$$\frac{\partial \boldsymbol{\Psi}}{\partial t} - \boldsymbol{\Psi} \frac{\partial^2 \mathbf{F}(\mathbf{U})}{\partial x \partial \mathbf{U}} - \mathbf{K}\boldsymbol{\Psi} = \mathbf{U} - \mathbf{U}_{\text{analytical}} \tag{4.31}$$

The adjoint field equations need to be solved throughout the assimilation window $[t_1, t_2]$. The initial conditions determined by the term $\int_x \left[ \frac{\delta \mathbf{U}}{\delta \mathbf{K}} \boldsymbol{\Psi} \right]_{t_1}^{t_2} \mathrm{d}x$, as this is the only term that $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}$ remains. At $t = t_1$, the $\mathbf{U}$ is known and therefore doesn't change during the optimization loop, thus $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}\Big|_{t_1} = 0$. To avoid taking into account $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}\Big|_{t_2}$, the initial condition of the adjoint field equations is $\boldsymbol{\Psi}(t = t_2) = \mathbf{0}$. It is worth noting that the adjoint field equations are solved backward in time.

Next, the integration scheme will be constructed. Firstly, the $\frac{\partial^2 \mathbf{F}(\mathbf{U})}{\partial x \partial \mathbf{U}}$ term is discretized using Roe's method:

$$\frac{\partial^2 \mathbf{F}(\mathbf{U})}{\partial x \partial \mathbf{U}} = \frac{\partial}{\partial x} \left( \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} \right) \approx \frac{\partial \tilde{\mathbf{A}}}{\partial x} \tag{4.32}$$

and then using a 1st order accurate scheme in space:

$$\frac{\partial \tilde{\mathbf{A}}}{\partial x}\Big|_i^n = \frac{\tilde{\mathbf{A}}_{i+1}^n - \tilde{\mathbf{A}}_i^n}{\Delta x} \tag{4.33}$$

Finally, using 1st order backward finite differences for the time derivative discretization, the following integration scheme, that is similar to the Godunov-like scheme presented in the previous sections, is devised:

$$\boldsymbol{\Psi}_i^{n-1} = \boldsymbol{\Psi}_i^n \left( 1 - \frac{\Delta t}{\Delta x} \frac{\partial \tilde{\mathbf{A}}}{\partial x}\Big|_i^n - \frac{\Delta t}{\Delta x} \mathbf{K} \right) - \frac{\Delta t}{\Delta x} \left( \mathbf{U}_i^n - \mathbf{U}_{\text{analytical,i}}^n \right) \tag{4.34}$$

or taking into account eq. (4.33):

$$\boldsymbol{\Psi}_i^{n-1} = \boldsymbol{\Psi}_i^n \left[ 1 - \frac{\Delta t}{\Delta x} \left( \tilde{\mathbf{A}}_{i+1}^n - \tilde{\mathbf{A}}_i^n \right) - \frac{\Delta t}{\Delta x} \mathbf{K} \right] - \frac{\Delta t}{\Delta x} \left( \mathbf{U}_i^n - \mathbf{U}_{\text{analytical,i}}^n \right) \tag{4.35}$$

where $i$ is the space index, and $n$ is the time index.

#### 4.3.1.2 Sensitivity Derivatives

After defining the adjoint field equations and using Fubini's theorem again, the final expression of the sensitivity derivatives becomes:

$$\frac{\delta \mathbf{J}}{\delta \mathbf{K}} = -\int_{t_1}^{t_2} \int_x \mathbf{\Psi} \left( \mathbf{U}_{\text{analytical}} - \mathbf{U} \right) \mathrm{d}t \, \mathrm{d}x \tag{4.36}$$

In summary, the new procedure incorporating the optimization loop is formed:

- $t \in [0, t_1]$ :
  - $\mathbf{K}$ is initialized, e.g., $\mathbf{K}_{\text{initial}} = 5\mathbf{I}_3$
  - The 1-D Euler equations using nudging are solved for every time-step
- $t \in [t_1, t_2]$ (assimilation window)
  Until a convergence criterion is reached, e.g., $|\mathbf{K}_{\text{new}} - \mathbf{K}_{\text{old}}| \leq 10^{-3}$ the optimization loop is executed:
  - The 1-D Euler equations using nudging are solved forward in time
  - The 1-D adjoint field equations are solved backward in time
  - The sensitivity derivatives are calculated
  - $\mathbf{K}$ is updated using the steepest descent method
- $t \in [t_2, t_{end}]$ :
  - The 1-D Euler equations using nudging are solved for every time-step

### 4.3.2 Optimization Results

The computational code was extended to implement the procedure described in the previous section. Firstly, the assimilation window was selected to be $(\mathrm{NT}_1, \mathrm{NT}_2) = (500, 1500)$ or equivalently $(t_1, t_2) = (0.05, 0.15)$ s with an end time of $t_{end} = 0.42$s. So the assimilation window is around 1/4 of the total simulation problem, and a sufficient fraction of the problem is solved before the assimilation window opens. Additionally, a step size of $\eta = 10^4$ was selected after experimenting with the steepest descent method. Finally, the convergence criterion was set to $|\mathbf{K}_{\text{new}} - \mathbf{K}_{\text{old}}| \leq 10^{-3}$, and the maximum number of optimization iterations was set to 15.

In fig. 4.17 the time evolution of the RMSE components, before and after optimization for different $\mathbf{K}$ initializations is shown. As expected, the optimal gain matrix $\mathbf{K}_{\text{optimal}}$ lead to lower (around an order of magnitude) RMSE minimum values and slightly lower maximum values. It is also worth noting that using this procedure, the off-diagonal elements of the gain matrix are filled, improving the modeling of the implicit correlations between the conservative variables in the nudging source term. The change in the gain matrix is presented below:

$$\mathbf{K}_{\text{initial}} = \begin{bmatrix} 5.0 & 0.0 & 0.0 \\ 0.0 & 5.0 & 0.0 \\ 0.0 & 0.0 & 5.0 \end{bmatrix} \rightarrow \mathbf{K}_{\text{optimal}} = \begin{bmatrix} 10.03 & 2.84 & 4.81 \\ 2.63 & 13.78 & 5.80 \\ 5.79 & 5.98 & 29.06 \end{bmatrix}$$

it is evident that the diagonal terms are increased, leading to a bigger amplitude in the RMSE oscillations, as seen in fig. 4.13. Also, in comparison with fig. 4.13 the off-diagonal terms lead to a

slightly different oscillatory behavior. Finally, the behavior of RMSE when using the optimal gain matrix is similar after the optimization convergence, as expected for a gradient-based optimization method.

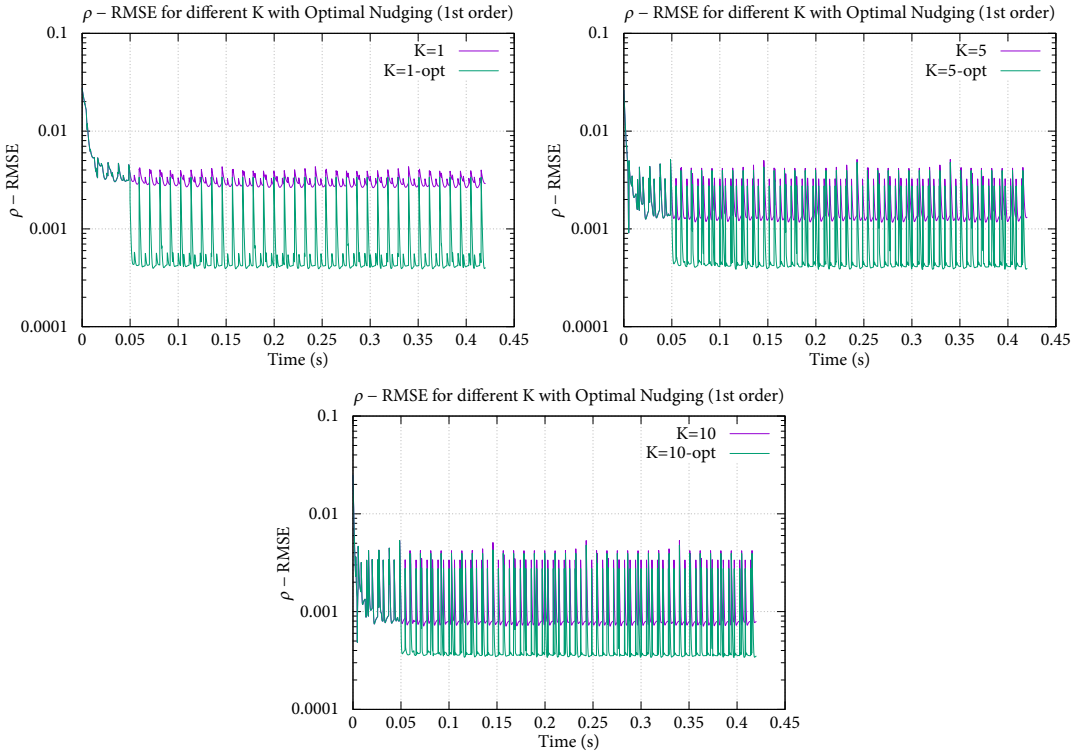



**Figure 4.17:** Time evolution of RMSE of $\rho$ for different initial gain matrices $\mathbf{K}$ and the respective optimized gain matrices.

In fig. 4.18 the time evolution of RMSE throughout the assimilation window during the optimization loop is presented. The improvement to the RMSE decreased as the iteration numbers increased. Specifically, from the 1st to 5th iteration, the minimum has a 50% decrease; from the 5th to 10th iteration, it has a 20% decrease; and from the 10th to 15th iteration, it has a 10% decrease. This observation is also verified by the optimization convergence as presented in fig. 4.19. The convergence rate is slower after the first 5 iterations, a behavior that can be expected because the cost function is really small and possibly reaches a local minimum. Furthermore, the behavior of the steepest descent method is to "zig-zag" with small steps near an optimum [59], especially for constant step size.

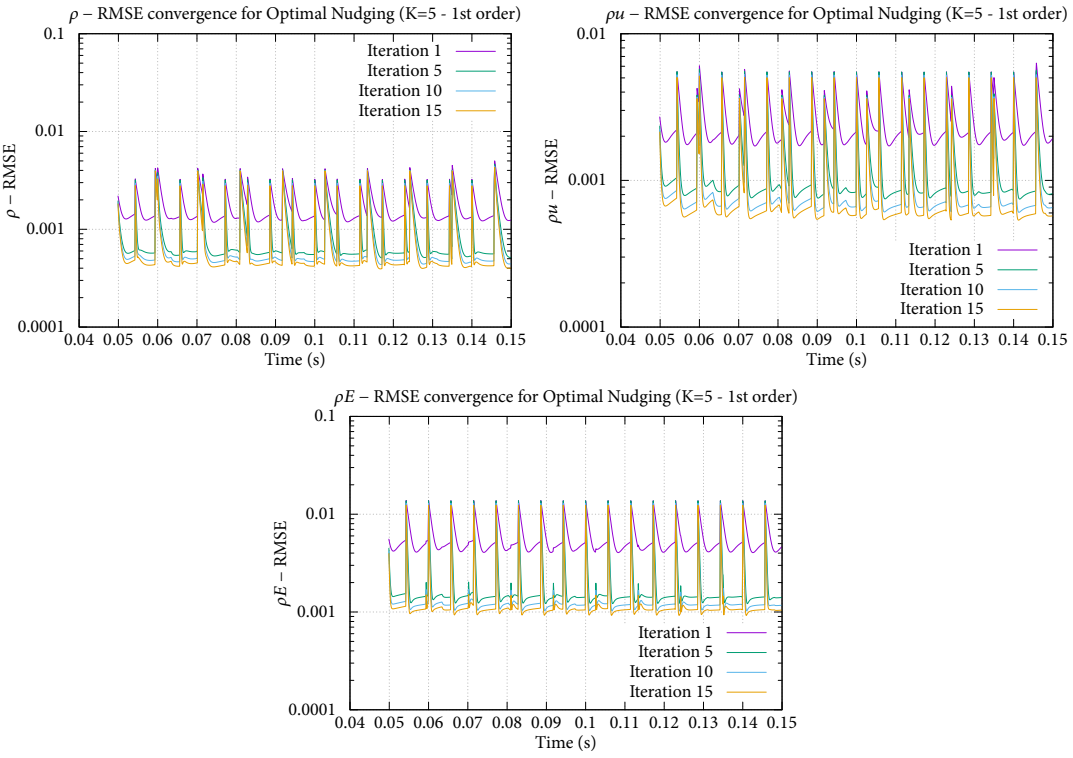



**Figure 4.18:** Time evolution of RMSE during the optimization loop for $\mathbf{K}_{\text{initial}} = 5\mathbf{I}_3$.

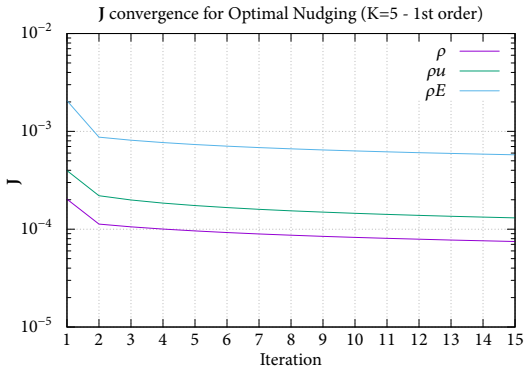



**Figure 4.19:** Convergence of $\mathbf{J}$ components during the optimization loops for $\mathbf{K}_{\text{initial}} = 5\mathbf{I}_3$.

### 4.3.3 Notes on Code Performance

The parallelization of the Fortran 95 code was achieved using OpenMP. OpenMP allows for parallelization in a shared memory computer, e.g., laptop or desktop computer, using compiler directives, which are high-level user-provided instructions informing the compiler that a code region, for example, a loop, can be parallelized and provide user-specified information about data sharing. Thus, the code parallelization is performed more abstractly, with the user identifying the parallelizable regions, the data-sharing options, the necessary synchronization regions, etc., leaving the heavy work to the compiler and the OpenMP library for the required system calls.

Next, the parallel performance of the code was investigated using two different CPUs used in standard laptops, with the same Fortran compiler and OpenMP version, on Linux distributions. The processors a presented briefly in table 4.1. An optimal nudging problem was solved with

the assimilation window defined in the section above and with a grid resolution of NX = 1000. Multiple runs were made for a varying number of processors to ensure that the standard deviation in the timing values is around 1% of the mean values presented.

**Table 4.1:** Brief description of the used CPUs.

| Name | Intel i7-7500U | AMD Ryzen 7 4800H |
|---|---|---|
| # of Processors | 2 | 8 |
| Base Frequency (GHz) | 2.90 | 2.90 |
| System RAM (GB) | 8 | 8 |
| Release Date | Q3'16 | Q2'20 |

In order to calculate the parallel code performance and scalability, the speedup was calculated. The speedup $S(n)$ is defined as the ratio

$$S(n) = \frac{T(1)}{T(n)} \tag{4.37}$$

where $T(1)$ and $T(n)$ are the time the code takes to be executed in one and $n$ processors, respectively. The speedup as a function of the number of processors is shown in fig. 4.20, indicating an asymptotic behavior as the processor's number increases. This behavior was expected, as various parameters influence the way a parallel code can be sped up, and specifically for a problem of this size, the main parameter is the fraction of the code that can be parallelized. Specifically, Amdahl's Law [60] states

$$S(n) = \frac{T(1)}{T(n)} = \frac{1}{\frac{F_{\text{parallel}}}{n} + \left(1 - F_{\text{parallel}}\right)} \tag{4.38}$$

where $F_{\text{parallel}}$ is the fraction of the code that can be executed in parallel. Solving eq. (4.38) for $F_{\text{parallel}}$ yields

$$F_{\text{parallel}} = \frac{n}{n-1}\left(1 - \frac{1}{S(n)}\right) \tag{4.39}$$

Assuming no other factors influenced the parallel performance due to the problem size, the fraction of the code that was parallelized can be approximated by

$$\bar{F}_{\text{parallel}} = \frac{\sum_{i=2}^{N} \frac{n_i}{n_i - 1}\left(1 - \frac{1}{S(n_i)}\right)}{N-1} \tag{4.40}$$

where $N$ is the total number of processors in the CPU. Thus, the mean value and standard deviation of the fraction of the code that can be parallelized were calculated using the results from the AMD CPU, yielding

$$\bar{F}_{\text{parallel}} = 0.463 \pm 0.035 \tag{4.41}$$

Then, the above result was verified for the Intel CPU, with $S_{\text{measured}} \approx 1.307$ and the speedup was calculated using eq. (4.38), $S_{\text{Amdahl}} \approx 1.301$, a really accurate approximation.

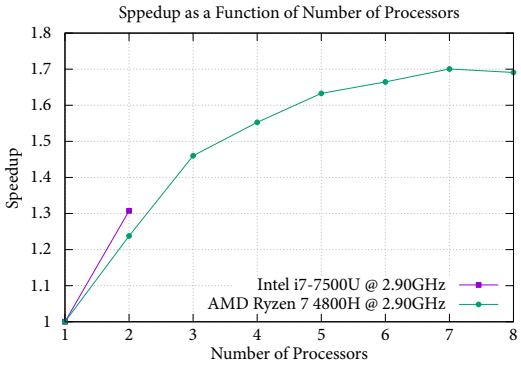



**Figure 4.20:** Speedup as a function of the number of processors.

# Chapter 5

# Application 2: Pressure Calculation from Stereo-PIV Data

In this chapter, Data Assimilation methods are applied to the calculation of pressure fields using velocity fields measurements for the steady-state 2-D turbulent flow around a wall-mounted cube inside the atmospheric boundary layer using the OpenFOAM CFD toolbox. The velocity and Reynolds stress fields are available from previous experimental work [41, 42] at the Laboratory for Innovative Environmental Technologies at the School of Mechanical Engineering at the National Technical University of Athens. The data were obtained using the PIV method, precisely, Stereo-PIV [61].

For the pressure calculation, two different methods were investigated, one using a Poisson equation to determine the pressure field and one using the SIMPLE algorithm using the PIV data for initial and boundary conditions [44]. Firstly, the PIV data were converted into data formats compatible with OpenFOAM. Then, both pressure calculation methods were implemented using OpenFOAM and compared. Afterward, a grid independence study was carried out in order to investigate the sensitivity of the two methods in the grid resolution. Finally, Data Assimilation methods based on nudging and Kalman Filtering were implemented in the developed OpenFOAM-based solver, and their effect on the final pressure and velocity fields was investigated.

## 5.1 Problem Description

The atmospheric boundary layer flow past a wall-mounted cube with vertical openings was experimentally investigated in the wind tunnel of the School of Mechanical Engineering of the National Technical University of Athens in [41, 42]. Two different simulated upstream boundary layer conditions, high-shear (*HS*) and low-shear (*LS*), and two different cube options (with and without openings - open and closed cube) were investigated using Stereo-PIV measurements. The measurement planes are along the cube centerline and located upwind, above, and downwind of the cube as shown in fig. 5.1. The flow is turbulent, as the Reynolds number at cube height is $\mathrm{Re} = 2.4{\cdot}10^4$, as the air velocity is $\rho = 1.21\mathrm{kg/m}^3$ and the air kinematic viscosity is $\nu = 1.479338^{-5}\mathrm{m}^2/\mathrm{s}$. Furthermore, the Reynolds stress tensor field is calculated by the time resolution of the PIV measurements.

Then, for the calculation of the pressure fields, OpenFOAM was used. OpenFOAM is the leading open-source CFD toolbox, written in C++ in an object-oriented way. It provides pre-made solvers and utilities (e.g., for mesh generation and manipulation, data interpolation, etc.) but it is constructed in a way that allows the users to extend it and use its components in their applications. In this Thesis, the OpenFOAM incompressible flow solvers using the SIMPLE

algorithm, called simpleFOAM was extended. Additionally, OpenFOAM allows for the solutions to run in parallel, using MPI (*Message Passing Interface*). Here, all simulations were performed using two processors.

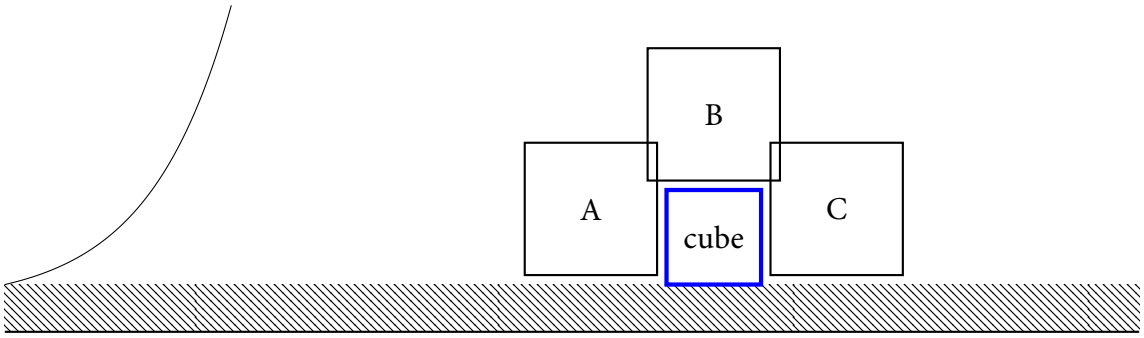



**Figure 5.1:** Planes where PIV measurements are available [42].

## 5.2 Methods Description

### 5.2.1 SIMPLE Implementation

The implementation of the SIMPLE algorithm in OpenFOAM differs from the original described in section 2.2.2.5 [28, 30]. Firstly, the relaxation is performed in an implicit way. At the $k+1-$th iteration eq. (2.51) is transformed to

$$\frac{1}{\beta_u} a_P \mathbf{u}_P = \mathbf{H}(\mathbf{u}) + \frac{1-\beta_u}{\beta_u} a_P \mathbf{u}_P^k - \nabla p_P^k \tag{5.1}$$

where $p_P^k$ and $\mathbf{u}_P^k$ are the pressure and velocity fields at the $k-$th iteration respectively. Then, the $a_P$ coefficient is normalized using the under relaxation factor $a_P^* = \frac{a_P}{\beta_U}$. Additionally, the $\mathbf{H}$ matrix is augmented by the contribution of the terms of the previous iterations. Dividing eq. (5.1) by $a_P^*$ yields

$$\mathbf{u}_P = \frac{\mathbf{H}(\mathbf{u})}{a_P^*} - \frac{\nabla p_P^k}{a_P^*} \tag{5.2}$$

Then, eq. (5.2) is solved, leading to a velocity field $\mathbf{u}^*$ that satisfies the momentum conservation equation but not necessarily the mass conservation equation. Thus, a way to satisfy the continuity equation is needed. In the OpenFOAM implementation, a new pressure field is calculated to satisfy the mass conservation equation instead of calculating a pressure correction field. Firstly, the velocity field is interpolated into the cell faces. For example, the velocity vector at point $P$ becomes

$$\mathbf{u}_P^*|_f = \left(\frac{\mathbf{H}(\mathbf{u}^*)}{a_P^*}\right)_f - \left(\frac{\nabla p_P^k}{a_P^*}\right)_f \tag{5.3}$$

Additionally, the mass conservation equation in discretized form is given by

$$\nabla \cdot \mathbf{u}_P^* = \sum_f \mathbf{S}_f \cdot \mathbf{u}_P^*|_f = 0 \tag{5.4}$$

where $\mathbf{S}_f$ is the surface vector at the $f$ face, given by $\mathbf{S}_f = S_f \mathbf{n}_f$, where $S_f$ is the face area, and $\mathbf{n}_f$ is the normal vector. Combining eqs. (5.3) and (5.4), a Poisson equation is devised for the

calculation of the new pressure field $p^*$

$$\nabla \cdot \left( \frac{1}{a_P^*} \nabla p_P^* \right)_f = \nabla \cdot \left( \frac{1}{a_P^*} \mathbf{H}(\mathbf{u}^*) \right) \tag{5.5}$$

Then, the pressure field is relaxed explicitly

$$p^{k+1} = p^k + \beta_p \left( p^* - p^k \right)$$

Then the volumetric fluxes across the cell faces are corrected:

$$\mathbf{Flux}_f = \mathbf{S}_f \cdot \mathbf{u}_f^* = \mathbf{S}_f \cdot \left[ \left( \frac{\mathbf{H}(\mathbf{u}^*)}{a_P^*} \right)_f - \left( \frac{\nabla p_P^{k+1}}{a_P^*} \right)_f \right] \tag{5.6}$$

Finally, the final velocity field is given by

$$\mathbf{u}_P^{k+1} = \frac{\mathbf{H}(\mathbf{u}^*)}{a_P^*} - \frac{\nabla p_P^{k+1}}{a_P^*} \tag{5.7}$$

In general, the pressure-velocity coupling in collocated grids, as in OpenFOAM, presents an oscillatory behavior in the pressure values between neighboring cells, called the checkerboard effect. This behavior stems from the spatial discretization of $\nabla p$ and the linear interpolation required for the flux reconstruction on the cell faces. This behavior can be avoided using staggered grids, which is more challenging to implement in a numerical code. Another way to prevent this behavior is using the Rhie-Chow interpolation [62]. In OpenFOAM, the Rhie-Chow interpolation is used in an implicit way. As eq. (5.5) is discretized using the Gauss theorem, the Laplacian term of $p$ uses the value of the gradient of $p$ on the cell face, and the gradient is calculated using neighboring cells and not neighboring faces. In comparison, the gradient term of $p$ in eq. (5.7) is calculated from the cell face values of $p$. The SIMPLE implementation in OpenFOAM can be summarized as:

- Initial guess of $p$ and $\mathbf{u}$
- Until convergence is reached:
  1. Momentum predictor: Solve discretized momentum equation, eq. (5.2) $\rightarrow \mathbf{u}^*$
  2. Pressure predictor: Solve pressure equation, eq. (5.5) $\rightarrow p^*$
  3. Correct volumetric fluxes, eq. (5.6)
  4. Relax pressure field, $\rightarrow p^{k+1}$
  5. Momentum corrector: Solve velocity correction equation eq. (5.7) $\rightarrow \mathbf{u}^{k+1}$
- Finalize procedure

### 5.2.2 PIV and CFD Grids

Next, in order to perform calculations using the PIV data, they had to be converted into a format compatible with OpenFOAM. Therefore, a Python script was created to convert the PIV grid into the PLOT3D format [63] and subsequently converted into the OpenFOAM data format using the plot3DToFOAM utility. As the PIV values are known in the cell nodes, instead of the cell centers, a small OpenFOAM application was developed to load the PIV data, perform bilinear

interpolation of the nodal values to the cell centers and then initialize the fields in the native OpenFOAM data format. The bilinear interpolation was used due to the fact that the PIV grids are almost uniform and orthogonal grids. So the velocity values on the cell center $P$ are given by

$$\mathbf{u}_P = \frac{1}{4} \sum_{\text{nodes}} \mathbf{u}_{n,P} \tag{5.8}$$

where $\mathbf{u}_{n,P}$ are the values in the $P$ cell nodes. Additionally, for the cell centers on the boundary faces, a simple linear interpolation is performed using the two closest boundary nodes. For example, for the west boundary (using the notation from fig. 2.2)

$$\mathbf{u}_{P,W} = \frac{\mathbf{u}_N + \mathbf{u}_S}{2} \tag{5.9}$$

This approach ensures that only interpolations are performed, avoiding extrapolations of the experimental data. Also, it ensures that the PIV and CFD grids have precisely the same dimensions and boundaries. In fig. 5.2 the PIV and CFD grids are shown. The cell nodes where the PIV data are known are colored in red, while the cell centers where the values are stored in OpenFOAM are colored in blue. Additionally, in the center of the left subfigure of fig. 5.2 four cells are colored in red, indicating the nodes that are used in each interpolation to the cell center, and four cell centers are connected with blue lines to indicate the cells that a single node influences when the interpolation is performed.

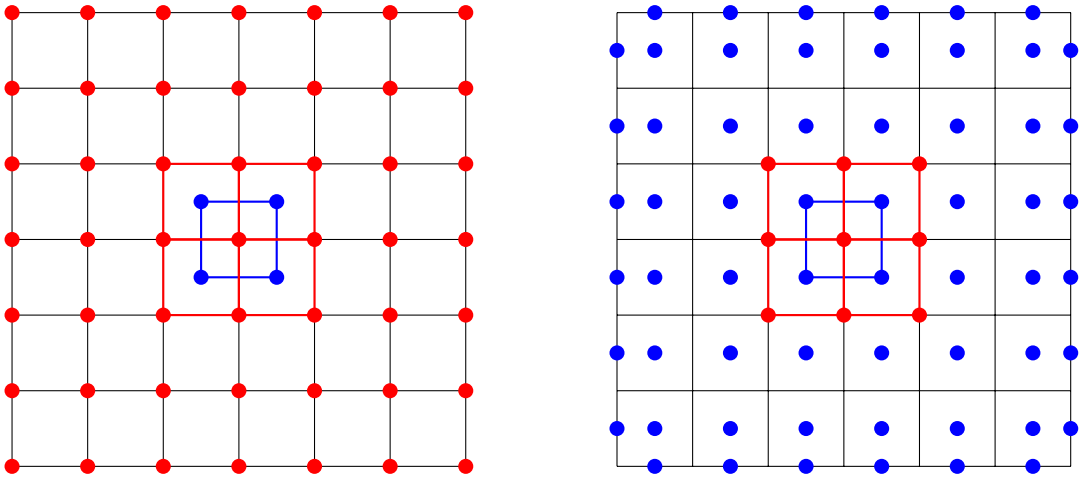


**Figure 5.2:** PIV (left) & CFD (right) grids. The cell nodes where PIV data are available are colored in red, while the cell centers used in OpenFOAM are colored in blue.

### 5.2.3 Pressure Calculation from PIV Data

In the literature, four prominent families of methods of calculating pressure fields from PIV velocity fields [64] exist and can be summarized as:

- Bernoulli equation
- Integration methods
- Poisson equation
- Iterative methods [44, 65]

Two methods were implemented in this Thesis: the solution of a Poisson equation for the pressure fields and an iterative method that uses the SIMPLE algorithm to correct both pressure and velocity fields [44]. The first method was selected as it is straightforward to implement, and the only assumption it needs is that the mass conservation equation stands. The process of obtaining the Poisson equation is similar to the method used to obtain the SIMPLE pressure equation. Specifically, the divergence of the RANS momentum equation, assuming that the mass conservation equation stands, yields:

$$\nabla^2\left(\frac{\bar{p}}{\rho}\right) = -\nabla\cdot\left[\nabla\cdot(\mathbf{u}_{\mathrm{PIV}}\mathbf{u}_{\mathrm{PIV}}) + \nabla\cdot\overline{\mathbf{u}'_{\mathrm{PIV}}\mathbf{u}'_{\mathrm{PIV}}}\right] \tag{5.10}$$

where $\mathbf{u}_{\mathrm{PIV}}$ is the mean velocity field, but the bar is removed for simplicity as in all equations in this Chapter, and $\overline{\mathbf{u}'_{\mathrm{PIV}}\mathbf{u}'_{\mathrm{PIV}}}$ is the Reynolds stress tensor available from the PIV data.

On the other hand, the formulation of the SIMPLE algorithm was performed as in described in the sections above, using the PIV velocity field as the initial condition for the velocity field, $\mathbf{u}_{\mathrm{initial}} = \mathbf{u}_{\mathrm{PIV}}$. The substantive difference between the two methods emanates from the assumption made in the first method, that the mass conservation equation stands. This assumption, as shown later in this chapter, doesn't necessarily stand. However, both methods require the solution of a Poisson equation, which requires special consideration for the pressure boundary conditions.

In this Thesis, zero Dirichlet boundary conditions were selected for the pressure in the boundaries not adjacent to the cube, and zero gradient (Neumann) boundary conditions for the boundaries adjacent to the cube. Specifically, zero gradient boundary conditions were enforced in the east boundary of plane A, the south boundary of plane B, and the west boundary of plane C. An additional consideration was made for the south boundaries of the A and C planes, as they are close to the ground, which is a solid boundary. However, on the one hand, the distance between their southern boundaries and the ground is around 0.1 to 0.15 times the cube height; on the other hand, the distances between the boundaries that neighbor the cube are 2 to 3 times smaller than the respective ground distance. Thus, no special treatment was considered for the southern boundaries of planes A and C.

In order to use the SIMPLE algorithm, the definition of velocity boundary conditions is also needed. According to the method developed at the Laboratory for Innovative Environmental Technologies at the School of Mechanical Engineering at the National Technical University of Athens [44], the use of Dirichlet boundary conditions, using the PIV data was selected. Additionally, the Dirichlet boundary conditions, were expanded into a zone of two cells in the interior part of the mesh, where the velocity field is considered constant and equal to the respective PIV data. It is worth noting that in general, this selection introduced a zone of high continuity residuals, but its impact in the method convergence and stability was evident.

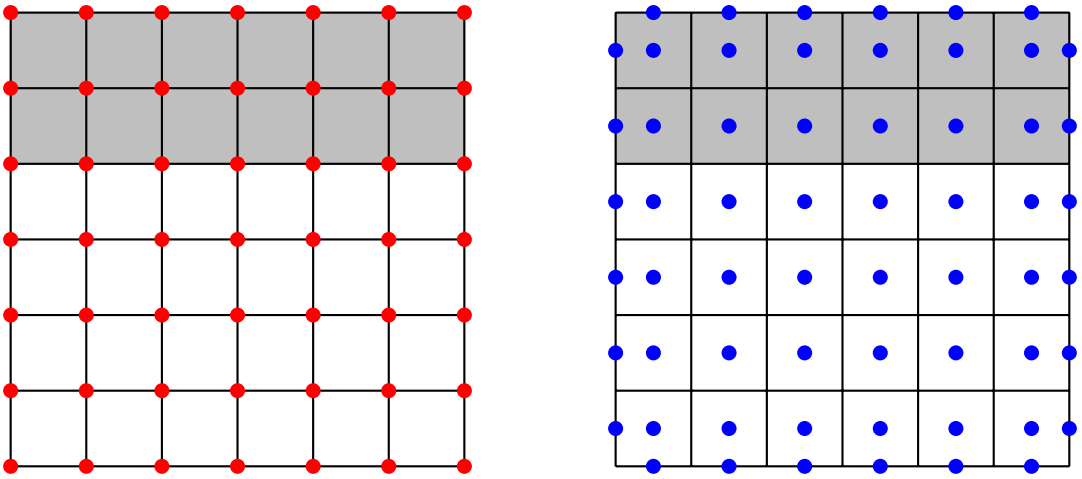


**Figure 5.3:** Fixed cell zone in northern boundary [44].

### 5.2.4 Simulation Setup

In this section, a brief description of the simulation setup is presented. As described before, the Reynolds stress tensor from the PIV data was used "as-is" in the solution of the NS equations, and no turbulence model was needed.

Additionally, the convergence criteria were set as described in section 2.2.2.5. The solution of the momentum and pressure equations, in the matrix form, was performed using iterative methods. Specifically, the linear system equation $\mathbf{Ax} = \mathbf{b}$ is considered as an optimization problem, the $\mathbf{x}$ that minimizes the residual $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$ is calculated iteratively using an optimization method. Here, the preconditioned conjugate gradient (PCG) method coupled with the Diagonal-based Incomplete Cholesky (DIC) preconditioner was used for the solution of the (symmetric matrix) pressure equation. Respectively, the Preconditioned bi-conjugate gradient (PBiCG) method coupled with the Diagonal-based Incomplete LU (DILU) preconditioner was used for the solution of the (asymmetric matrix) momentum equation.

Finally, the discretization schemes described in sections 2.2.2.1 to 2.2.2.4 were used and are summarized in table 5.1.

**Table 5.1:** Summary of used OpenFOAM discretization schemes.

| Term | Name | Order of Accuracy |
|---|---|---|
| $\nabla p$ | Gauss linear | 2nd |
| $\nabla \mathbf{u}$ | Gauss linear | 2nd |
| $\nabla \cdot (\mathbf{uu})$ | Gauss upwind | 1st |
| $\nabla \cdot (\mathbf{u}_{\mathrm{PIV}} \mathbf{u}_{\mathrm{PIV}})$ | Gauss upwind | 1st |
| $\nabla \cdot (\overline{\mathbf{u}_{\mathrm{PIV}'} \mathbf{u}_{\mathrm{PIV}'}})$ | Gauss linear | 2nd |
| $\nabla \cdot (\nu \nabla \mathbf{u})$ | Gauss linear orthogonal | 2nd |

### 5.2.5 Metrics

A small number of pressure measurements along the cube centerline were available from the experiment [41, 42] and will contribute to the verification of the method for the high-shear cases.

Thus the pressure coefficient $C_p$ used as a metric and is given by

$$C_p = \frac{p - p_{\text{ref.}}}{p_{\text{dyn.}}} \tag{5.11}$$

where $p_{\text{ref}}$ is a reference pressure and $p_{\text{dyn}}$ is the dynamic pressure $\frac{1}{2}\rho U_{\text{inf}}^2$ where $U_{\text{inf}}$ is the freestream velocity. Due to the nature of the numerical solution, as the flow is solved in planes, and the fact that the reference pressure used in the experiments was well above the cube height of $h_{\text{cube}} = 0.11\text{m}$, the pressure reference and the dynamic pressure varied for each plane, taking into account the flow behavior in it. Specifically, due to the presence of the stagnation point in plane A, it was selected $p_{\text{ref}} = 0$ and $p_{\text{dyn}} \approx p_{\text{stagnation}} \approx 9\text{Pa}$. Additionally, as the flow in plane B resembles the one over a flat plate, it was selected $p_{\text{ref}} = 0$ and $p_{\text{dyn}} = \frac{1}{2}\rho U_{\text{cube}}^2 \approx 6\text{Pa}$ where $U_{\text{cube}}^2$ is the freestream velocity at the cube height, as calculated six cube heights downstream. Finally, for plane C, the reference pressure was selected to be the pressure in the upper left corner of the plane, $p_{\text{ref}} \approx 2\text{Pa}$, while the dynamic pressure was calculated using the freestream velocity, $p_{\text{dyn}} \approx 12\text{Pa}$

Additionally, for the validation of the quality of the PIV measurements and the CFD results, some metrics were defined that quantify the velocity field divergence or the failure to obey the mass conservation equation and the change of the fields using the SIMPLE algorithm. Thus, the normalized continuity residual CRN is given by

$$\text{CRN} = \frac{\nabla \cdot \mathbf{u}}{\left(\frac{\langle |\mathbf{u}_{\text{PIV}}| \rangle}{2\Delta x}\right)} \tag{5.12}$$

where $\langle |\mathbf{u}_{\text{PIV}}| \rangle$ is the mean magnitude of the PIV velocity field. Additionally, the relative velocity calculation error is defined as

$$\varepsilon_{r,\text{Velocity}} = \frac{|\mathbf{u}_{\text{CFD}} - \mathbf{u}_{\text{PIV}}|}{\langle |\mathbf{u}_{\text{PIV}}| \rangle} \times 100\% \tag{5.13}$$

The normalization is performed using the mean magnitude of the velocity field in order to avoid obtaining extreme values near the stagnation point where the velocity field values are really small. Finally, the absolute pressure calculation error is given by

$$\varepsilon_{a,\text{Pressure}} = p_{\text{SIMPLE}} - p_{\text{Poisson}} \tag{5.14}$$

The metrics defined above are scalar fields; thus, in addition to their contour plots, the histograms of the field values were created. On the horizontal axis, the bins of the field values are presented, and on the vertical axis, the relevant volume fraction values are presented, defined as

$$\text{Volume fraction} = \frac{\sum_{celli} \left[ V_{celli} | \min(class) \leq \varphi_{celli} < \max(class) \right]}{\sum_{celli} \left[ V_{celli} | \min(field) \leq \varphi_{celli} < \max(field) \right]} \tag{5.15}$$

### 5.2.6 Grid Independence - Mesh Refinement

In CFD, the mesh quality is one of the most important factors influencing the final solution's quality. The meshes used in this application are available from the PIV measurements and are structured and uniform (and orthogonal). However, they might be considered to be relatively coarse compared to a typical CFD mesh. For this reason, the investigation of the effect of mesh resolution on the CFD solution is of interest. Thus, the initial PIV meshes were refined using the

refineMesh OpenFOAM utility and the data interpolation from the coarse to the finer meshes was performed using the mapFields OpenFOAM utility.

The refineMesh utility splits each cell of the orthogonal mesh into four equal cells. Thus a finer mesh (×4) and a finest mesh (×16) were produced for each studied case, as shown in fig. 5.4. Then, after the refinement process, the flow was solved in each refined mesh and compared to the results obtained from the nominal meshes.

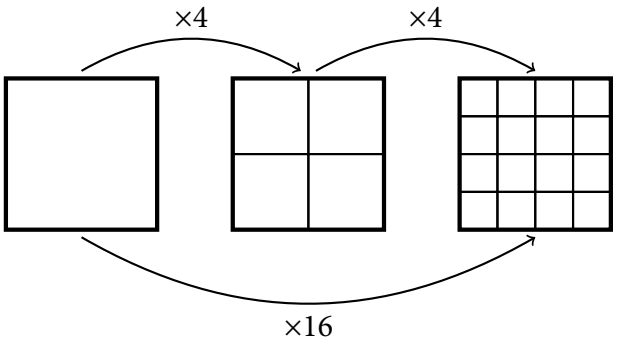



**Figure 5.4:** Mesh refinement strategy for the grid independence study.

## 5.3 Plane A

Firstly, the methods were applied to plane A. This plane is located upwind of the cube, resulting in a stagnation point appearing at $y \approx 0.7h_{\text{cube}}$. As shown in the velocity magnitude contours in figs. 5.6, 5.15, 5.23 and 5.27, the SIMPLE-based method managed to construct velocity fields that align with the experimental data. Additionally, as evident in the CRN contour plots, figs. 5.8, 5.17, 5.23 and 5.27, and the CRN distribution histograms, figs. 5.11, 5.20, 5.24 and 5.28, the SIMPLE-based method managed to reduce the continuity residuals of the fields.

Furthermore, as shown in the pressure contours in figs. 5.7, 5.16, 5.23 and 5.27 and the $C_p$ distribution in figs. 5.5 and 5.14, the pressure fields produced from the SIMPLE-based method, were close to both the experimental data and the previous implementation [44]. The discrepancies between the results stem from the different pressure references used and the use of the zero Dirichlet boundary condition for the pressure in the southern plane boundary, which led to a smoother pressure gradient profile. On the contrary, the method that used the Poisson equation failed to adequately predict the pressure fields, as shown in figs. 5.7, 5.16, 5.23 and 5.27 and figs. 5.5 and 5.14. This failure emanated from the continuity errors present in the PIV measurements, as shown in the CRN contours in figs. 5.8, 5.17, 5.23 and 5.27 and the CRN distribution histograms, figs. 5.11, 5.20, 5.24 and 5.28.

Additionally, it is evident from figs. 5.6, 5.7, 5.15, 5.16, 5.23 and 5.27, that the mesh refinement doesn't result in substantial differences in the final velocity and pressure fields. The most noteworthy difference is observed in the CRN behavior. Due to the mesh refinement, the fixed velocity boundary zones are reduced as a percentage of the total cell number as the refinement quadruples the number of cells, while the number of fixed boundary cells doubles. Thus, the discontinuity observed in figs. 5.8 and 5.17 at the boundary between the fixed velocity zone and the internal mesh was reduced as the mesh was refined. The CRN histograms, shown in figs. 5.11, 5.20, 5.24 and 5.28, also verify this observation, as the volume fraction of the cells that have great variance from the continuity was reduced as the mesh was refined.

Finally, it is worth noting that the distributions of the normalized relative velocity errors, shown in figs. 5.12, 5.21, 5.25 and 5.29 and absolute pressure errors, shown in figs. 5.13, 5.22,

5.26 and 5.30 do not change significantly as the mesh is refined. This behavior is expected, as the relative velocity errors and absolute pressure errors are not significantly influenced by the discontinuity at the boundary zones, as shown in the relevant contour plots figs. 5.9 and 5.18 and figs. 5.10 and 5.19 Thus, in the following sections, the relevant contour plots are omitted for brevity.

**Table 5.2:** Number of cells for the different cases in plane A.

| Mesh | High Shear | | Low Shear | |
|---|---|---|---|---|
| | Open | Closed | Open | Closed |
| Nominal | 9118 | 16125 | 9118 | 9118 |
| ×4 | 36472 | 64500 | 36472 | 36472 |
| ×16 | 145888 | 258000 | 145888 | 145888 |

### 5.3.1 High Shear Conditions

#### 5.3.1.1 Open Cube

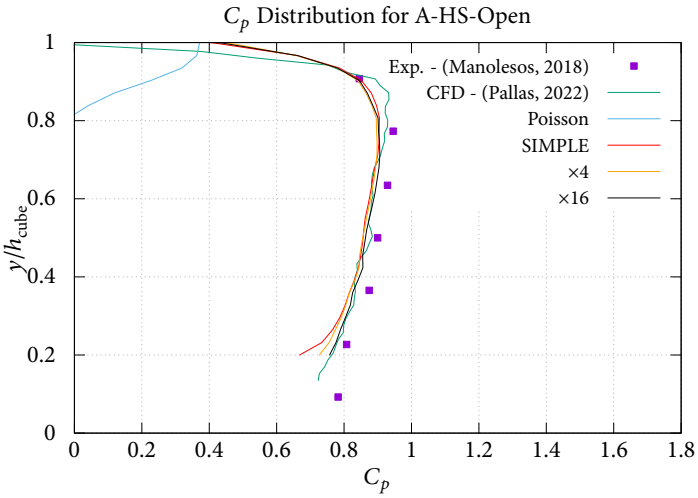
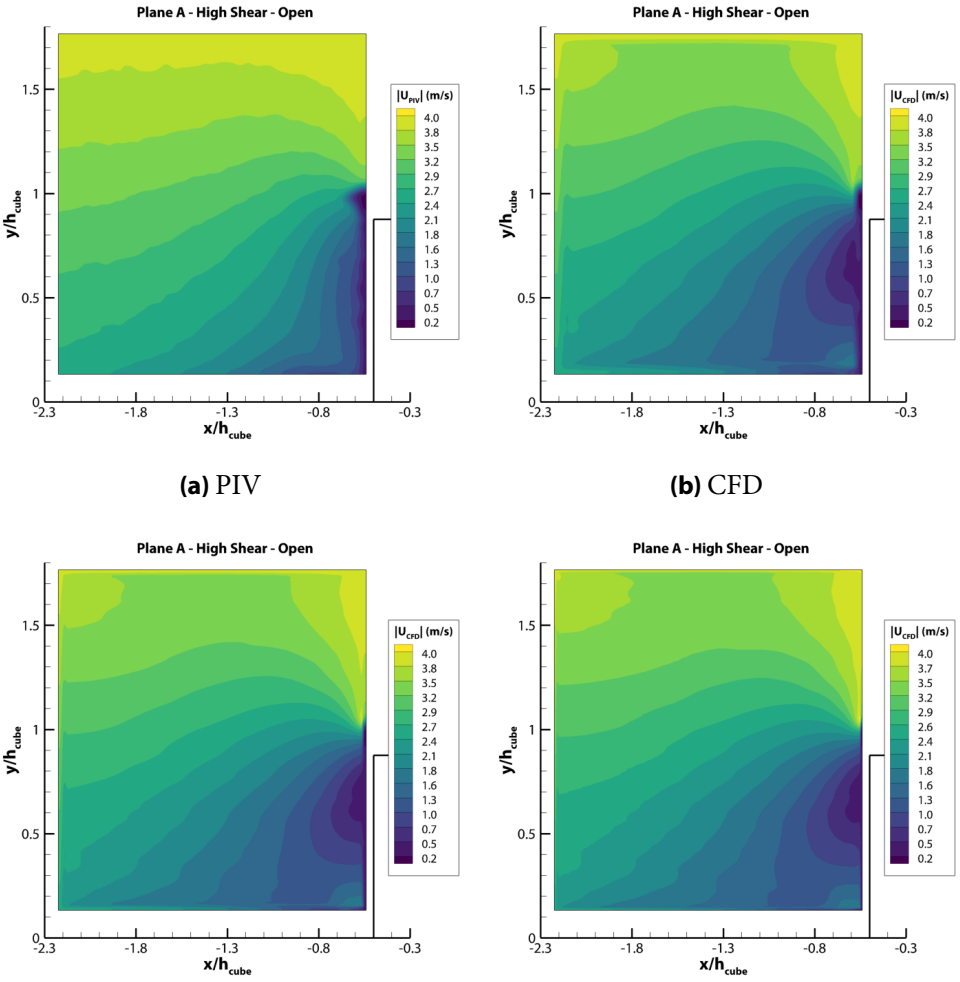


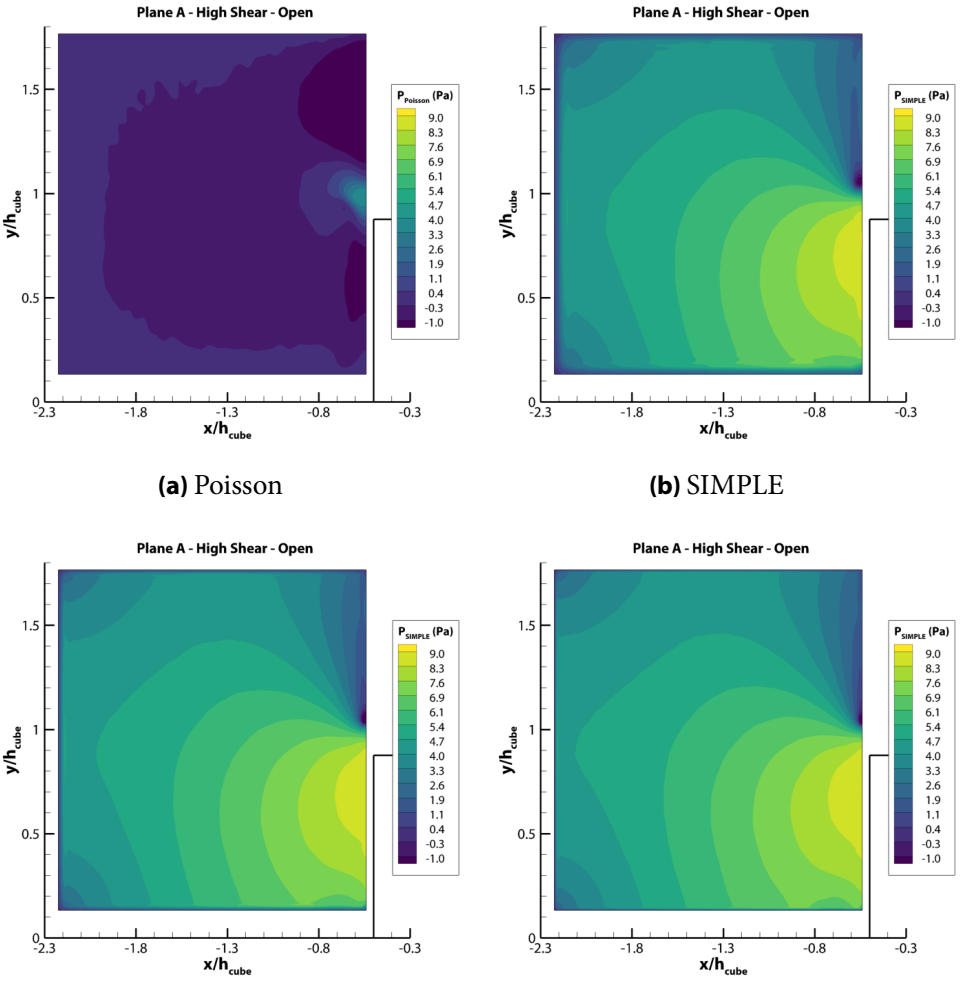

**Figure 5.5:** Pressure coefficient distribution along the cube for A-HS-Open.




**(a)** PIV **(b)** CFD

**(c)** CFD Refined ×4 **(d)** CFD Refined ×16

**Figure 5.6:** Velocity magnitude contours for A-HS-Open.



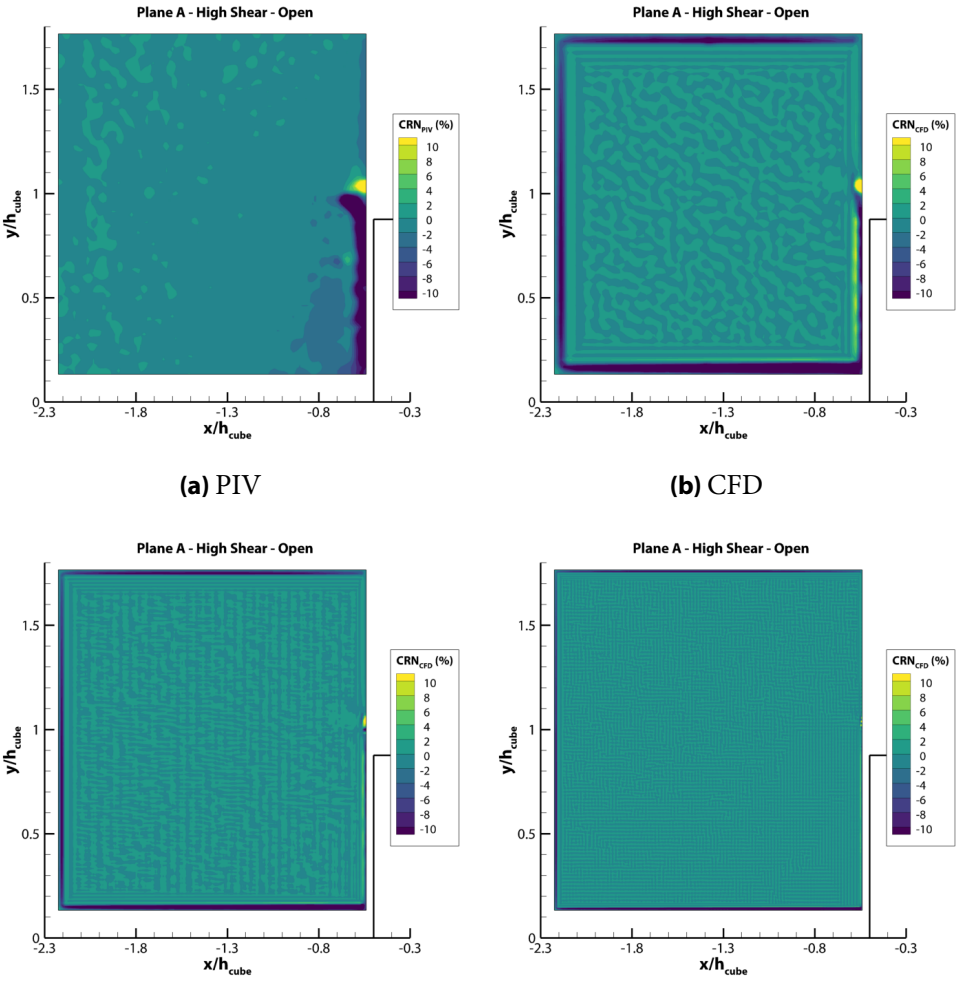
**(a)** Poisson

**(b)** SIMPLE

**(c)** SIMPLE Refined ×4

**(d)** SIMPLE Refined ×16

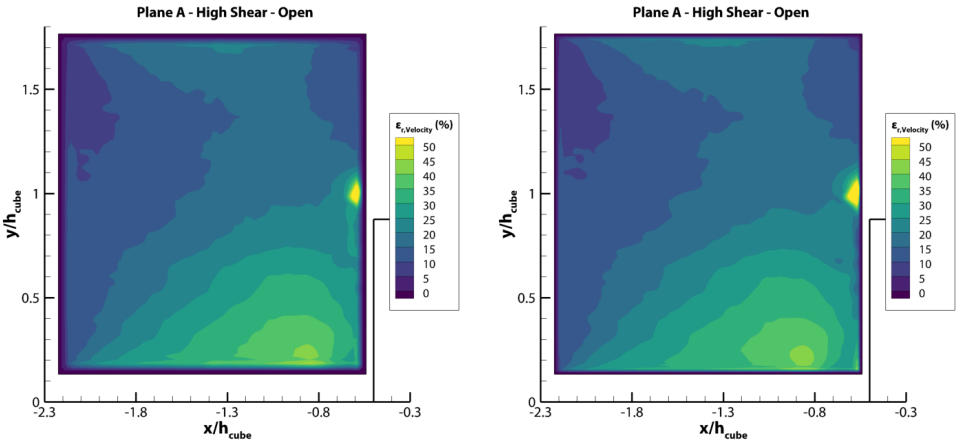
**Figure 5.7:** Pressure contours for A-HS-Open.




**(a)** PIV **(b)** CFD

**(c)** CFD Refined ×4 **(d)** CFD Refined ×16

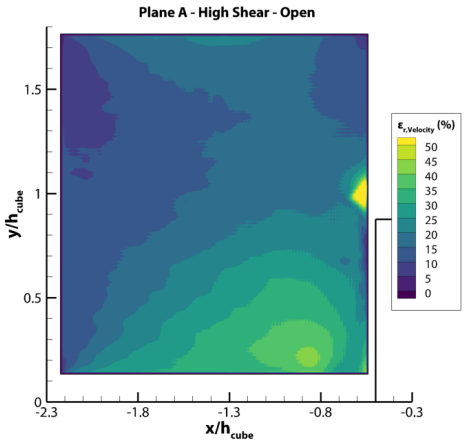
**Figure 5.8:** Normalized continuity residual contours for A-HS-Open.




**(a)** Nominal Grid

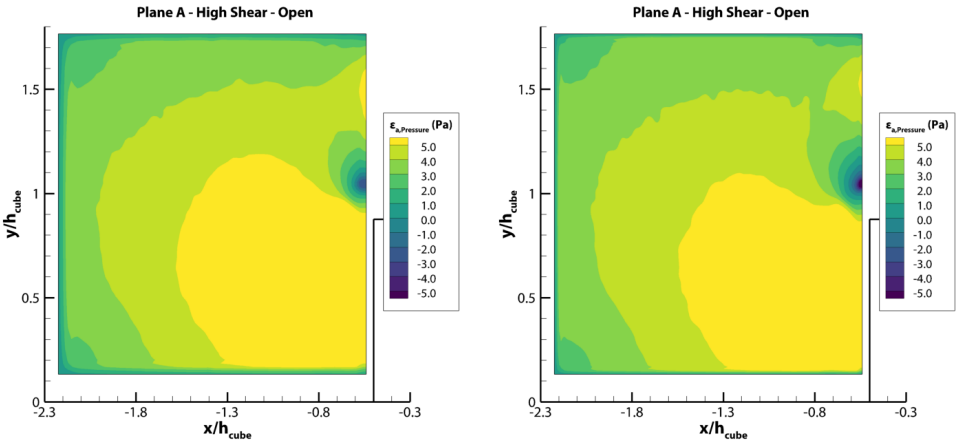



**(b)** ×4 Refined Grid

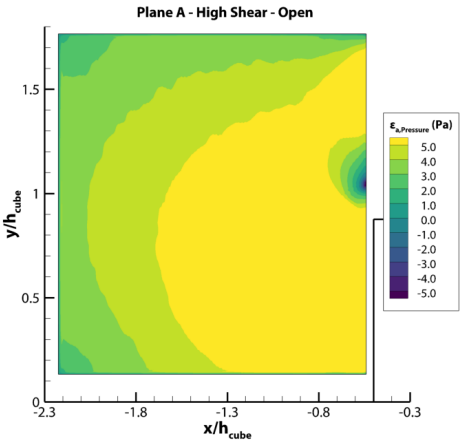



**(c)** ×16 Refined Grid

**Figure 5.9:** Relative velocity error contours for A-HS-Open.

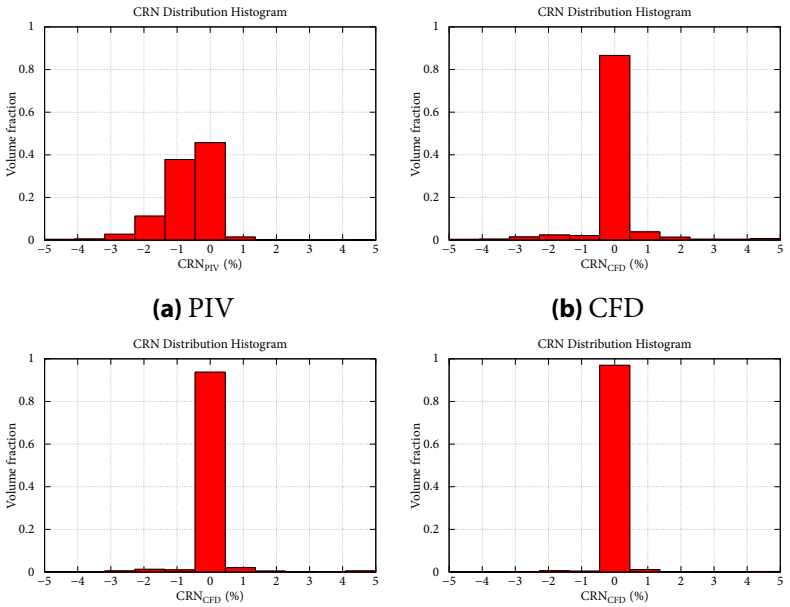



**(a)** Nominal Grid **(b)** ×4 Refined Grid




**(c)** ×16 Refined Grid

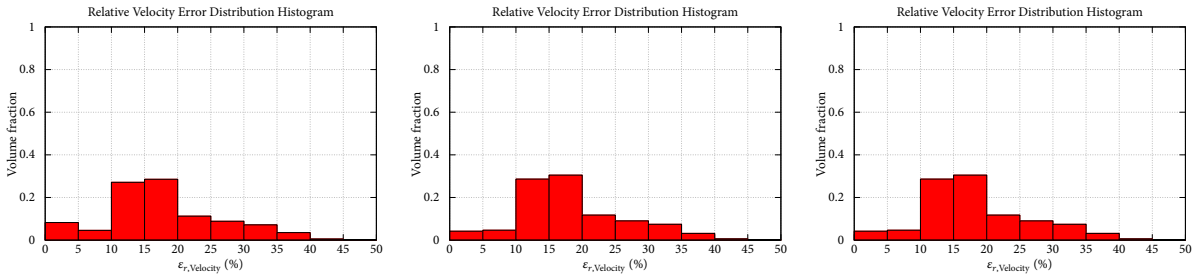
**Figure 5.10:** Absolute pressure error contours for A-HS-Open.




**(a)** PIV **(b)** CFD

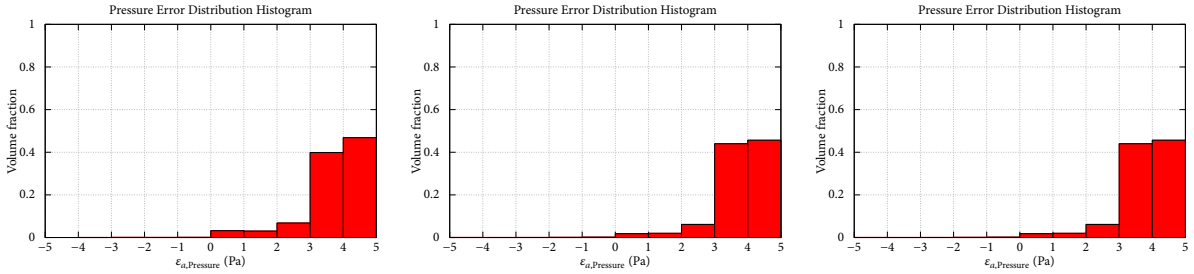
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.11:** Distribution of normalized continuity residual field for A-HS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.12:** Distribution of relative velocity error field for A-HS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.13:** Distribution of absolute pressure error field for A-HS-Open.

#### 5.3.1.2 Closed Cube

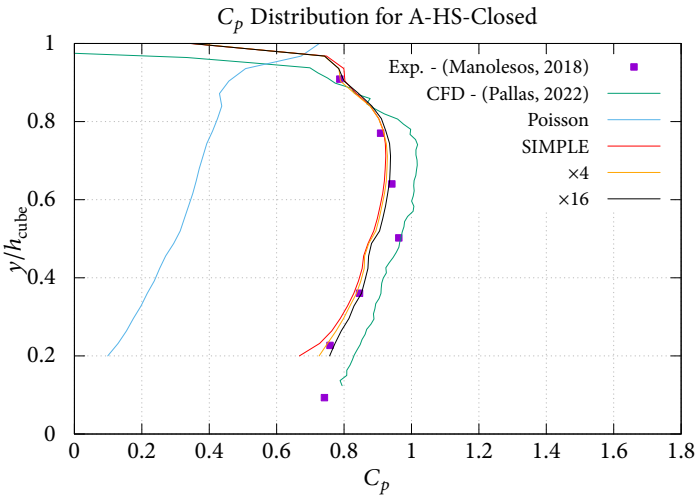
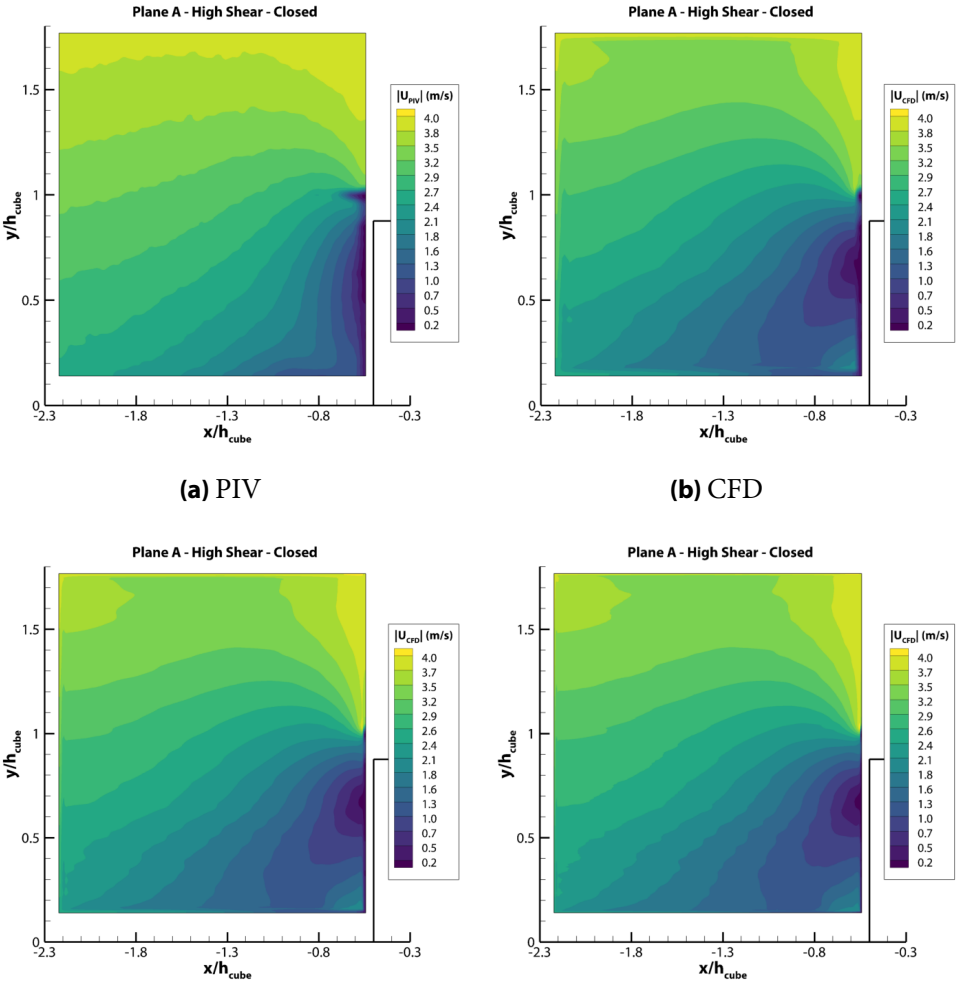


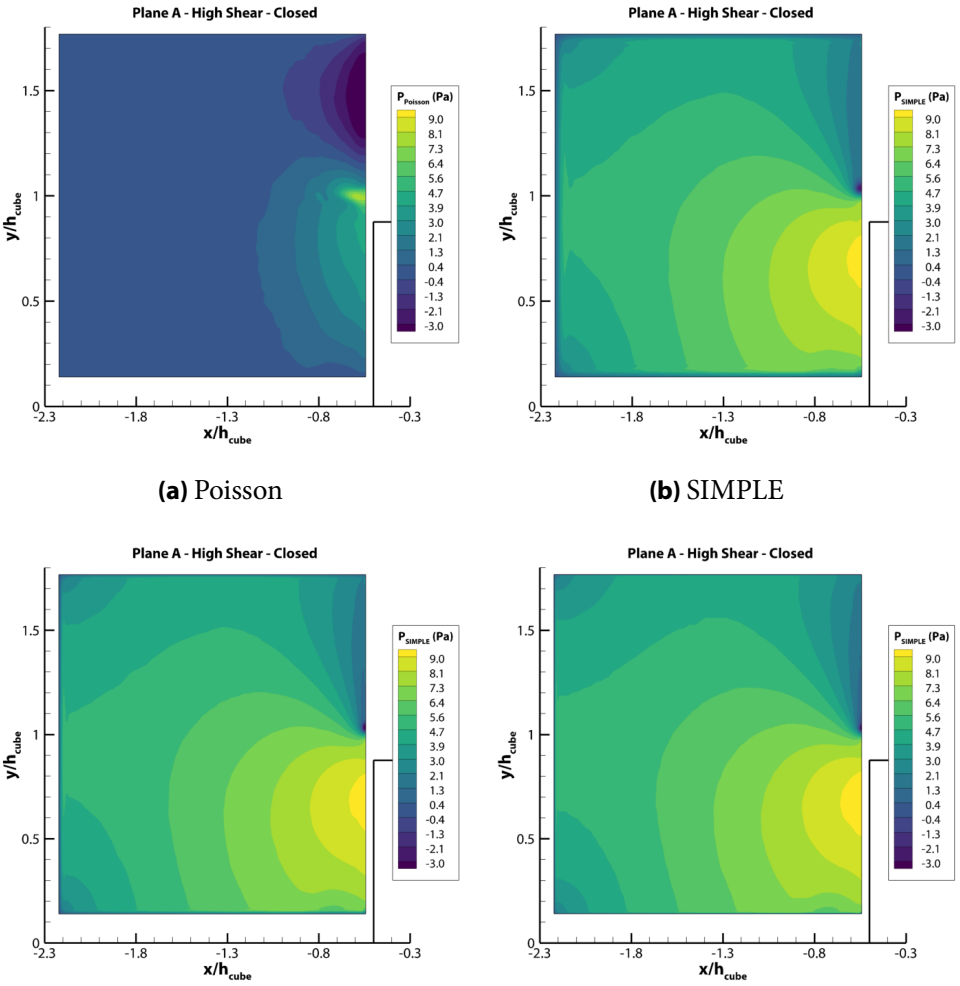

**Figure 5.14:** Pressure coefficient distribution along the cube for A-HS-Closed.




**(a)** PIV **(b)** CFD

**(c)** CFD Refined ×4 **(d)** CFD Refined ×16

**Figure 5.15:** Velocity magnitude contours for A-HS-Closed.



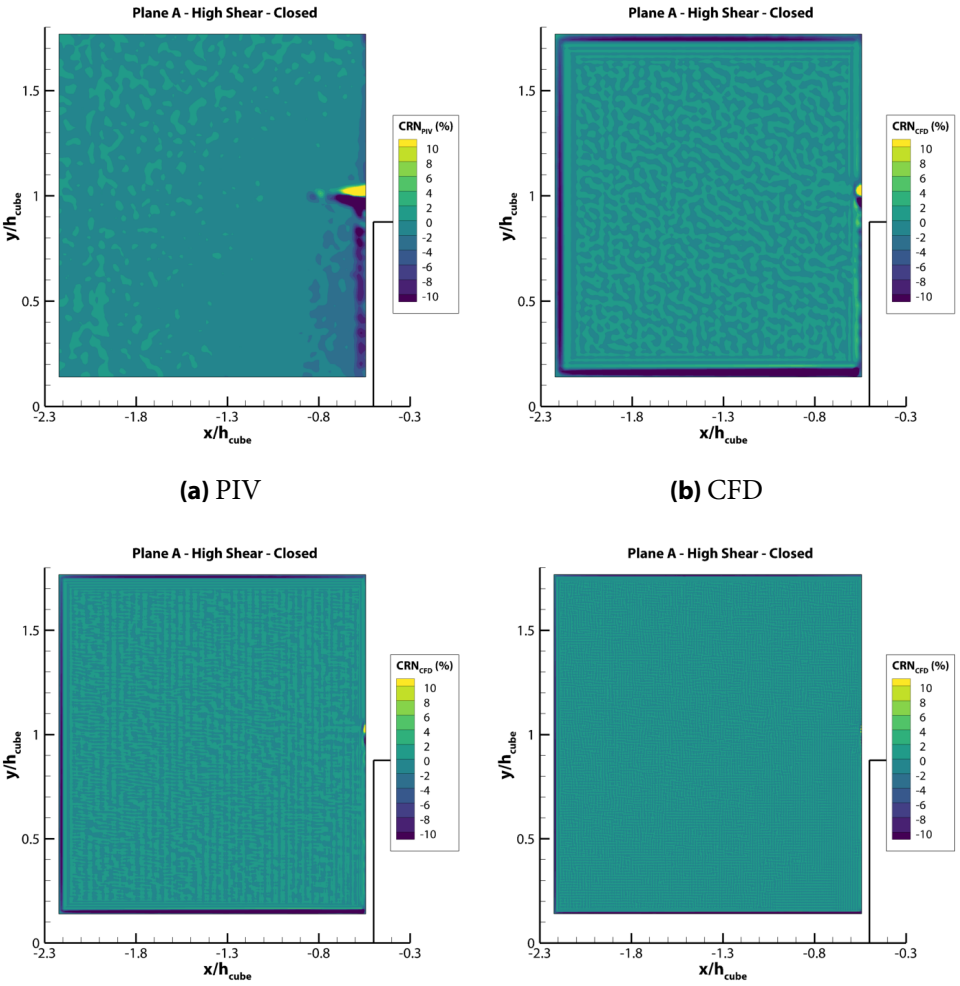

**(a)** Poisson

**(b)** SIMPLE

**(c)** SIMPLE Refined ×4

**(d)** SIMPLE Refined ×16

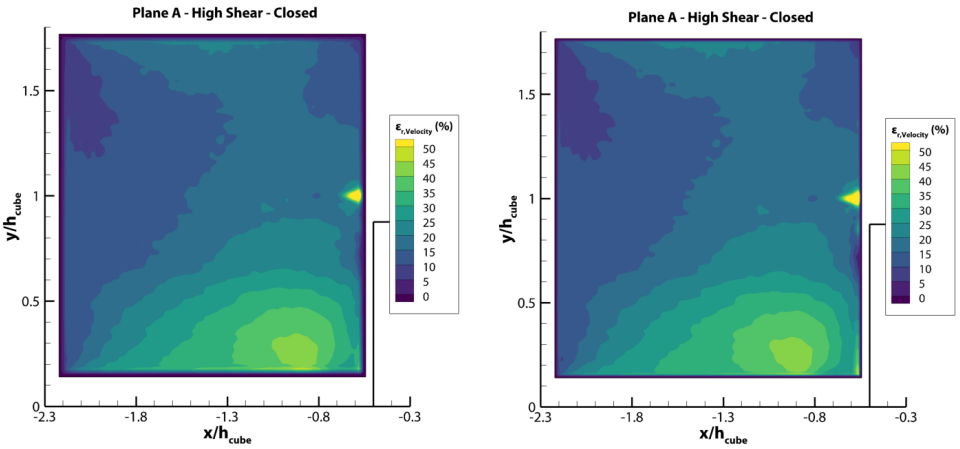
**Figure 5.16:** Pressure contours for A-HS-Closed.




**(a)** PIV **(b)** CFD

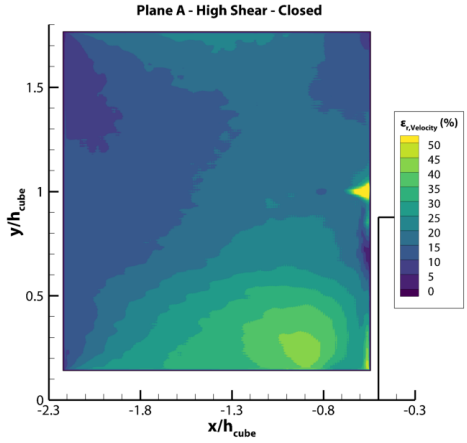
**(c)** CFD Refined ×4 **(d)** CFD Refined ×16

**Figure 5.17:** Normalized continuity residual contours for A-HS-Closed.



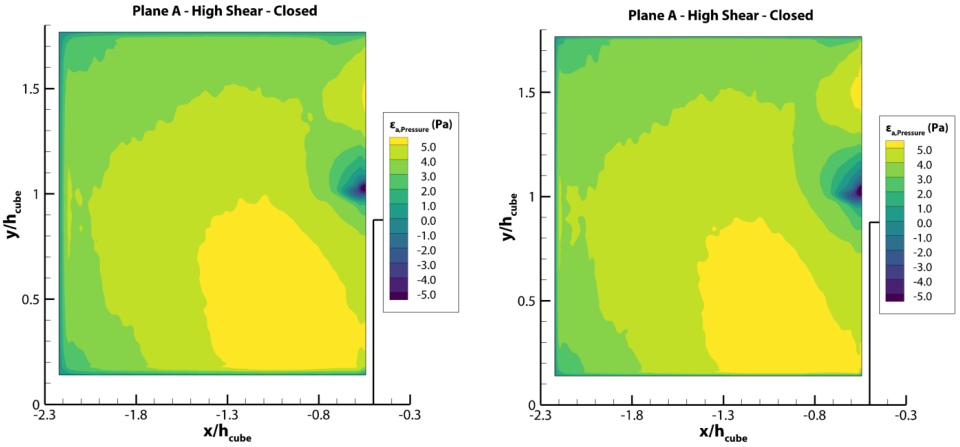

**(a)** Nominal Grid

**(b)** ×4 Refined Grid

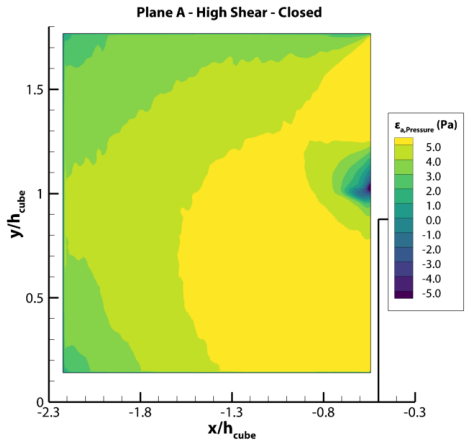



**(c)** ×16 Refined Grid

**Figure 5.18:** Relative velocity error contours for A-HS-Closed.

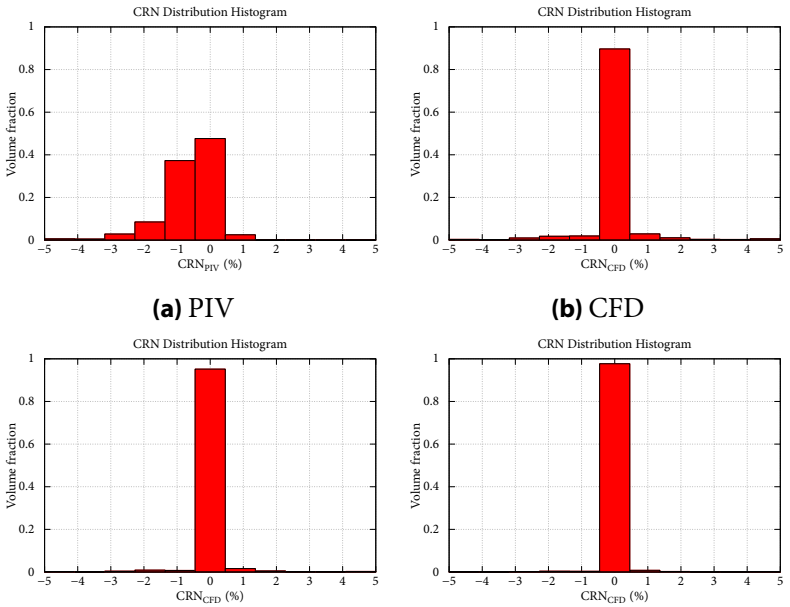



**(a)** Nominal Grid **(b)** ×4 Refined Grid




**(c)** ×16 Refined Grid

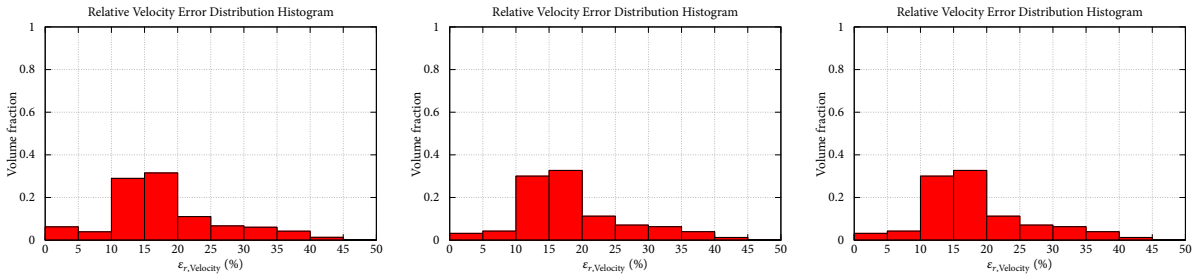
**Figure 5.19:** Absolute pressure error contours for A-HS-Closed.




**(a)** PIV **(b)** CFD

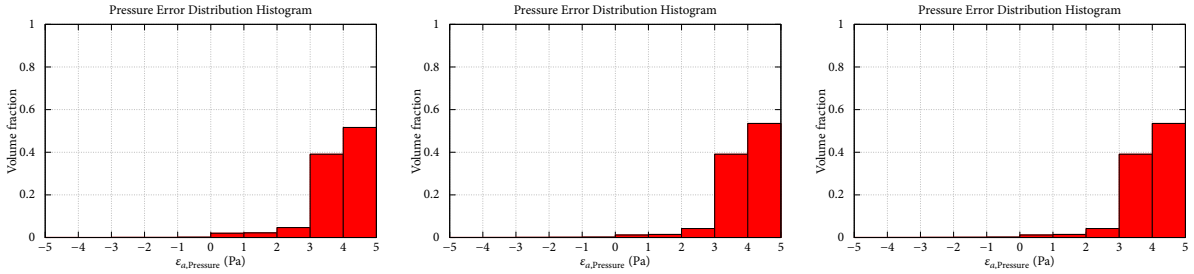
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.20:** Distribution of normalized continuity residual field for A-HS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.21:** Distribution of relative velocity error field for A-HS-Closed.

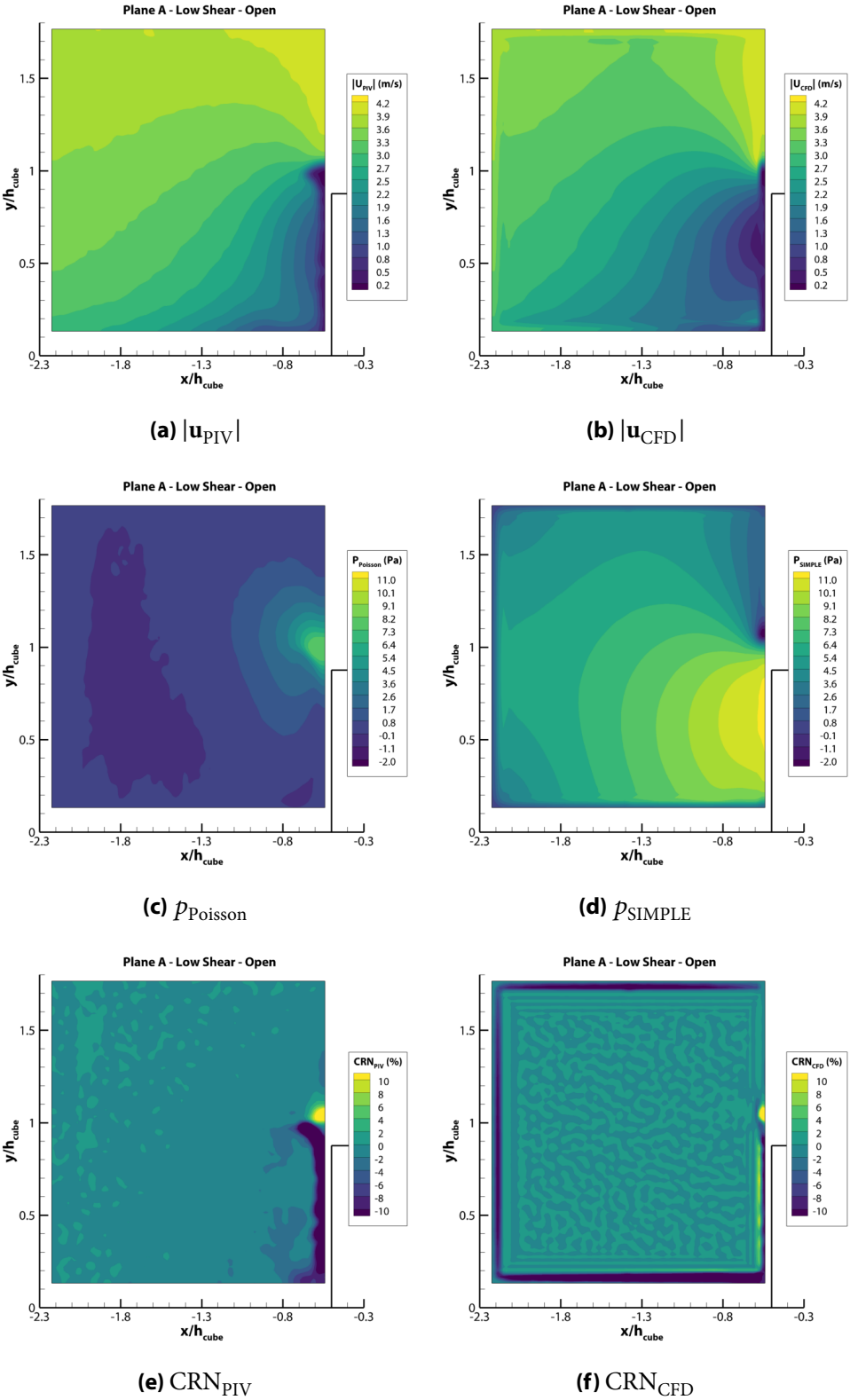



**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.22:** Distribution of absolute pressure error field for A-HS-Closed.

### 5.3.2 Low Shear Conditions

#### 5.3.2.1 Open Cube

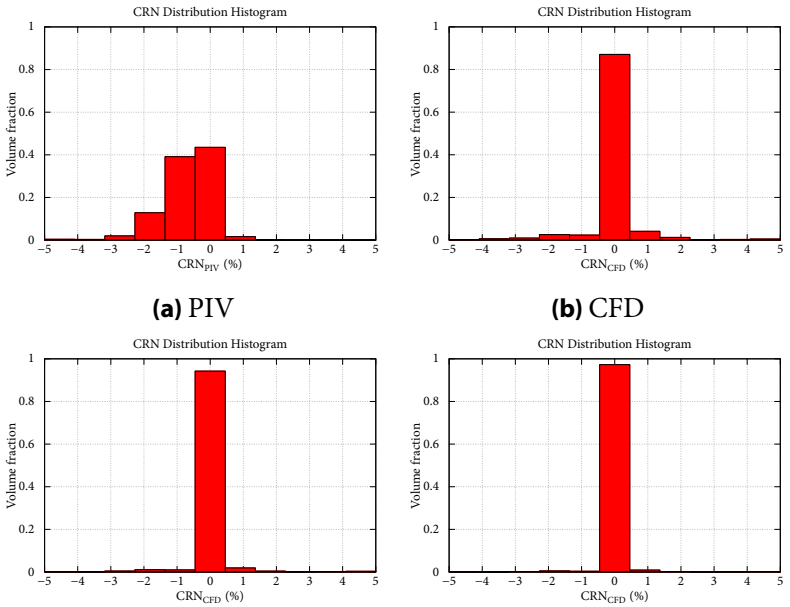



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

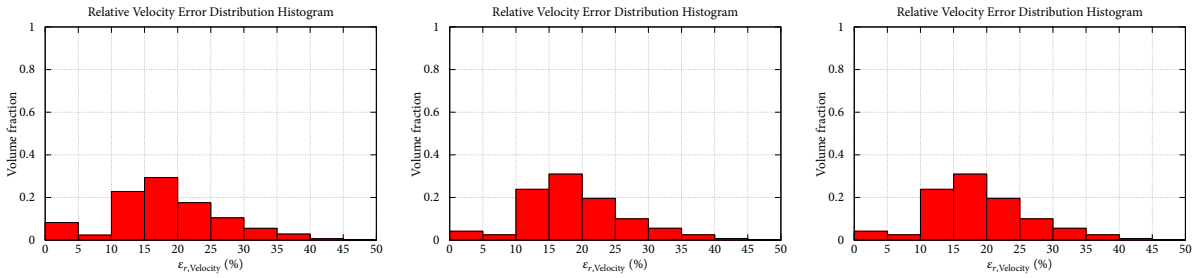
**Figure 5.23:** Contours for A-LS-Open.




**(a)** PIV **(b)** CFD

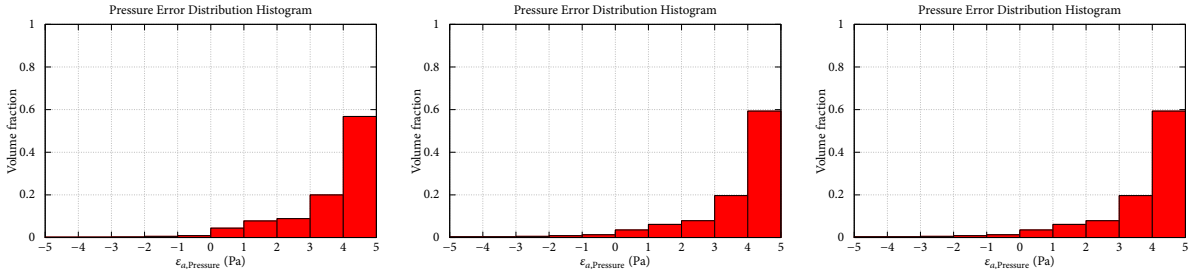
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.24:** Distribution of normalized continuity residual field for A-LS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

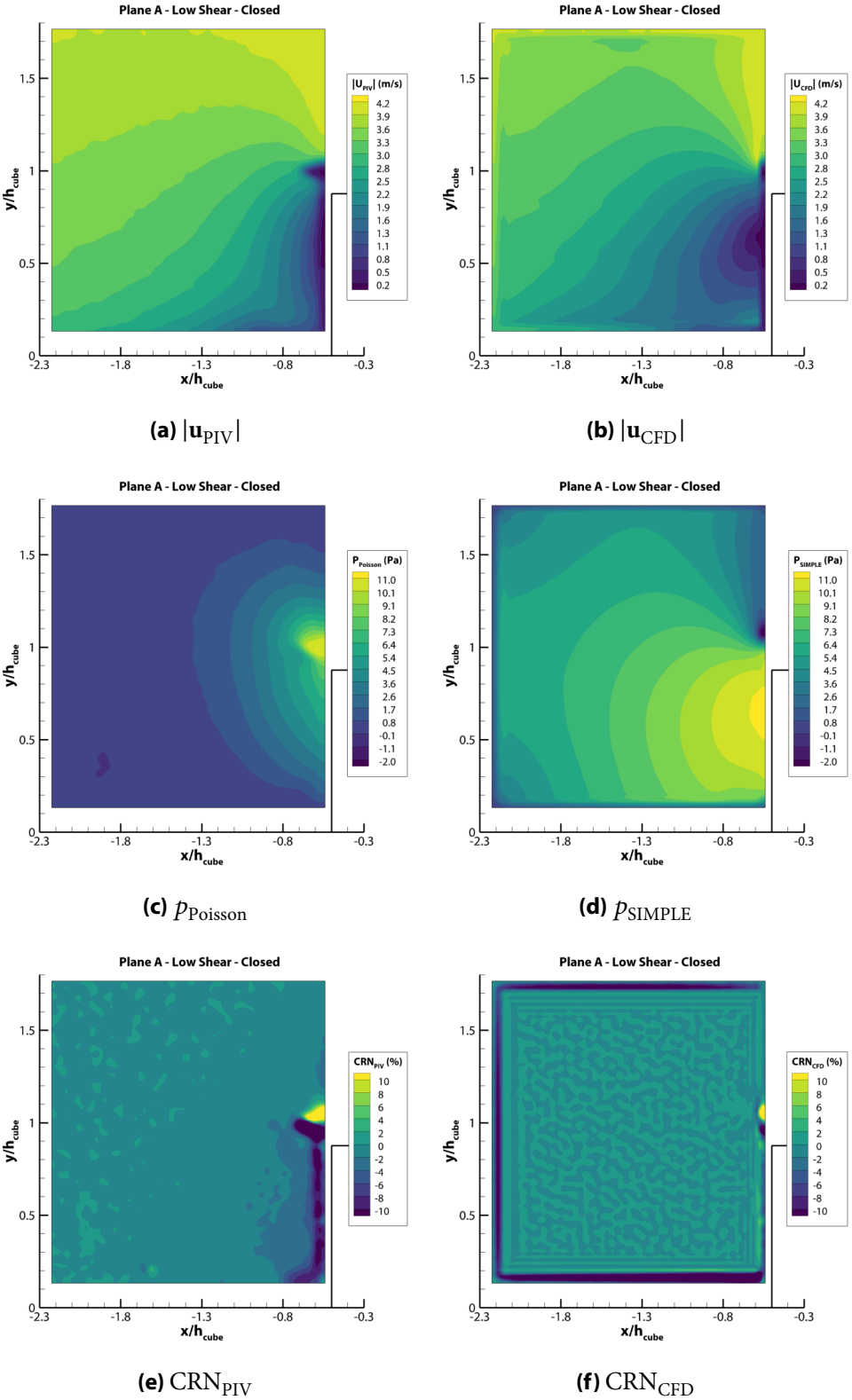
**Figure 5.25:** Distribution of relative velocity error field for A-LS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.26:** Distribution of absolute pressure error field for A-LS-Open.

#### 5.3.2.2 Closed Cube

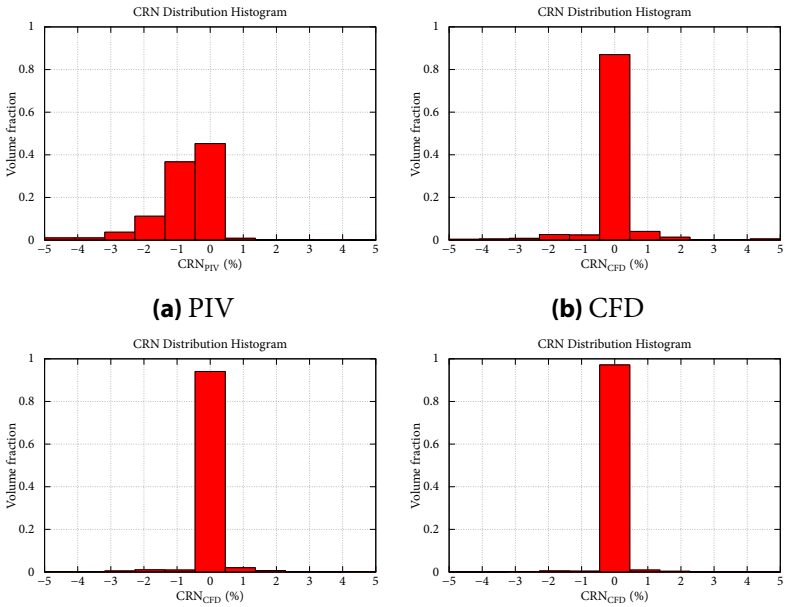



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

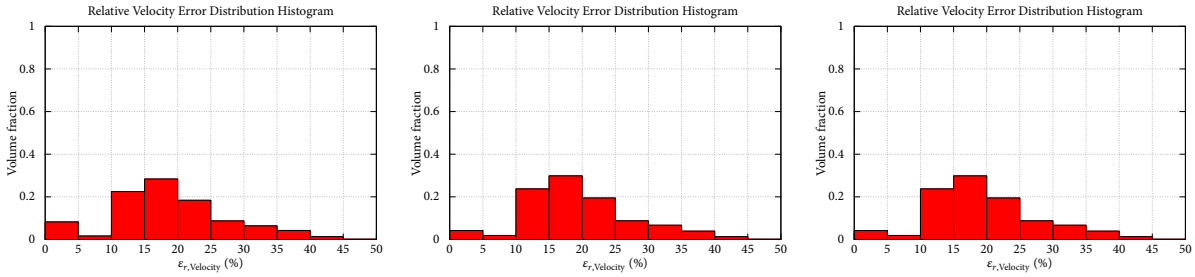
**Figure 5.27:** Contours for A-LS-Closed.




**(a)** PIV **(b)** CFD

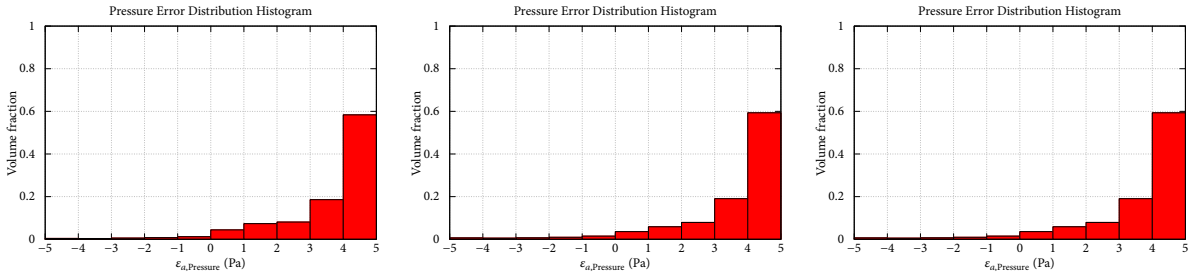
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.28:** Distribution of normalized continuity residual field for A-LS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.29:** Distribution of relative velocity error field for A-LS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.30:** Distribution of absolute pressure error field for A-LS-Closed.

## 5.4 Plane B

Then, the methods were applied to plane B. This plane lies above the cube, and a separation zone is evident in the south boundary of the plane. Firstly, as shown in the velocity magnitude contour plots, in figs. 5.32, 5.37, 5.41 and 5.45, the SIMPLE-based method managed to produce velocity fields that were in accordance with the experimental data. Additionally, both the Poisson and the SIMPLE-based methods produced similar pressure fields, as shown in the pressure contour plots, in figs. 5.32, 5.37, 5.41 and 5.45, the absolute pressure error distributions in figs. 5.35, 5.40, 5.44 and 5.48 and the $C_p$ distributions figs. 5.31 and 5.36.

This behavior is explained due to the lower CRN values compared to the other planes, as shown in the CRN contour plots in figs. 5.32, 5.37, 5.41 and 5.45 and the CRN distributions in figs. 5.33, 5.38, 5.42 and 5.46.

Additionally, the mesh refinement process does not significantly influence the resulting pressure and velocity fields, as shown in the relative velocity error distributions in figs. 5.34, 5.39, 5.43 and 5.47 and the absolute pressure error distributions in figs. 5.35, 5.40, 5.44 and 5.48.

**Table 5.3:** Number of cells for the different cases in plane B.

| Mesh | High Shear | | Low Shear | |
|---|---|---|---|---|
| | Open | Closed | Open | Closed |
| Nominal | 10094 | 10094 | 8652 | 8652 |
| ×4 | 40376 | 40376 | 34608 | 34608 |
| ×16 | 161504 | 161504 | 138432 | 138432 |

### 5.4.1 High Shear Conditions

#### 5.4.1.1 Open Cube

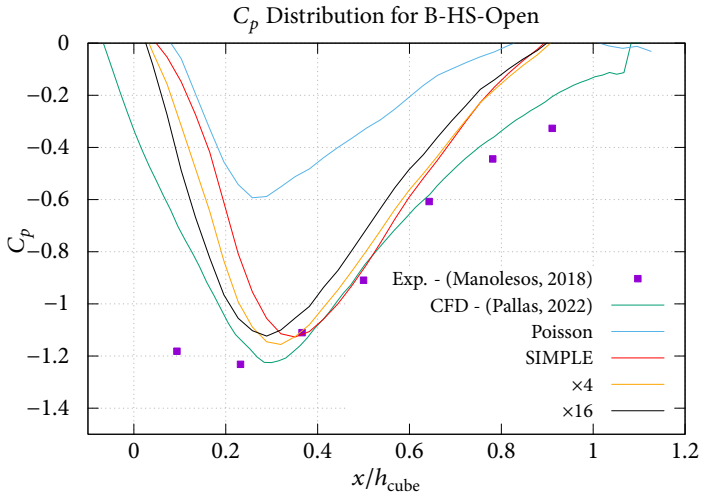
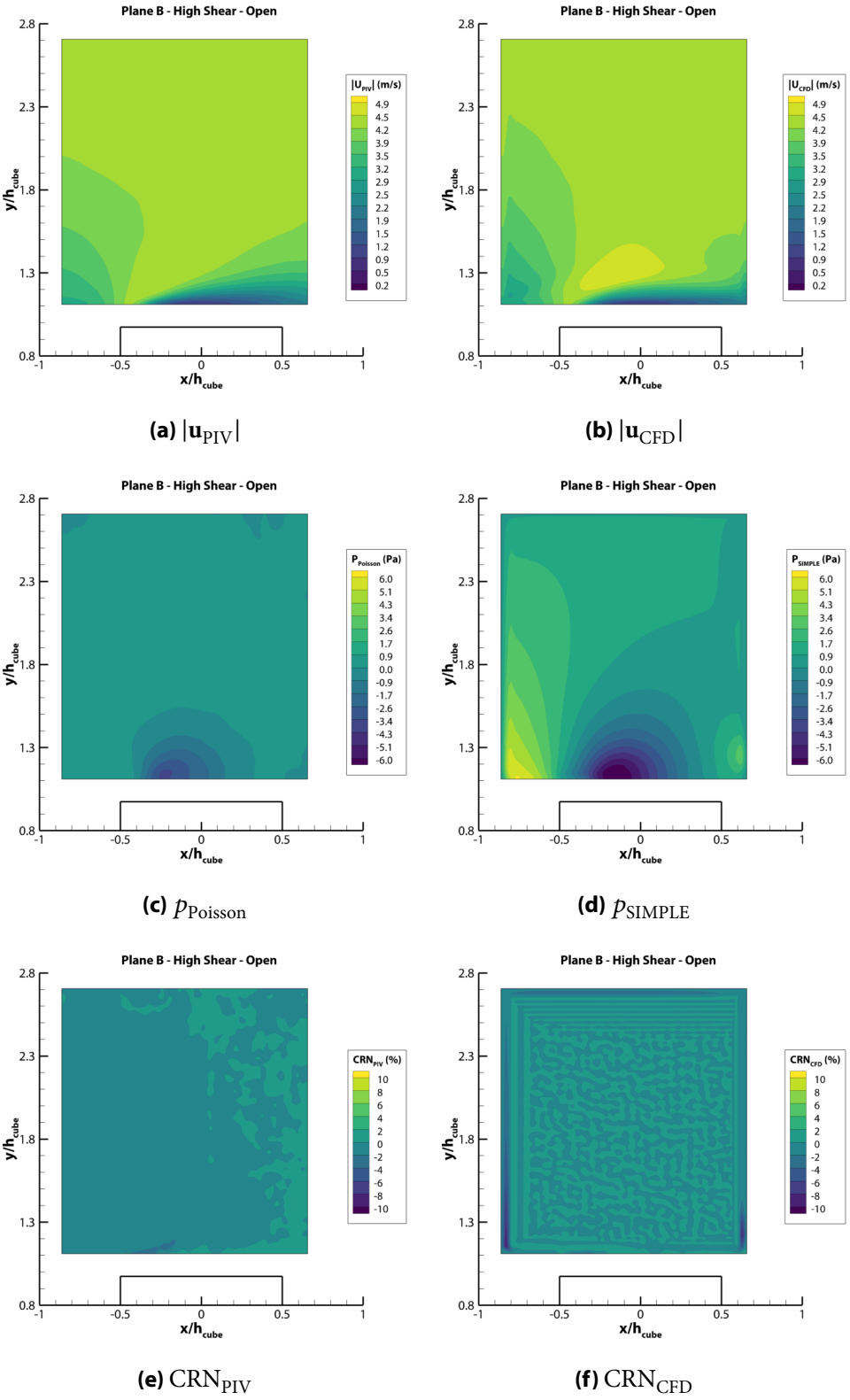



**Figure 5.31:** Pressure coefficient distribution along the cube for B-HS-Open.

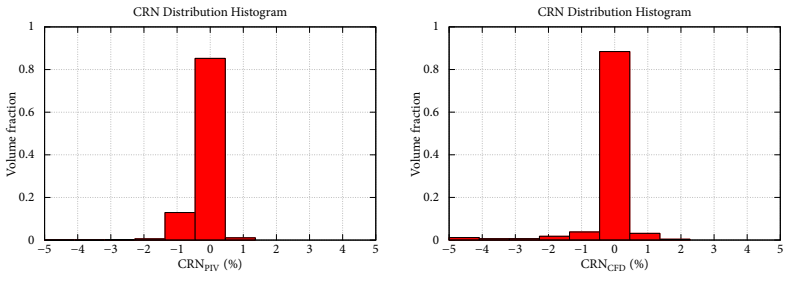



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

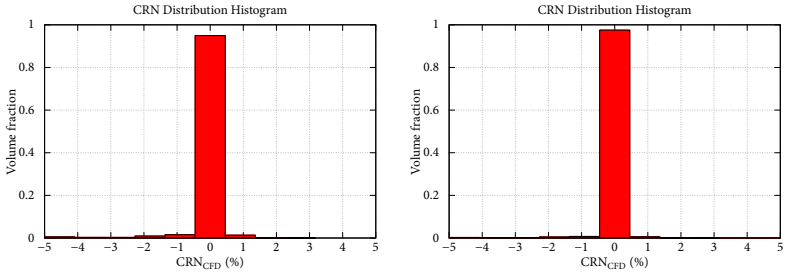
**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

**Figure 5.32:** Contours for B-HS-Open.

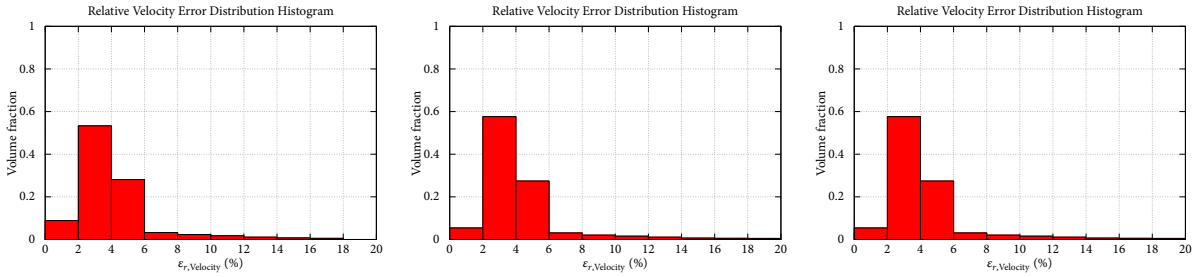



**(a)** PIV **(b)** CFD



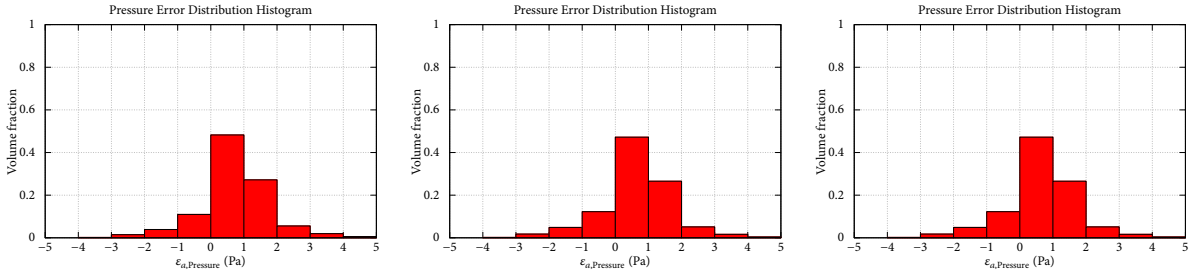

**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.33:** Distribution of normalized continuity residual field for B-HS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.34:** Distribution of relative velocity error field for CFD results on the three different grids, for B-HS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.35:** Distribution of absolute pressure error field for CFD results on the three different grids, for B-HS-Open.

#### 5.4.1.2 Closed Cube

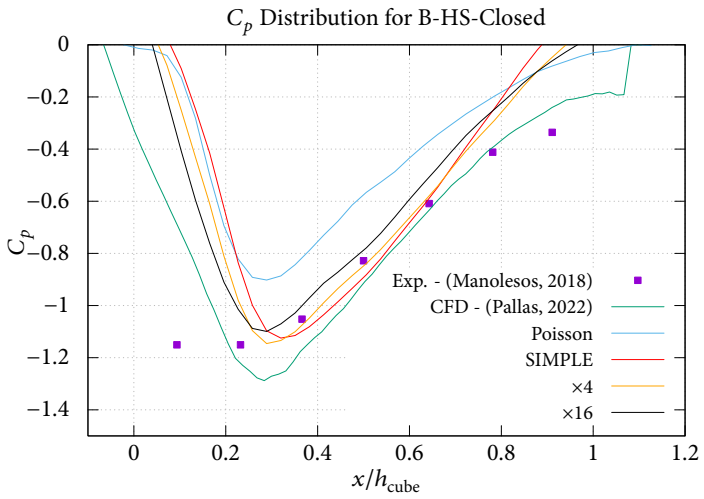
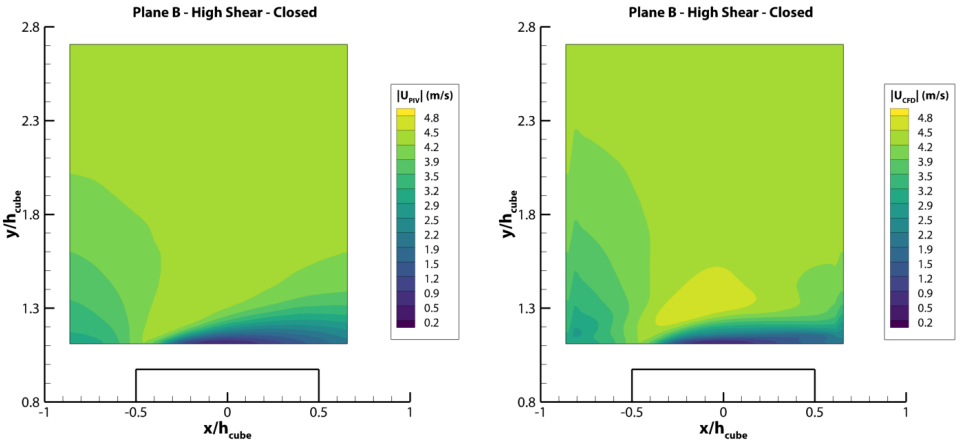


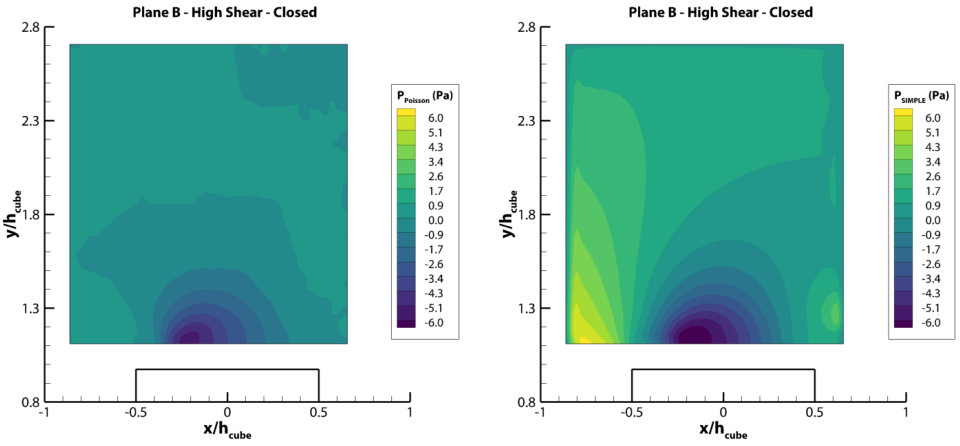

**Figure 5.36:** Pressure coefficient distribution along the cube for B-HS-Closed.




**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$



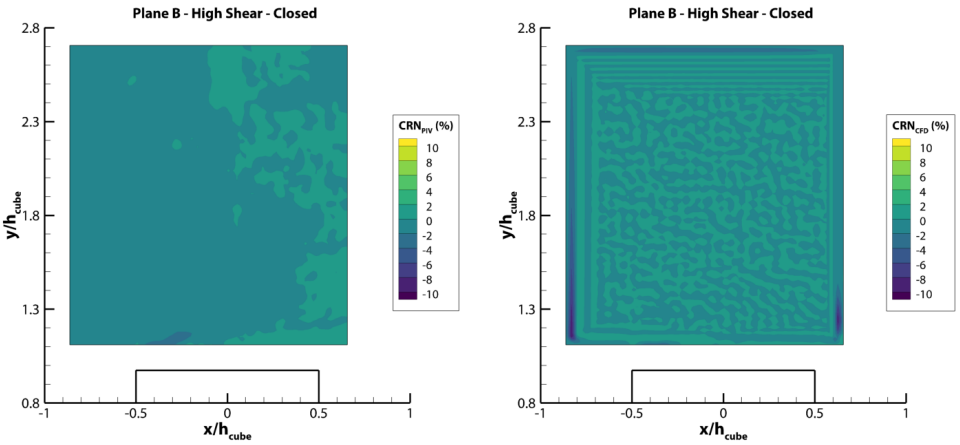

**(b)** $|\mathbf{u}_{\mathrm{CFD}}|$




**(c)** $p_{\mathrm{Poisson}}$




**(d)** $p_{\mathrm{SIMPLE}}$

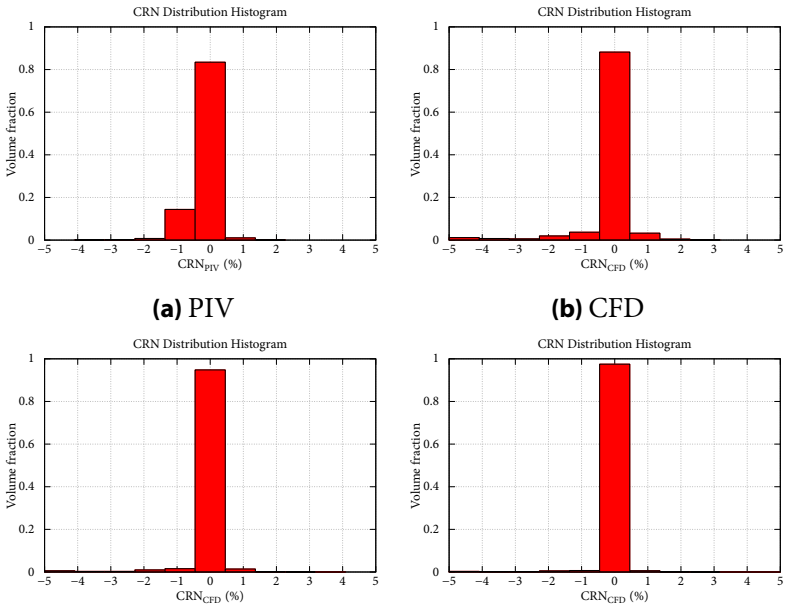



**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$




**(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

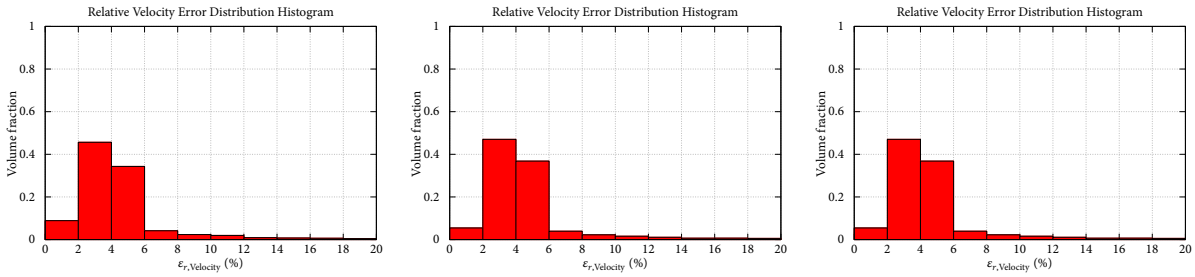
**Figure 5.37:** Contours for B-HS-Closed.




**(a)** PIV **(b)** CFD

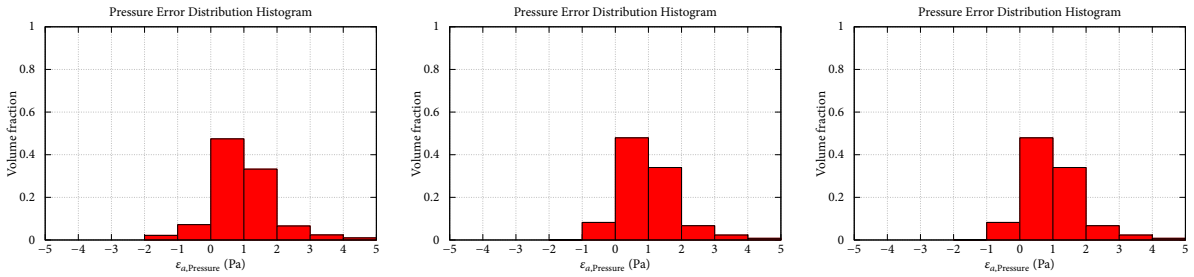
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.38:** Distribution of normalized continuity residual field for B-HS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.39:** Distribution of relative velocity error field for CFD results on the three different grids, for B-HS-Closed.

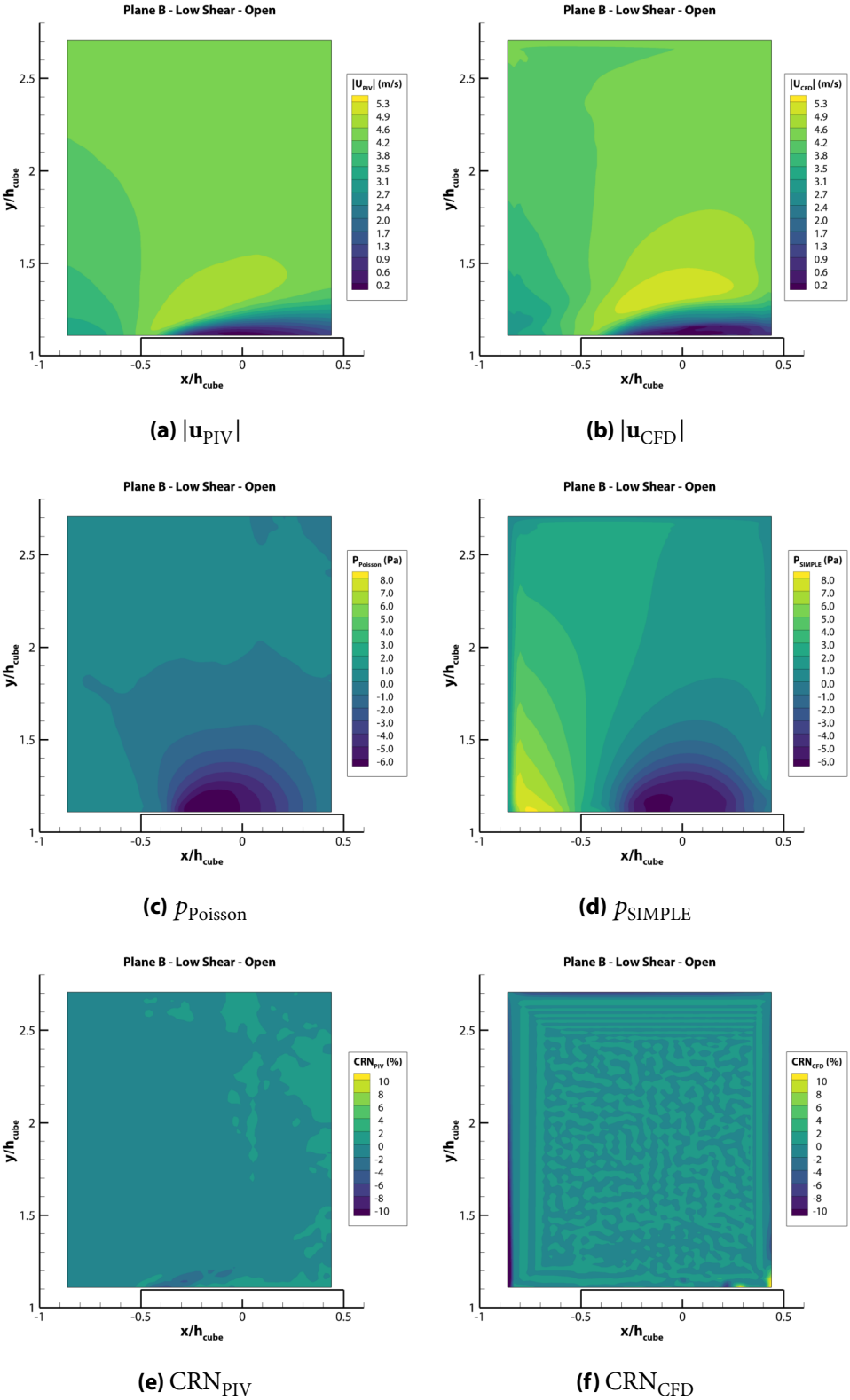



**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.40:** Distribution of absolute pressure error field for CFD results on the three different grids, for B-HS-Closed.

## 5.4.2 Low Shear Conditions

### 5.4.2.1 Open Cube

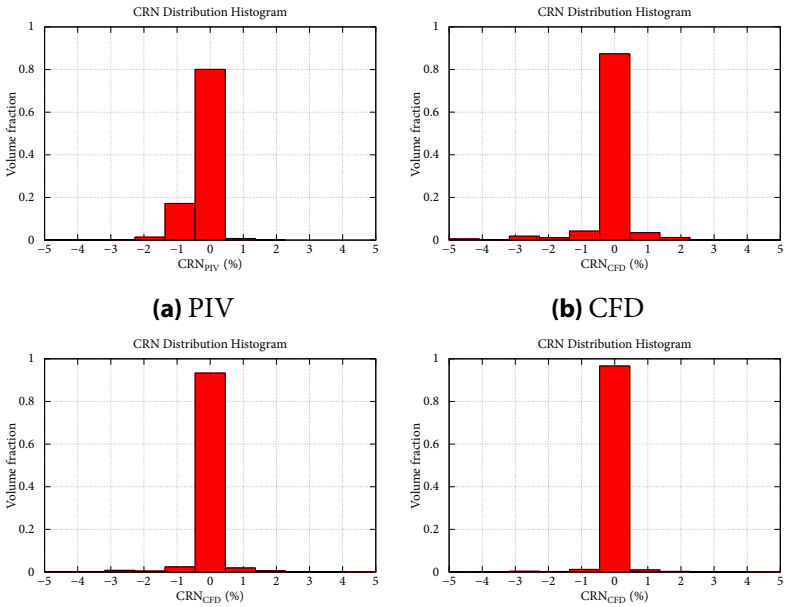



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

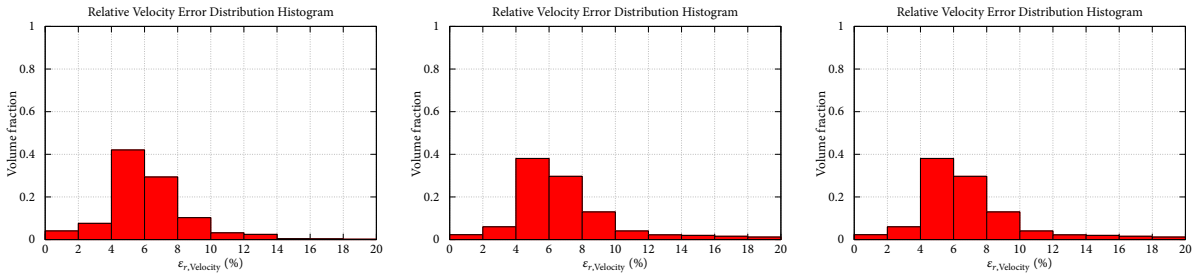
**Figure 5.41:** Contours for B-LS-Open.




**(a)** PIV **(b)** CFD

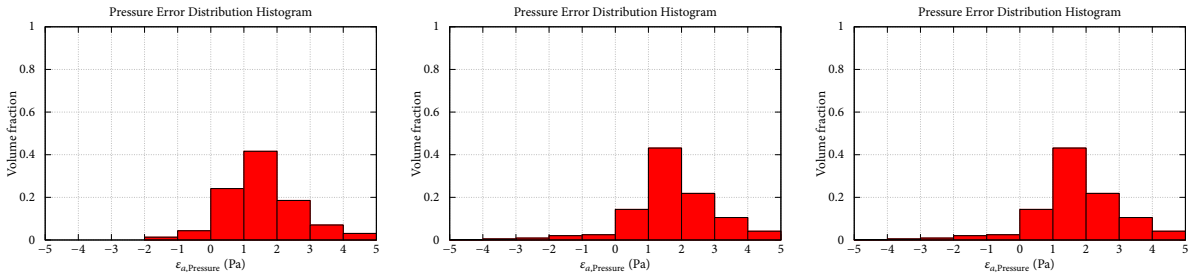
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.42:** Distribution of normalized continuity residual field for B-LS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

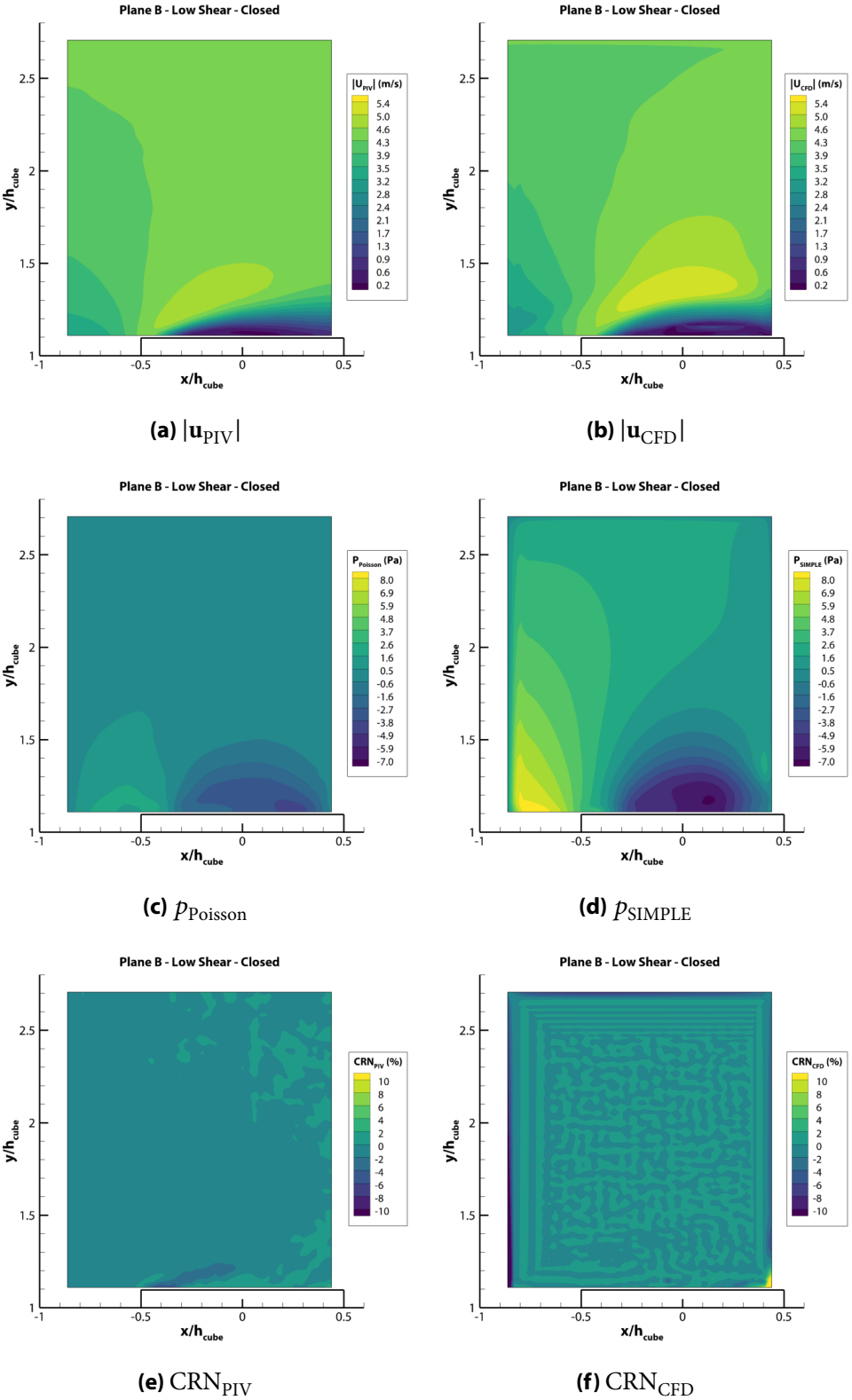
**Figure 5.43:** Distribution of relative velocity error field for CFD results on the three different grids, for B-LS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.44:** Distribution of absolute pressure error field for CFD results on the three different grids, for B-LS-Open.

#### 5.4.2.2 Closed Cube

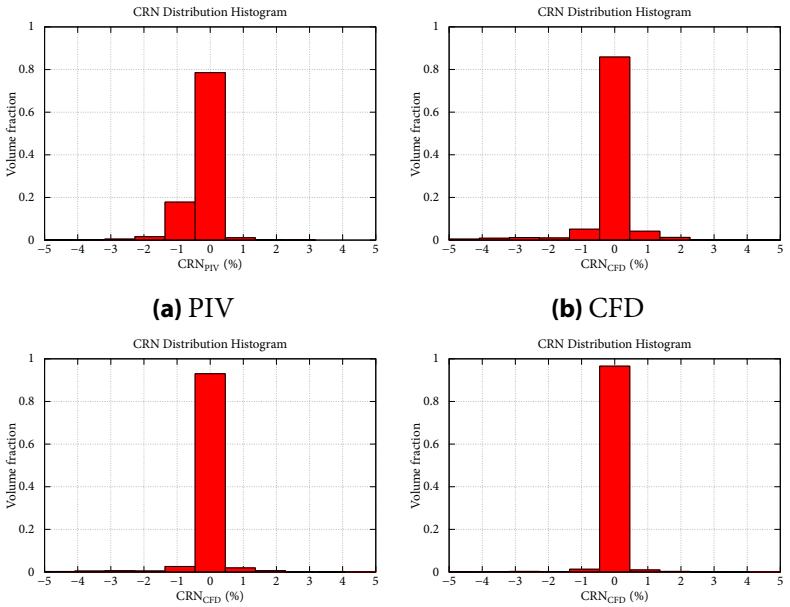



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

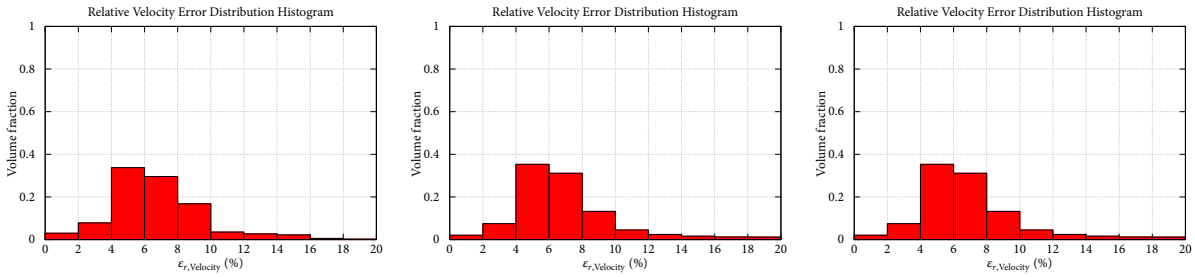
**Figure 5.45:** Contours for B-LS-Closed.




**(a)** PIV **(b)** CFD

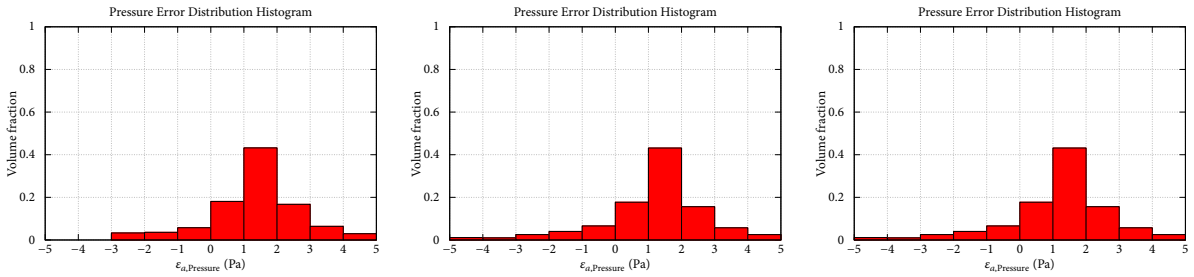
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.46:** Distribution of normalized continuity residual field for B-LS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.47:** Distribution of relative velocity error field for CFD results on the three different grids, for B-LS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.48:** Distribution of absolute pressure error field for CFD results on the three different grids, for B-LS-Closed.

## 5.5 Plane C

Finally, the methods were applied to plane C. This plane is located downwind of the cube, and as a result, a recirculation zone is observed. It is important to note that for the high shear conditions with the closed cube, the PIV (Nominal) mesh is notably coarser than the other case meshes, as shown in table 5.4, and the relevant measurement resolution is inferior.

Firstly, as shown in the velocity magnitude contour plots, in figs. 5.50, 5.55, 5.59 and 5.63, the SIMPLE-based method produces velocity fields close to the experimental data.

Additionally, as shown in the pressure contour plots in in figs. 5.50, 5.55, 5.59 and 5.63, the SIMPLE-based methods produces pressure fields that are superior to the ones produced by the Poisson equation, as evident by the $C_p$ distributions shown in fig. 5.49. Additionally, the mesh refinement process does not significantly influence the resulting pressure and velocity fields, as shown in the relative velocity error distributions in figs. 5.52, 5.57, 5.61 and 5.65 and the absolute pressure error distributions in figs. 5.53, 5.58, 5.62 and 5.66.

**Table 5.4:** Number of cells for the different cases in plane C.

| Mesh | High Shear | | Low Shear | |
|---|---|---|---|---|
| | Open | Closed | Open | Closed |
| Nominal | 11336 | 2107 | 11336 | 11336 |
| ×4 | 45344 | 8428 | 45344 | 45344 |
| ×16 | 181376 | 33712 | 181376 | 181376 |

### 5.5.1 High Shear Conditions

#### 5.5.1.1 Open Cube

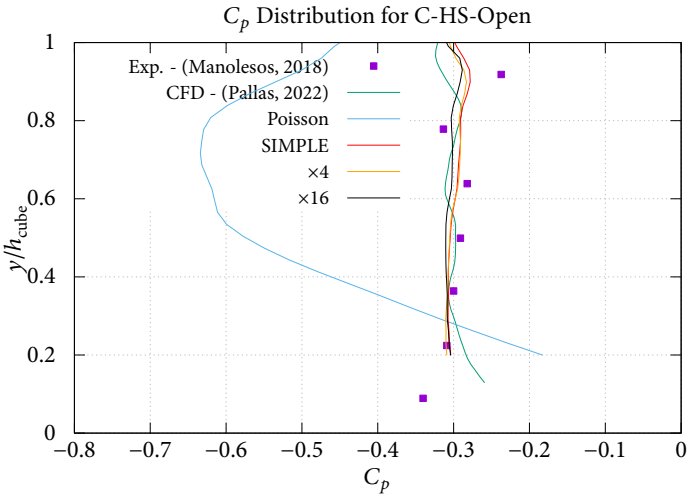
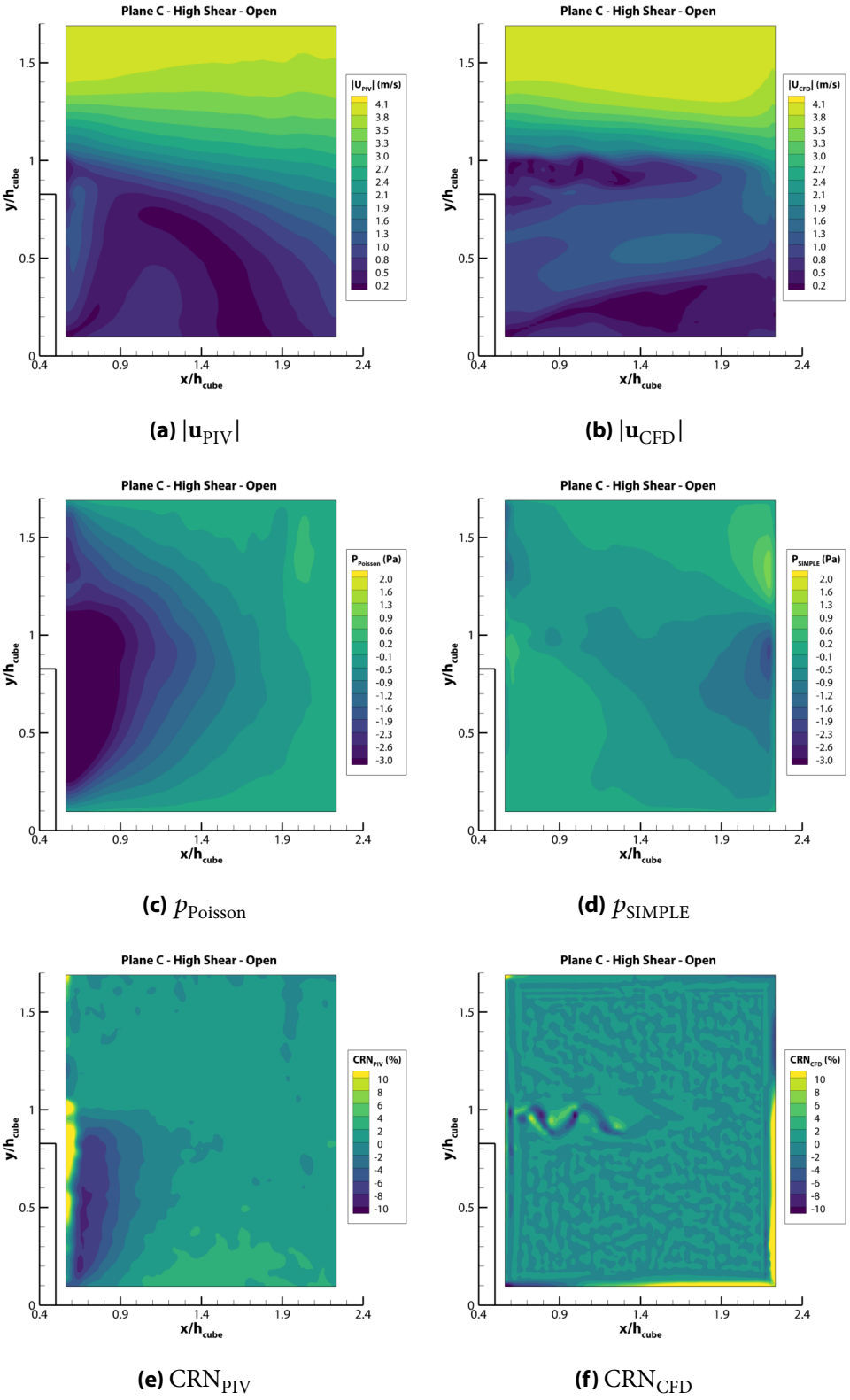



**Figure 5.49:** Pressure coefficient distribution along the cube for C-HS-Open.

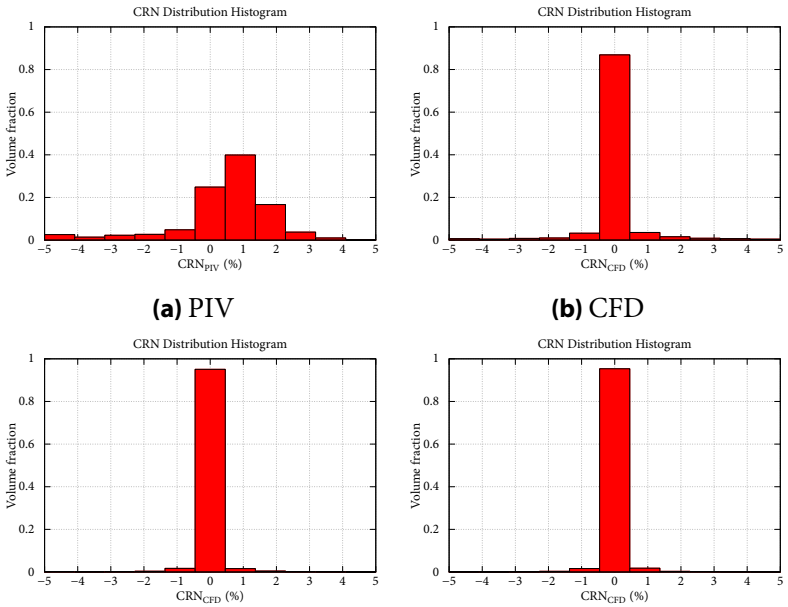



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

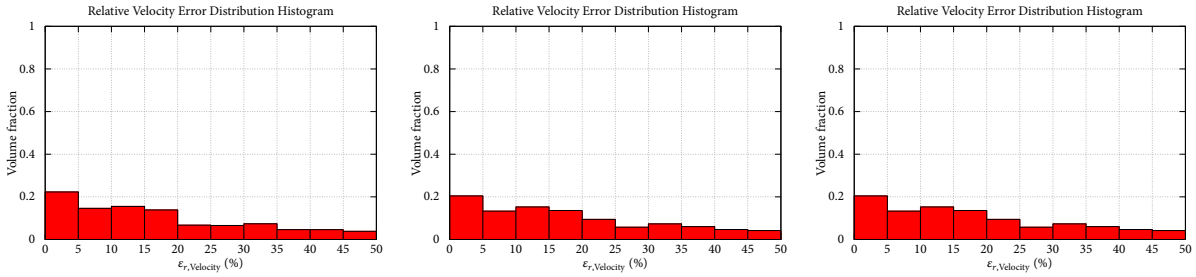
**Figure 5.50:** Contours for C-HS-Open.




**(a)** PIV **(b)** CFD

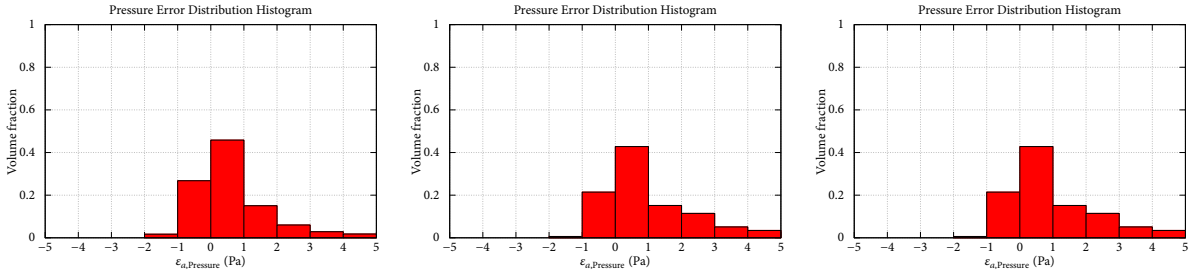
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.51:** Distribution of normalized continuity residual field for C-HS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.52:** Distribution of relative velocity error field for CFD results on the three different grids, for C-HS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.53:** Distribution of absolute pressure error field for CFD results on the three different grids, for C-HS-Open.

#### 5.5.1.2 Closed Cube

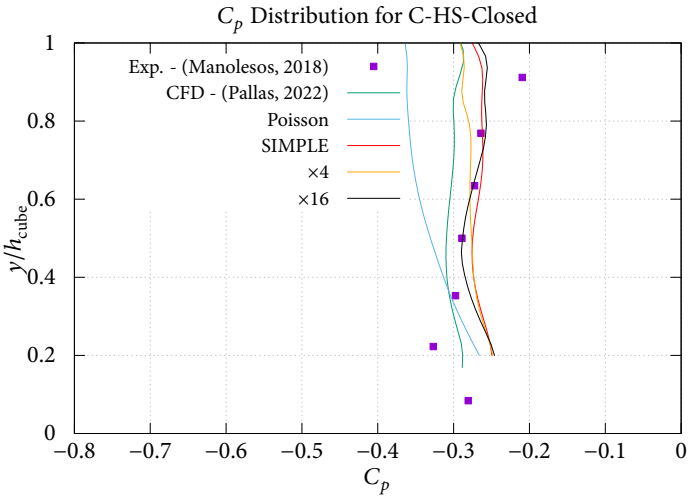
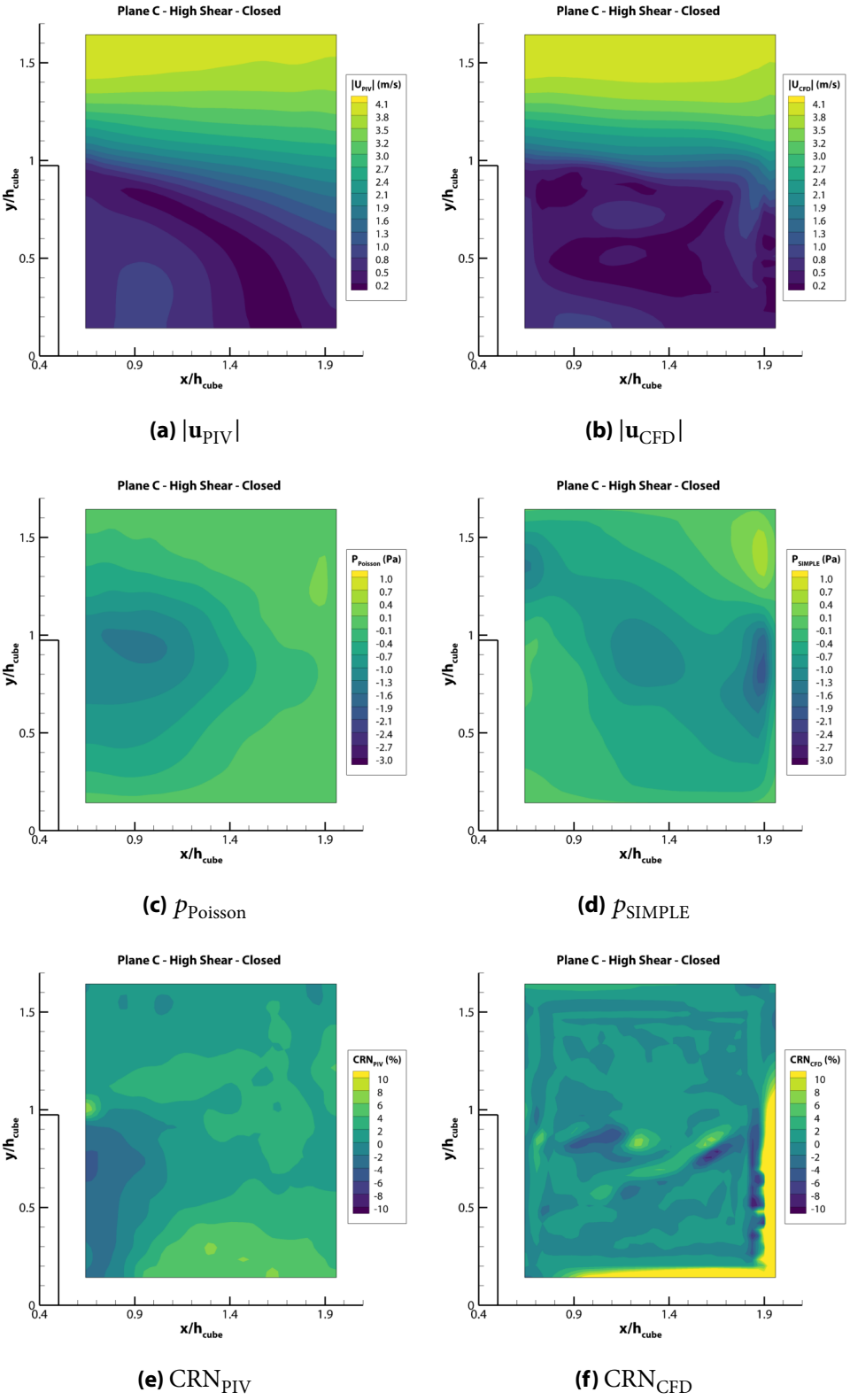



**Figure 5.54:** Pressure coefficient distribution along the cube for C-HS-Closed.

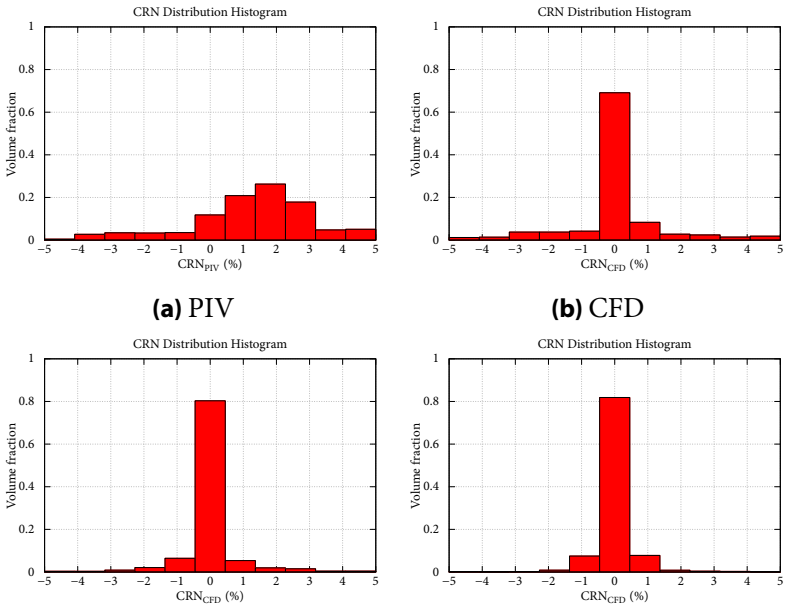



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

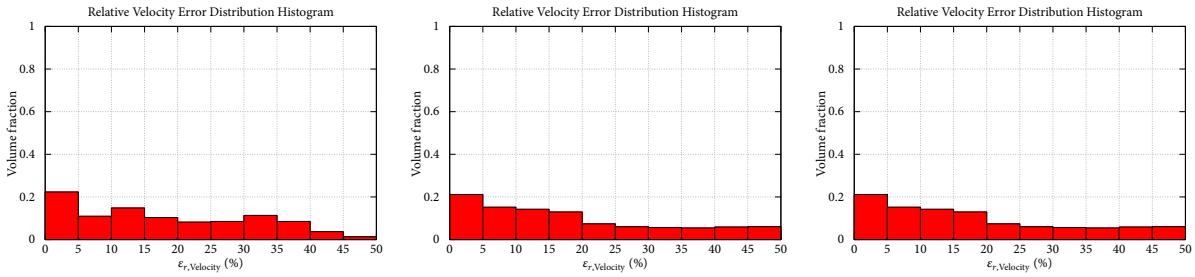
**Figure 5.55:** Contours for C-HS-Closed.




**(a)** PIV **(b)** CFD

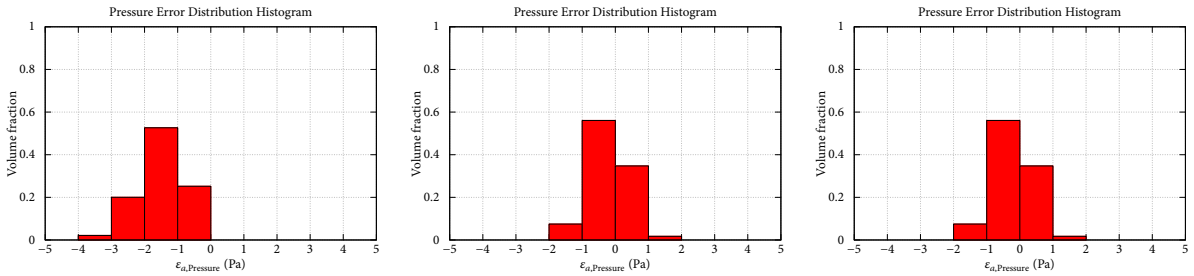
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.56:** Distribution of normalized continuity residual field for C-HS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.57:** Distribution of relative velocity error field for CFD results on the three different grids, for C-HS-Closed.

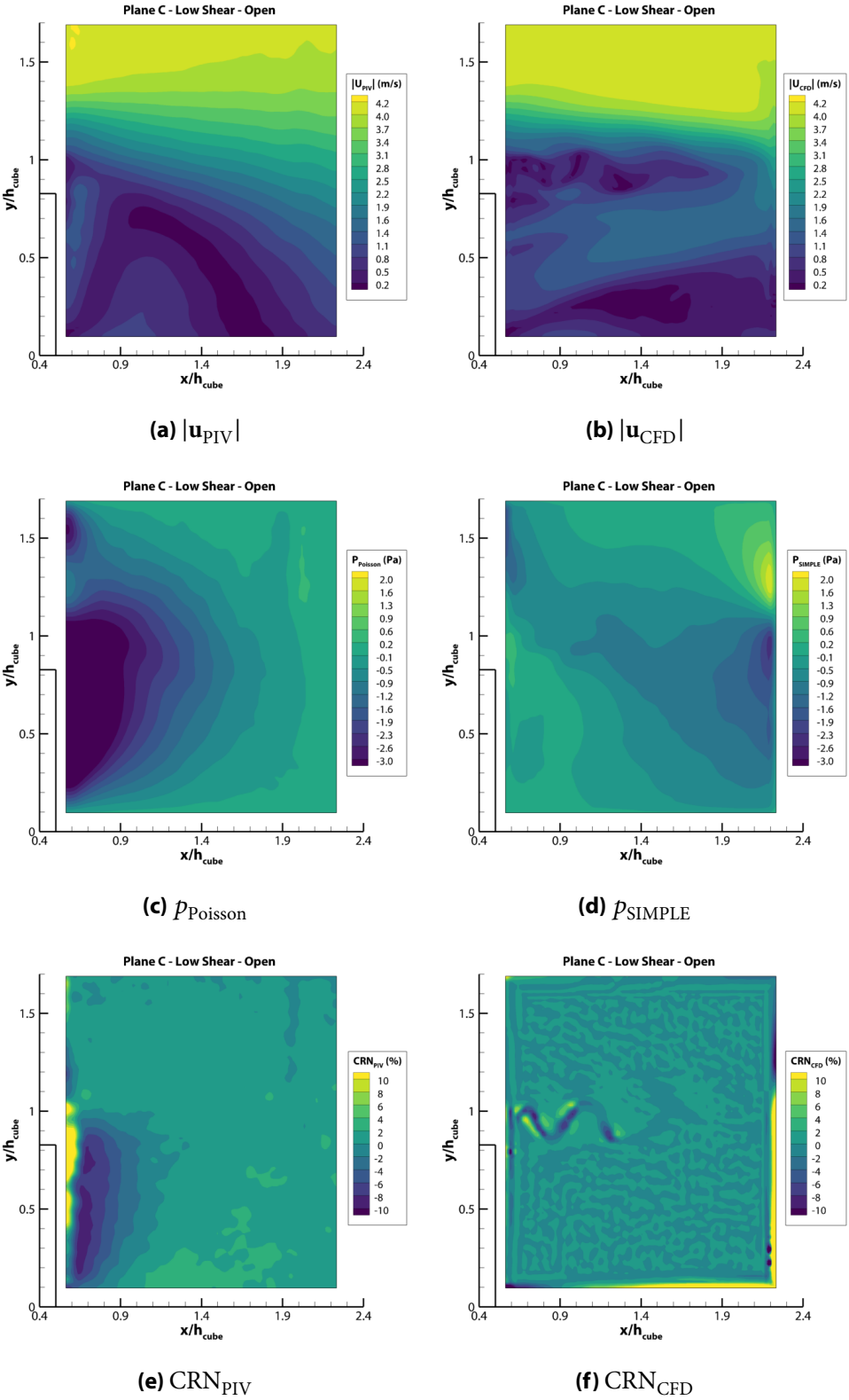



**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.58:** Distribution of absolute pressure error field for CFD results on the three different grids, for C-HS-Closed.

### 5.5.2 Low Shear Conditions

#### 5.5.2.1 Open Cube

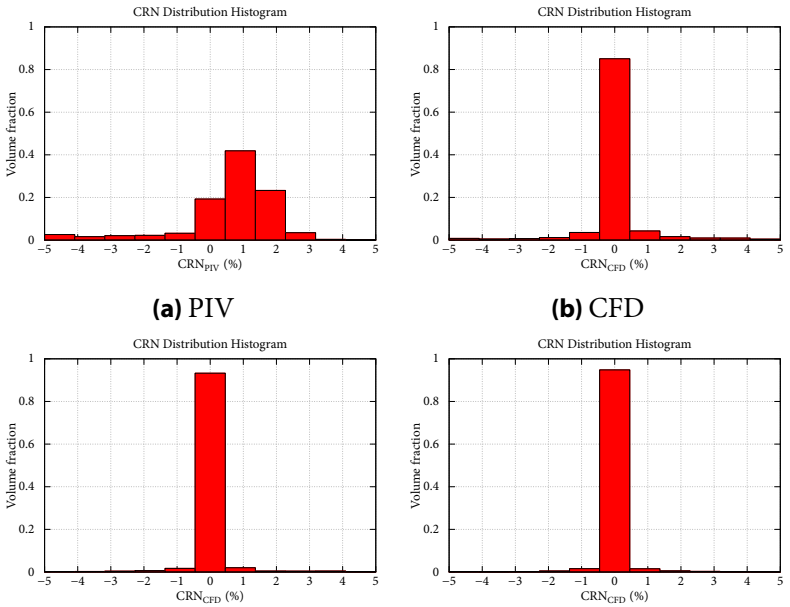



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

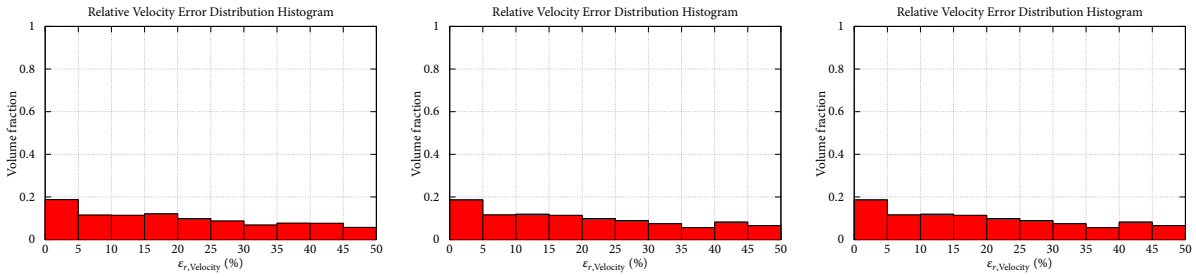
**Figure 5.59:** Contours for C-LS-Open.




**(a)** PIV **(b)** CFD

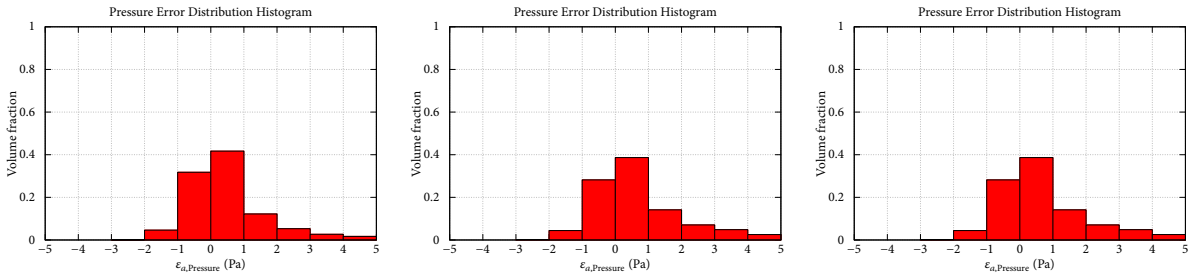
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.60:** Distribution of normalized continuity residual field for C-LS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

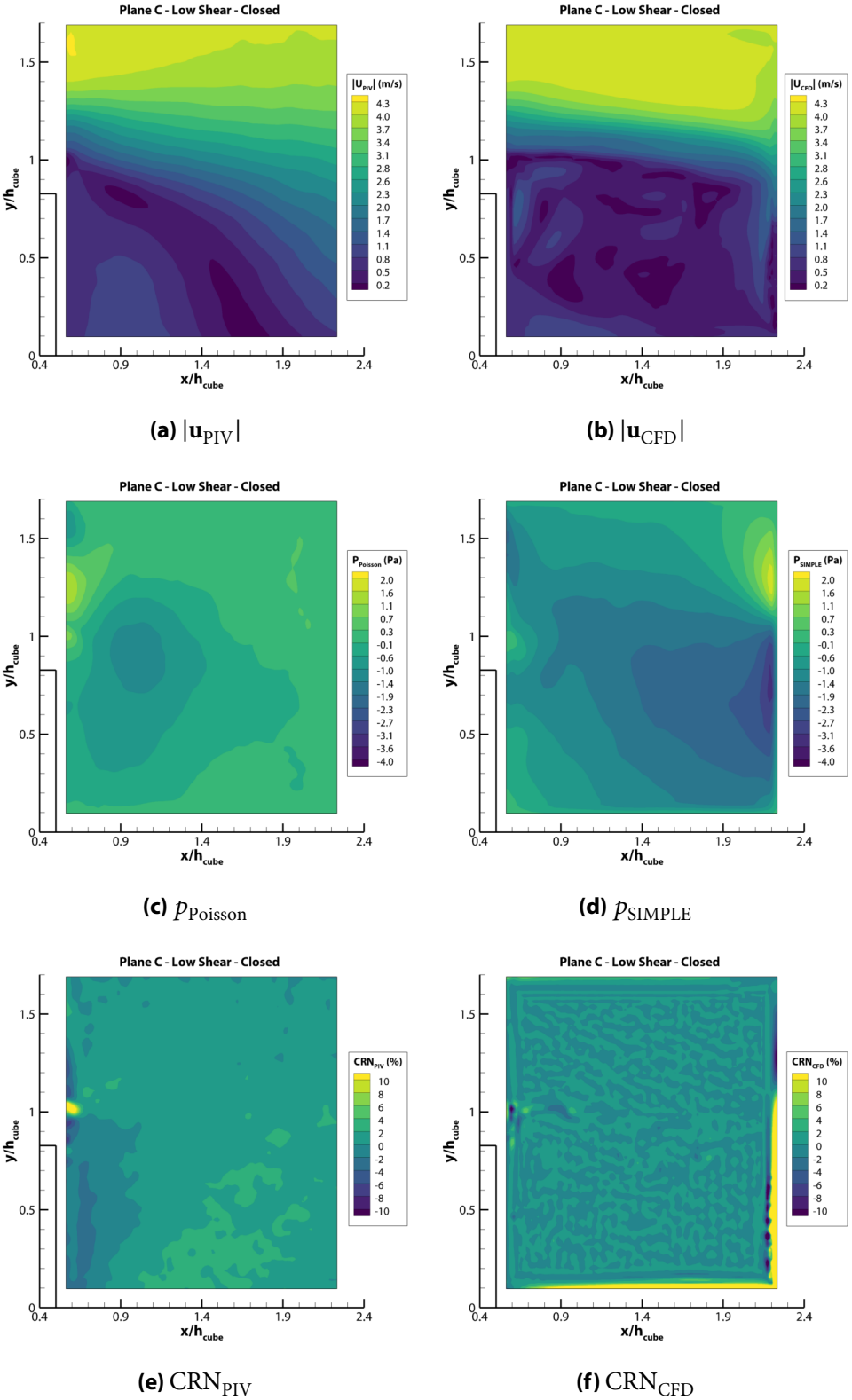
**Figure 5.61:** Distribution of relative velocity error field for CFD results on the three different grids, for C-LS-Open.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.62:** Distribution of absolute pressure error field for CFD results on the three different grids, for C-LS-Open.

#### 5.5.2.2 Closed Cube

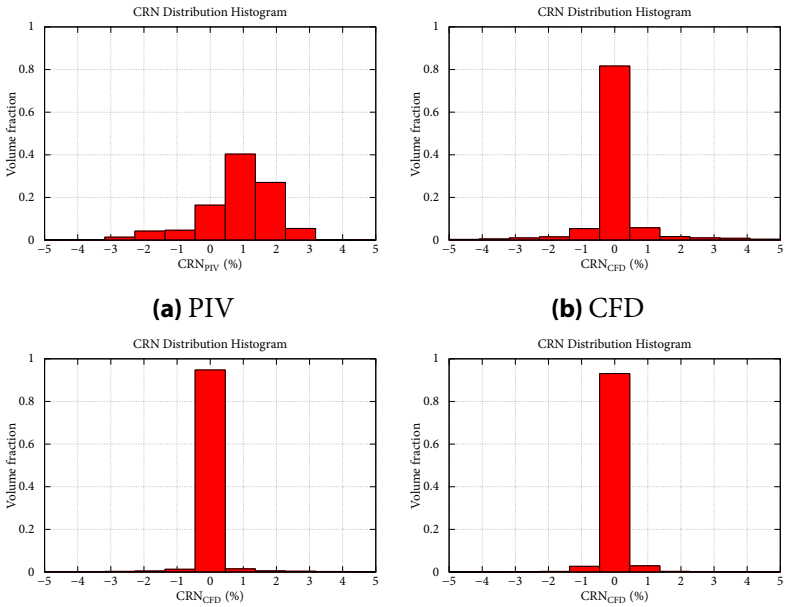



**(a)** $|\mathbf{u}_{\mathrm{PIV}}|$ **(b)** $|\mathbf{u}_{\mathrm{CFD}}|$

**(c)** $p_{\mathrm{Poisson}}$ **(d)** $p_{\mathrm{SIMPLE}}$

**(e)** $\mathrm{CRN}_{\mathrm{PIV}}$ **(f)** $\mathrm{CRN}_{\mathrm{CFD}}$

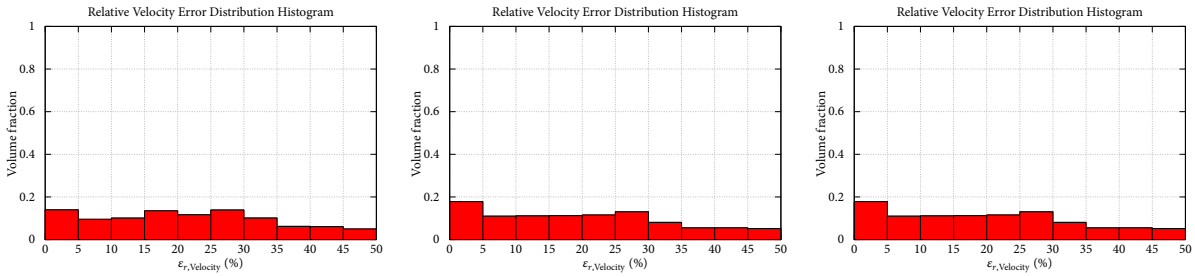
**Figure 5.63:** Contours for C-LS-Closed.




**(a)** PIV **(b)** CFD

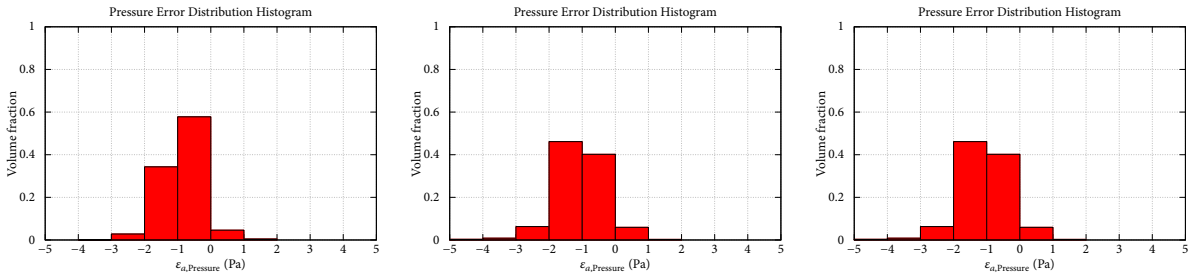
**(c)** ×4 Refined Grid **(d)** ×16 Refined Grid

**Figure 5.64:** Distribution of normalized continuity residual field for C-LS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.65:** Distribution of relative velocity error field for CFD results on the three different grids, for C-LS-Closed.




**(a)** Nominal Grid **(b)** ×4 Refined Grid **(c)** ×16 Refined Grid

**Figure 5.66:** Distribution of absolute pressure error field for CFD results on the three different grids, for C-LS-Closed.

## 5.6 Data Assimilation

DA (*Data Assimilation*) methods were applied to the problem via two different approaches: the nudging method and the KF (*Kalman Filter*) method. In this application, the flow is steady-state, but because it is solved with an iterative method, dynamic DA approaches can be used.

### 5.6.1 Nudging method

The nudging method was thoroughly described in section 3.4. The steady-state NS equations are modified by adding the nudging forcing term that is proportional to the misfit between the velocity field provided by CFD and the PIV measurements. In this application, the CFD meshes were identical to the PIV meshes; thus, $\mathbf{H}$ is the identity operator and the modified RANS equations are given by

$$\nabla \cdot \mathbf{u} = 0 \tag{5.16}$$

$$\rho \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \rho \nabla \cdot (\nu \nabla \mathbf{u}) - \rho \nabla \cdot \left( \overline{\mathbf{u}'_{\mathrm{PIV}} \mathbf{u}'_{\mathrm{PIV}}} \right) + \mathbf{K} (\mathbf{u}_{\mathrm{PIV}} - \mathbf{u}) \tag{5.17}$$

all of the above fields correspond to the mean fields, but the overbar symbols were dropped for simplicity.

The modification described above was implemented in the developed OpenFOAM solver, and two different options for the selection of the gain matrix $\mathbf{K}$ were implemented:

1. Identity nudging
2. BLUE nudging

#### 5.6.1.1 Identity nudging

The most simple option was to avoid any assumptions about the error structures and use the identity matrix as the gain matrix. In this approach, the numerical solution and observation misfit was constantly forced into the system equations without any other considerations. This approach is the least effective, but it also provided a baseline to see if nudging managed to improve the final solution. Thus, the gain matrix was given by

$$\mathbf{K} = \mathbf{I} \tag{5.18}$$

#### 5.6.1.2 BLUE nudging

Next, the BLUE estimator was used to calculate a gain matrix that minimizes the error variance. As described in eq. (3.19) and taking into account that $\mathbf{H} = \mathbf{I}$, the gain matrix is given by

$$\mathbf{K} = \mathbf{P} (\mathbf{P} + \mathbf{R})^{-1} \tag{5.19}$$

where $\mathbf{P}$ and $\mathbf{R}$ are the forecast and measurement error covariance matrices, respectively.

The measurement error covariance matrix is determined using the Reynolds stress tensor. Assuming there was no correlation between the measurements in different spatial positions and there was no correlation between the measurements of the horizontal and vertical components of the velocity, the measurement error covariance matrix is considered diagonal and is given by

$$\mathbf{R} = \sigma^2_{\mathrm{PIV},i} \delta_{ij} \tag{5.20}$$

where $\delta_{ij}$ is the Kronecker delta.

In the BLUE approach, the forecast error covariance matrix $\mathbf{P}$ is equal to the model error covariance matrix $\mathbf{Q}$. The task of determining the model error covariance matrix is more complex, as discussed in section 3.2.3. Therefore, a reduced-cost option was implemented by making some assumptions about its structure based on the knowledge of CFD modeling [6]. The basic assumption is that it is diagonal and depends only on the spatial (and temporal for transient problems) discretization. This is a valid assumption, as the spatial and temporal discretizations, on the one hand, influence the flow features that can be resolved, e.g., turbulence scales, recirculation length, etc. But on the other hand, they affect the order of discretization/truncation error of the numerical discretization schemes that were used. Based on these assumptions, the model error covariance matrix is considered a power function of the average cell length, $\Delta x$, and the time-step $\Delta t$, given by [6]

$$\mathbf{Q} = C\left(\Delta x^{\mathrm{so}} \Delta t^{\mathrm{to}}\right)\mathbf{I} \tag{5.21}$$

where the so (spatial order) and to (temporal order) exponents correspond to the minimum order of accuracy of the used discretization schemes. Additionally, $C$ is a used-defined coefficient that ensures dimensional similarity. In this application, the problem is steady-state, and the minimum order of the used schemes is one, thus the model error covariance matrix becomes

$$\mathbf{Q} = C\Delta x\mathbf{I} \tag{5.22}$$

The assumption that $\mathbf{Q}$ is diagonal may be considered aggressive, but because a Poisson equation was used for the pressure equation; it does not wrongly influence the solution. This is because Poisson equations are elliptical PDEs, meaning that information is propagated instantly throughout the whole domain. Thus, the model error is propagated along the neighboring cells without needing a more complex model for the model error correlation between neighboring cells.

### 5.6.2 Kalman Filter

Next, a more complex DA method was implemented, the KF (*Kalman Filter*) [5]. The KF is a sequential DA method and consists of a prediction and a correction step, as described in section 3.2.2. It also requires a matrix that represents the system dynamics in order to model the evolution of the forecast error covariance matrix. Thus, an approximation of the system matrix is needed in order to formulate the KF in this application.

Firstly, it is worth noting that NS equations are non-linear, but because in this application, a steady-state case was studied using an iterative method that contains a Poisson equation, it was considered that no linearization was needed to obtain the simplified state matrix. Additionally, in the spirit of the approximation of the model error covariance matrix, in BLUE, the state matrix was selected to be diagonal and equal to the inverse of the diagonal coefficient matrix [6], thus it is given by

$$\mathbf{M} = \mathbf{A}^{-1}\mathbf{I} \tag{5.23}$$

where $\mathbf{A}$ is the matrix that contains all the diagonal coefficients $a_P$, as shown in eq. (5.2).

Thus, all the matrices that influence the calculation of $\mathbf{K}$ are initialized as diagonal matrices, the $\mathbf{M}$ matrix that propagates the forecast error covariance matrix through the iterative process is also diagonal, and consequently, $\mathbf{K}$ is also diagonal. This structure of $\mathbf{K}$, on the one hand, is beneficial from a computational cost point of view, and it also allows for different KF formulations, ensuring that they don't disturb the diagonal dominance of the equation matrices used in the

SIMPLE method. On the other hand, the calculated gain matrix still minimizes the a posteriori error covariance equation, but the a priori and measurement error structures are simplified.

Two formulations were implemented for using the KF with the SIMPLE algorithm.

1. The classic formulation of the KF
2. An integrated formulation of the KF [7]

#### 5.6.2.1 Classic Formulation

Firstly, the classic formulation was implemented. Each iteration of the SIMPLE algorithm was used as the model that produces the forecast state for the KF. Then, the forecast error covariance matrix and the optimal gain matrix were calculated, and the analysis state was obtained using the innovation term. The algorithm can be summarized as follows:

- Initial guess of $p$, $\mathbf{u}$ and $\mathbf{P} = \mathbf{Q}$
- Until convergence is reached:
  1. Momentum predictor: Solve discretized momentum equation, eq. (5.2) $\rightarrow \mathbf{u}^*$
  2. Pressure predictor: Solve pressure equation, eq. (5.5) $\rightarrow p^*$
  3. Correct volumetric fluxes, eq. (5.6)
  4. Relax pressure field, $\rightarrow p^{k+1}$
  5. Momentum corrector: Solve velocity correction equation eq. (5.7) $\rightarrow \mathbf{u}_*^{k+1}$
  6. Update forecast error covariance matrix: $\mathbf{P}_{k+1}^* = \mathbf{M}\mathbf{P}_k\mathbf{M}^\mathsf{T} + \mathbf{Q}$
  7. Update gain matrix: $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^* \left(\mathbf{P}_{k+1}^* + \mathbf{R}\right)^{-1}$
  8. Update state: $\mathbf{u}^{k+1} = \mathbf{u}_*^{k+1} + \mathbf{K}_{k+1}\left(\mathbf{u}_{\text{PIV}} - \mathbf{u}_*^{k+1}\right)$
  9. Update forecast error covariance matrix: $\mathbf{P}_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\right)\mathbf{P}_{k+1}^*$
- Finalize procedure

Thus, in each iteration of the SIMPLE algorithm, the final velocity field is updated, taking into account for the forecast error covariance of the iteration, which consists of a component that is propagated throughout the iterations using the $\mathbf{M}$ matrix and the $\mathbf{Q}$ component that corresponds to the model error of one SIMPLE iteration. The constant addition of the $\mathbf{Q}$ to the $P$ may seem counterintuitive, as the purpose of the KF is to minimize $P$, and for a steady-state problem, the constant addition of the $\mathbf{Q}$ may be considered as an accumulation of errors that do not represent the actual model behavior. But, under the developed KF framework, each SIMPLE *iteration* is considered as the system model, thus the KF manages to minimize $P$ for each iteration and the convergence of the SIMPLE algorithm is not disturbed.

#### 5.6.2.2 Integrated Formulation

A formulation that combines the two prediction/correction schemes used in this application, the SIMPLE and the KF algorithms, into one prediction correction scheme [7] was implemented. Firstly, the velocity field is predicted using the momentum equation, and then updated using the KF gain matrix. Then, the modified pressure equation is solved, that is given by

$$\nabla \cdot \left(\frac{1}{a_P^*}\nabla p_P^*\right)_f = \sum_f \mathbf{S}_f \cdot \left(\frac{1}{a_P^*}\mathbf{H}(\mathbf{u}^*) + \frac{1}{a_P^*}\mathbf{K}_k\left(\mathbf{u}_{\text{PIV}} - \mathbf{u}^*\right)\right)_f \tag{5.24}$$

Next, after the solution of eq. (5.24), the volumetric fluxes are corrected, and the velocity field is calculated again, using eq. (5.7), without applying the KF. Thus, it is the augmented velocity that field has zero divergence (the mass continuity is enforced) instead of the intermediate velocity field. This approach is of great interest as the non-zero divergence of the PIV velocity field is taken into account only in the pressure calculation, and its effect is reduced due to the asymptotic nature of the KF. The integrated procedure can be summarized as follows:

- Initial guess of $p$, $\mathbf{u}$ and $\mathbf{P} = \mathbf{Q}$
- Until convergence is reached:
  1. Momentum predictor: Solve discretized momentum equation, eq. (5.2) $\rightarrow \mathbf{u}^*$
  2. Update forecast error covariance matrix: $\mathbf{P}^*_{k+1} = \mathbf{M}\mathbf{P}_k\mathbf{M}^\mathsf{T} + \mathbf{Q}$
  3. Update gain matrix: $\mathbf{K}_{k+1} = \mathbf{P}^*_{k+1}\left(\mathbf{P}^*_{k+1} + \mathbf{R}\right)^{-1}$
  4. Update forecast error covariance matrix: $\mathbf{P}_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\right)\mathbf{P}^*_{k+1}$
  5. Pressure predictor: Solve modified pressure equation, eq. (5.24) $\rightarrow p^*$
  6. Correct volumetric fluxes, eq. (5.6)
  7. Relax pressure field, $\rightarrow p^{k+1}$
  8. Momentum corrector: Solve velocity correction equation eq. (5.7) $\rightarrow \mathbf{u}^{k+1}_*$
- Finalize procedure

### 5.6.3 Results

All the methods above were implemented in the OpenFOAM solver developed in the previous sections. Some additional metrics were used to quantify the improvement these methods offered, as the visual differences in the contour plots were minimal. Thus, the RMSE of the velocity magnitude was defined as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_i^{\mathrm{N_{cells}}} |\mathbf{u}_{\mathrm{CFD}} - \mathbf{u}_{\mathrm{PIV}}|^2}{N_{\mathrm{cells}}}} \tag{5.25}$$

where $N_{\mathrm{cells}}$ is the number of the mesh cells. Additionally, the mean value of the normalized continuity residual field $\mu_{\mathrm{CRN}}$ and the mean value of the relative velocity error field $\mu_{\varepsilon_{r,\mathrm{Velocity}}}$ were used.

In tables 5.5 to 5.16 the performance of the different DA methods for all the studied cases is shown. Specifically, the velocity RMSE, the CRN mean values and standard deviation, and relative velocity error mean values and standard deviation are shown. Excluding the case of plane C under high shear conditions with a closed cube, shown in table 5.14, whose mesh resolution is extremely coarser than the other cases, the KF method prevails over the other DA methods achieving both lower mean values and lower variances in RMSE, CRN, and relative velocity errors. On the one hand, it manages to be closer to the PIV data compared to the other techniques (lower RMSE), and on the other hand, it does not lead to solutions that worsen the field's ability to conserve the mass. Additionally, as expected, the BLUE-nudging method performs better than the Identity-nudging methods, achieving lower mean values and standard deviations. Finally, the integrated prediction/correction scheme, even though it is an interesting concept, does not improve the classic KF formulation, and it also does offer significant improvement compared to the BLUE-nudging method.

**Notes on Computational Cost** Another aspect of the performance measurement of the different methods is the computational cost associated with them. It is obvious that the nudging-based methods are cheaper by design compared to the KF methods, as the former requires, at worst, three additional matrices stored in memory and one multiplication to be performed in each iteration. In contrast, the latter requires, at best, four additional matrices stored in memory, four multiplications, and one matrix inversion to be performed in each iteration. Due to the size of the meshes, the effect of the additional matrix storage, and the multiplications was insignificant and could not be identified while the simulations were running. Additionally, one would expect that the required matrix inversion would provide a big difference in code performance, but its effect was also not identifiable. This was expected, as the matrix that has to be inverted in this implementation is diagonal, and its inverse can be found in linear time ($O(n)$). The main performance hit came from the solution of the two linear systems using iterative methods, which scale at best as $O(n^{3/2})$, and had to be used for all the DA methods. Although, a performance scalability study with an increasing mesh size would provide better insight.

**Table 5.5:** Data assimilation methods performance for A-HS-Open.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.5834 | **-0.9335** | 5.2101 | 17.7302 | 9.3202 |
| Nominal ×4 | 0.5899 | -0.4671 | 3.6474 | 18.3421 | 8.5859 |
| Nominal ×16 | 0.5932 | -0.2333 | 2.5802 | 18.6627 | 8.1573 |
| Identity nudging | 0.5857 | -0.9352 | 5.2237 | 17.8017 | 9.3561 |
| BLUE nudging | 0.5840 | -0.9352 | 5.2264 | 17.7632 | 9.3026 |
| Classic KF | **0.5699** | -0.9346 | 5.0512 | **17.4842** | 8.7847 |
| Integrated KF | 0.5836 | -0.9352 | 5.2231 | 17.7509 | 9.2954 |

**Table 5.6:** Data assimilation methods performance for A-HS-Closed.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.5854 | -0.7240 | 4.6071 | 17.9994 | 9.0746 |
| Nominal ×4 | 0.5891 | -0.3622 | 3.2285 | 18.4292 | 8.4739 |
| Nominal ×16 | 0.5917 | -0.1810 | 2.2797 | 18.6830 | 8.1253 |
| Identity nudging | 0.5873 | **-0.7236** | 4.6067 | 18.0345 | 9.1456 |
| BLUE nudging | 0.5848 | -0.7246 | 4.6114 | 17.9723 | 9.0853 |
| Classic KF | **0.5747** | -0.7237 | 4.5118 | **17.7976** | 8.6608 |
| Integrated KF | 0.5851 | -0.7241 | 4.6144 | 17.9890 | 9.0681 |

**Table 5.7:** Data assimilation methods performance for A-LS-Open.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.6476 | **-0.9794** | 5.4309 | 18.4289 | 9.2024 |
| Nominal ×4 | 0.6550 | -0.4900 | 3.8264 | 19.0621 | 8.4142 |
| Nominal ×16 | 0.6592 | -0.2448 | 2.7121 | 19.3879 | 7.9832 |
| Identity nudging | 0.6494 | -0.9817 | 5.4456 | 18.4950 | 9.2001 |
| BLUE nudging | 0.6477 | -0.9817 | 5.4453 | 18.4503 | 9.1704 |
| Classic KF | **0.6273** | -0.9811 | 5.2456 | **18.0127** | 8.5872 |
| Integrated KF | 0.6481 | -0.9817 | 5.44815 | 18.4616 | 9.1755 |

**Table 5.8:** Data assimilation methods performance for A-LS-Closed.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.6661 | -1.004774 | 5.4593 | 18.9450 | 9.6392 |
| Nominal ×4 | 0.6733 | -0.5027 | 3.8697 | 19.5634 | 8.8798 |
| Nominal ×16 | 0.6777 | -0.2513 | 2.7445 | 19.9096 | 8.4394 |
| Identity nudging | 0.6677 | -1.004772 | 5.4563 | 18.9813 | 9.6754 |
| BLUE nudging | 0.6661 | -1.004774 | 5.4594 | 18.9450 | 9.6401 |
| Classic KF | **0.6456** | -1.004782 | 5.2584 | **18.4836** | 9.0953 |
| Integrated KF | 0.6657 | **-1.004772** | 5.4568 | 18.9331 | 9.6340 |

**Table 5.9:** Data assimilation methods performance for B-HS-Open.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.19685 | -0.171594658 | 0.9399 | 4.0143 | 2.4842 |
| Nominal ×4 | 0.20548 | -0.08579542 | 0.6676 | 4.1740 | 2.6193 |
| Nominal ×16 | 0.21584 | -0.04289772 | 0.4798 | 4.2861 | 2.9022 |
| Identity nudging | 0.19742 | **-0.171594650** | 0.9406 | 4.0219 | 2.4983 |
| BLUE nudging | 0.19685 | -0.171594658 | 0.9399 | 4.0145 | 2.4842 |
| Classic KF | **0.18963** | -0.171594705 | 0.9363 | **3.7804** | 2.5281 |
| Integrated KF | 0.19681 | -0.171594657 | 0.9397 | 4.0134 | 2.4843 |

**Table 5.10:** Data assimilation methods performance for B-HS-Closed.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.2092 | -0.17751270 | 0.9780 | 4.2638 | 2.7035 |
| Nominal ×4 | 0.2270 | -0.08875377 | 0.6950 | 4.5381 | 3.0706 |
| Nominal ×16 | 0.2361 | -0.04437670 | 0.5011 | 4.6573 | 3.2852 |
| Identity nudging | 0.2098 | **-0.17751269** | 0.9788 | 4.2723 | 2.7197 |
| BLUE nudging | 0.2092 | -0.17751328 | 0.9780 | 4.2639 | 2.7035 |
| Classic KF | **0.2016** | -0.17751331 | 0.9755 | **4.0167** | 2.7456 |
| Integrated KF | 0.2091 | -0.17751328 | 0.9777 | 4.2628 | 2.7036 |

**Table 5.11:** Data assimilation methods performance for B-LS-Open.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.2920 | -0.238829242 | 1.6135 | 6.3189 | 2.8798 |
| Nominal ×4 | 0.3749 | -0.119420974 | 1.2791 | 7.4141 | 4.9552 |
| Nominal ×16 | 0.4071 | -0.059710476 | 0.8228 | 7.5152 | 6.1038 |
| Identity nudging | 0.2939 | -0.238829252 | 1.6166 | 6.3517 | 2.9178 |
| BLUE nudging | 0.2920 | **-0.238829242** | 1.6135 | 6.3193 | 2.8800 |
| Classic KF | **0.2686** | -0.238829284 | 1.5743 | **5.8135** | 2.6468 |
| Integrated KF | 0.2919 | -0.238829245 | 1.6123 | 6.3179 | 2.8792 |

**Table 5.12:** Data assimilation methods performance for B-LS-Closed.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.3360 | -0.254508085 | 1.7666 | 7.0147 | 3.8468 |
| Nominal ×4 | 0.4073 | -0.127264048 | 1.2837 | 7.6976 | 5.9001 |
| Nominal ×16 | 0.4273 | -0.063631992 | 0.8128 | 7.5508 | 6.8207 |
| Identity nudging | 0.3401 | -0.254508087 | 1.7945 | 7.0772 | 3.9329 |
| BLUE nudging | 0.3358 | -0.254508090 | 1.7509 | 7.0155 | 3.8343 |
| Classic KF | **0.2998** | -0.254508088 | 1.6762 | **6.2904** | 3.3724 |
| Integrated KF | 0.3380 | **-0.254508078** | 2.0429 | 7.0336 | 3.9090 |

**Table 5.13:** Data assimilation methods performance for C-HS-Open.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.6676 | 0.27584648 | 3.1189 | 26.8163 | 23.7149 |
| Nominal ×4 | 0.7173 | 0.06896894 | 1.4684 | 29.3295 | 24.8841 |
| Nominal ×16 | 0.7113 | 0.06896840 | 1.4001 | 28.9615 | 24.8234 |
| Identity nudging | 0.6908 | **0.27584636** | 3.2071 | 27.6417 | 24.6547 |
| BLUE nudging | 0.6682 | 0.27584692 | 3.5327 | 26.8380 | 23.7357 |
| Classic KF | **0.6373** | 0.27584659 | 3.2808 | **25.8077** | 22.3989 |
| Integrated KF | 0.6666 | 0.27584646 | 3.1225 | 26.7909 | 23.6583 |

**Table 5.14:** Data assimilation methods performance for C-HS-Closed.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | **0.4781** | 1.500992 | 10.2946 | **19.7739** | 18.1795 |
| Nominal ×4 | 0.6682 | 0.750845 | 8.4117 | 27.4648 | 25.6171 |
| Nominal ×16 | 0.5410 | 0.375419 | 5.3741 | 21.8773 | 21.1213 |
| Identity nudging | 0.4907 | **1.500991** | 11.2002 | 20.1055 | 18.8686 |
| BLUE nudging | 0.4882 | 1.500995 | 12.6456 | 19.9484 | 18.8267 |
| Classic KF | 0.6008 | 1.500994 | 12.8695 | 24.3878 | 23.3404 |
| Integrated KF | 0.5246 | 1.500993 | 12.1243 | 21.4679 | 20.1956 |

**Table 5.15:** Data assimilation methods performance for C-LS-Open.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.7490 | 0.3767411 | 4.8536 | 30.5008 | 23.7523 |
| Nominal ×4 | 0.7797 | 0.1883908 | 3.0929 | 31.7665 | 24.7085 |
| Nominal ×16 | 0.8416 | 0.0941959 | 2.0636 | 34.6039 | 26.2627 |
| Identity nudging | 0.7740 | 0.3767409 | 4.5331 | 31.3198 | 24.7948 |
| BLUE nudging | 0.7489 | **0.3767406** | 4.2466 | 30.5022 | 23.7456 |
| Classic-KF | **0.7220** | 0.3767408 | 4.3423 | **29.4681** | 22.8117 |
| Integrated-KF | 0.7487 | 0.3767413 | 4.6856 | 30.4956 | 23.7347 |

**Table 5.16:** Data assimilation methods performance for C-LS-Closed.

| Method | RMSE $\left(\frac{\text{m}}{\text{s}}\right)$ | $\mu_{\text{CRN}}(\%)$ | $\sigma_{\text{CRN}}(\%)$ | $\mu_{\varepsilon_{r,\text{Velocity}}}(\%)$ | $\sigma_{\varepsilon_{r,\text{Velocity}}}(\%)$ |
|---|---|---|---|---|---|
| Nominal | 0.8088 | 0.854170 | 8.2395 | 33.0376 | 24.8619 |
| Nominal ×4 | 0.9219 | 0.427125 | 5.7140 | 37.9191 | 27.9899 |
| Nominal ×16 | 0.8757 | 0.213562 | 3.8671 | 35.8245 | 26.8513 |
| Identity nudging | 0.8778 | 0.854170 | 8.5933 | 35.6271 | 27.2836 |
| BLUE nudging | 0.8391 | 0.854170 | 8.3798 | 34.1326 | 25.9735 |
| Classic-KF | **0.7983** | **0.854168** | 7.8502 | **32.5508** | 24.6091 |
| Integrated-KF | 0.8273 | 0.854169 | 8.0949 | 33.6525 | 25.6099 |

Finally, the results for plane A under High Shear conditions and a closed cube are presented in figs. 5.67 to 5.75 for completeness.

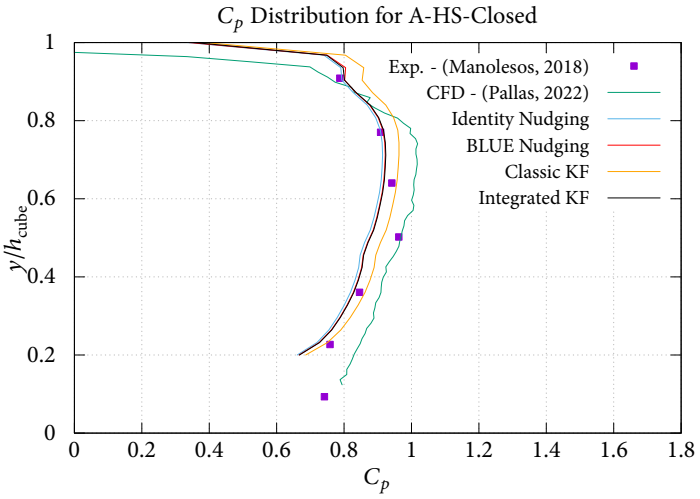
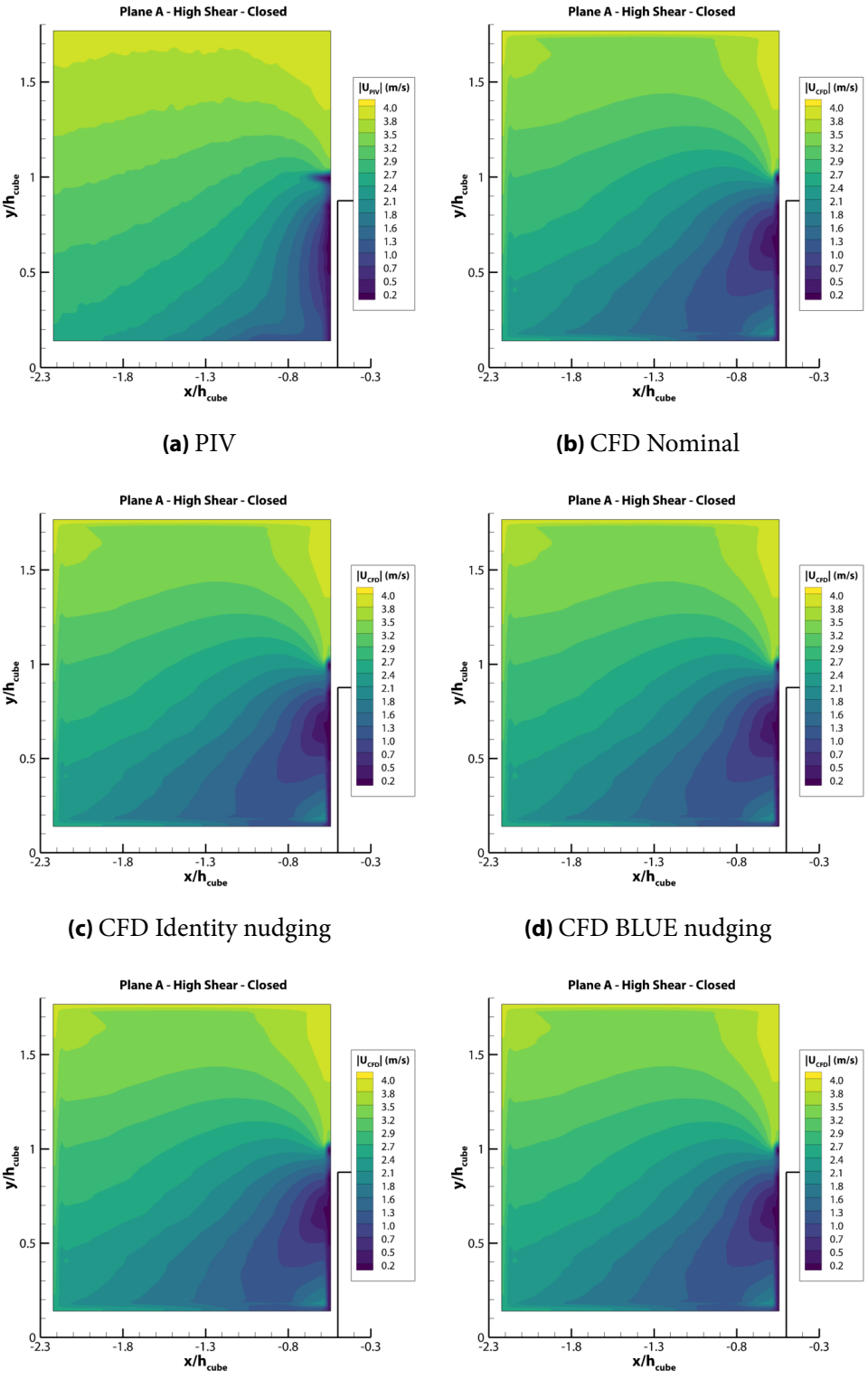



**Figure 5.67:** Pressure coefficient distribution along the cube for A-HS-Closed.




**(a)** PIV

**(b)** CFD Nominal

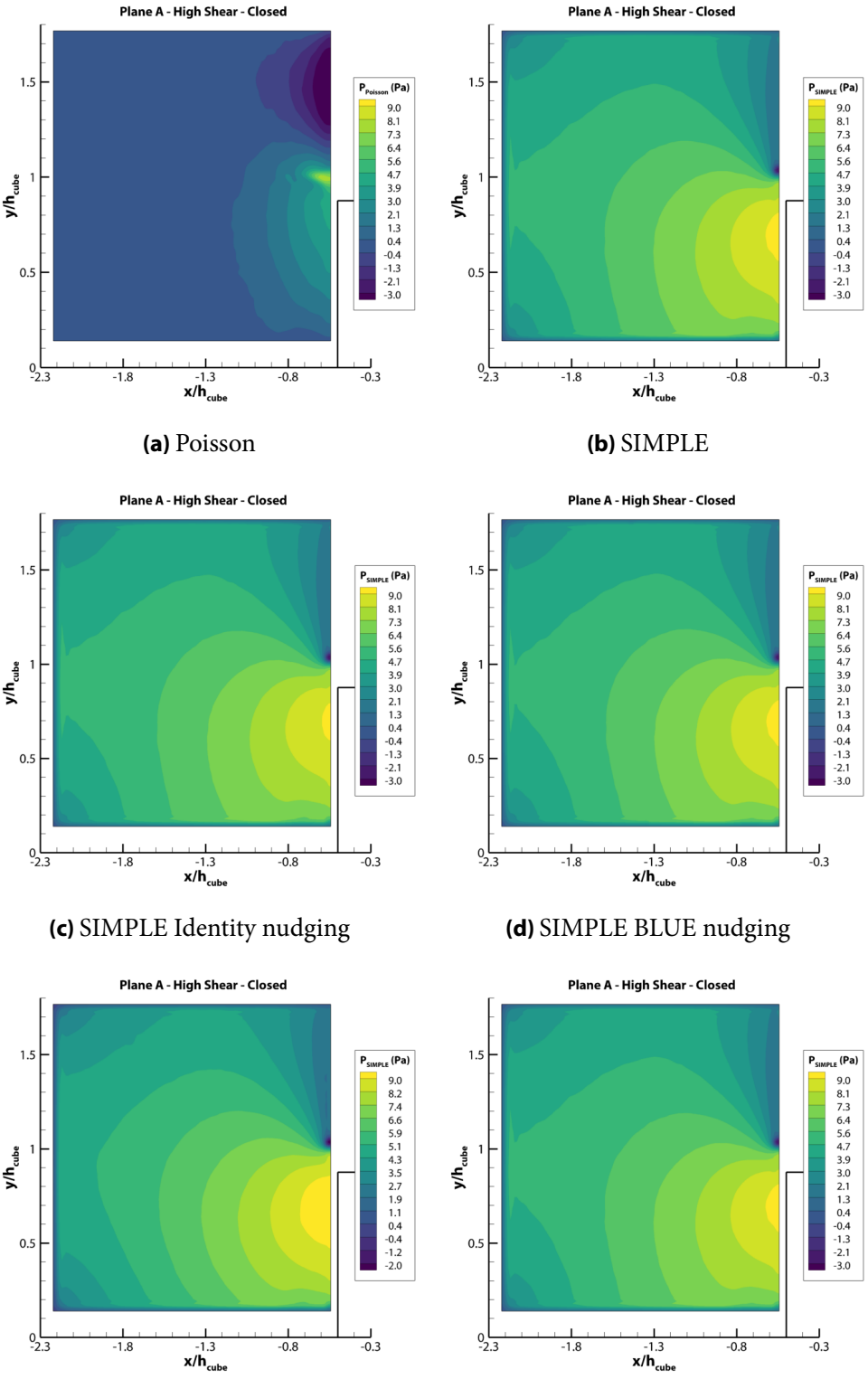
**(c)** CFD Identity nudging

**(d)** CFD BLUE nudging

**(e)** CFD Classic KF

**(f)** CFD Integrated KF

**Figure 5.68:** Velocity magnitude contours for A-HS-Closed.

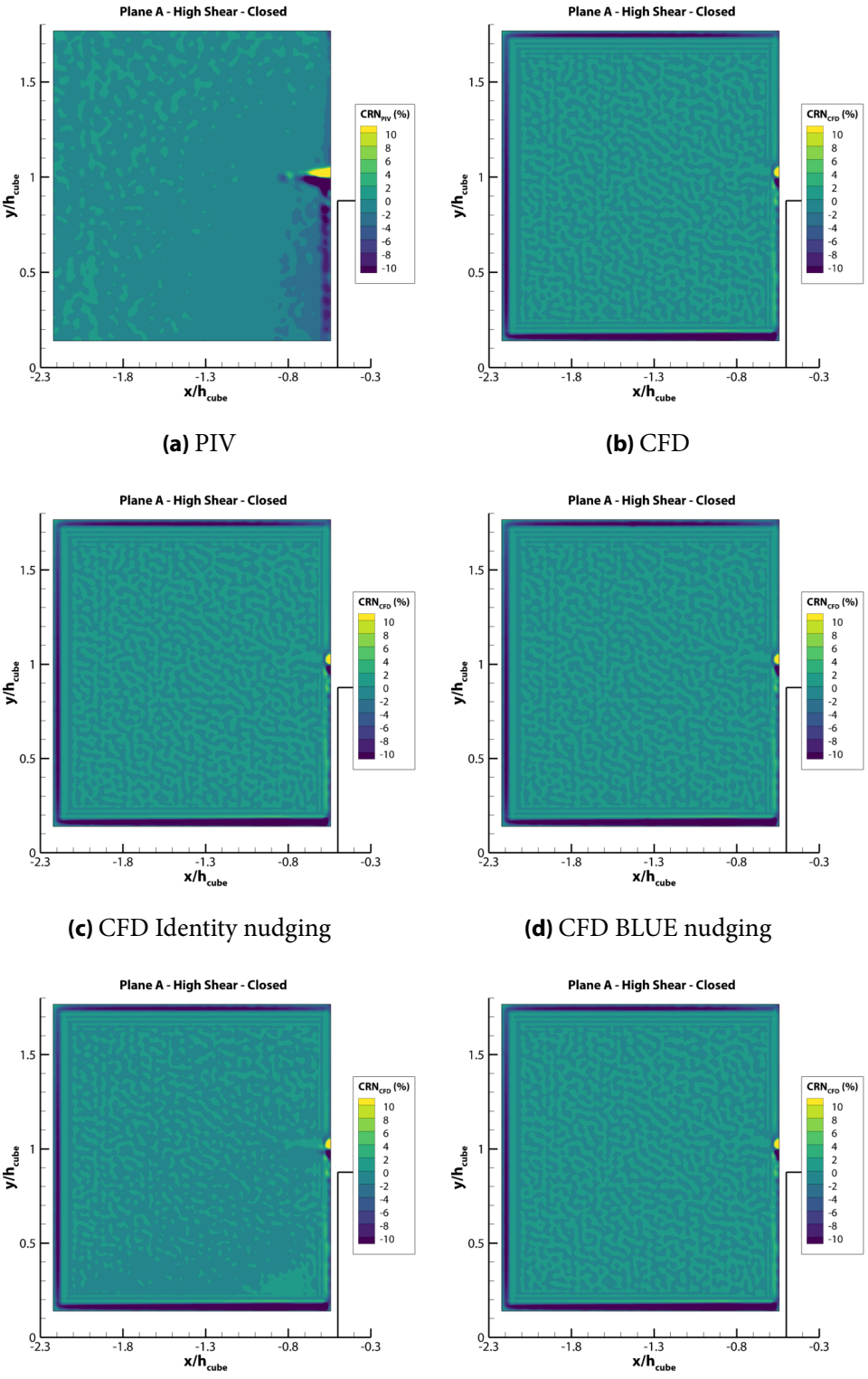


**(a)** Poisson **(b)** SIMPLE

**(c)** SIMPLE Identity nudging **(d)** SIMPLE BLUE nudging

**(e)** SIMPLE Classic KF **(f)** SIMPLE Integrated KF

**Figure 5.69:** Pressure contours for A-HS-Closed.




**(a)** PIV

**(b)** CFD

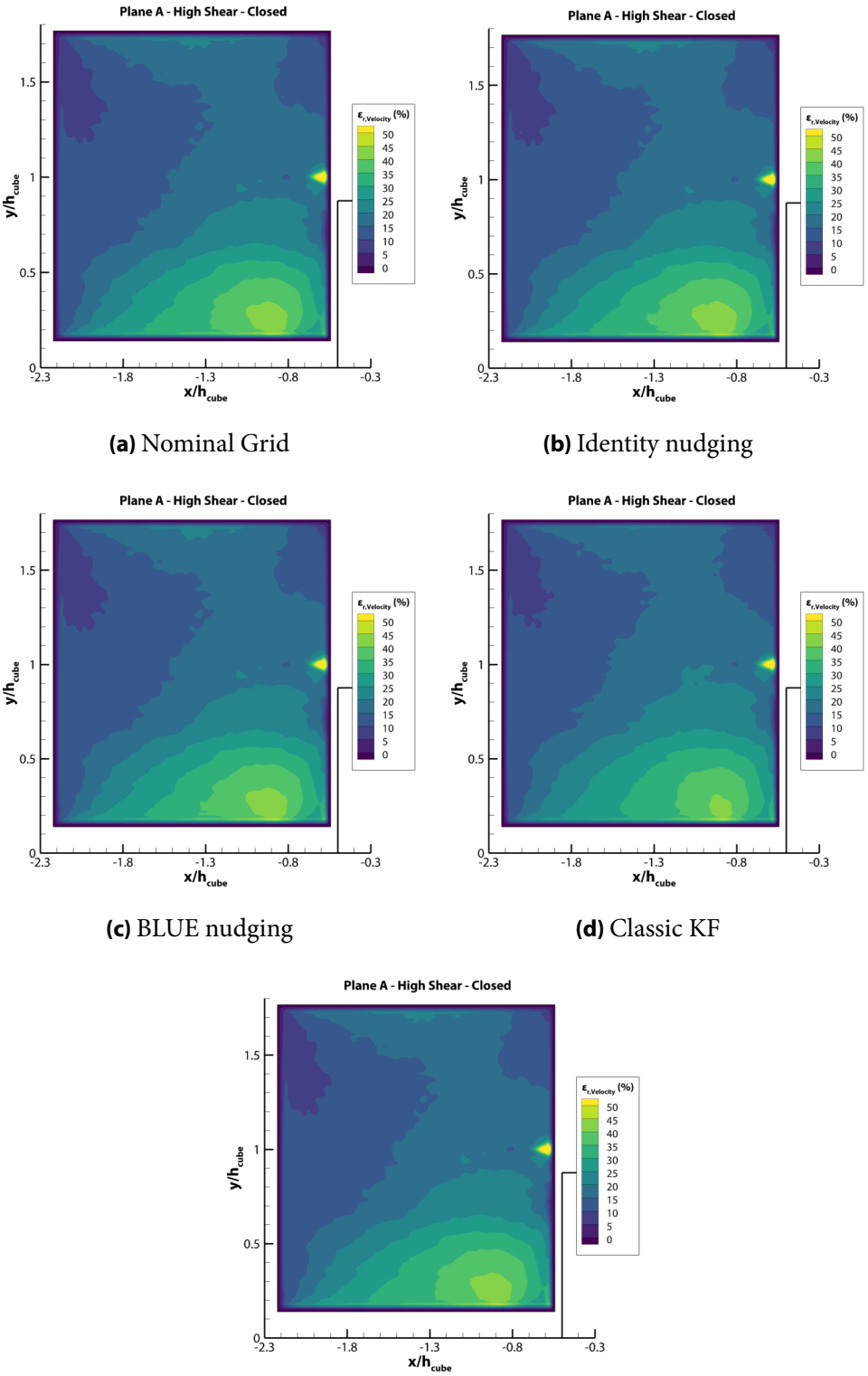
**(c)** CFD Identity nudging

**(d)** CFD BLUE nudging

**(e)** CFD Classic KF

**(f)** CFD Integrated KF

**Figure 5.70:** Normalized continuity residual contours for A-HS-Closed.




**(a)** Nominal Grid

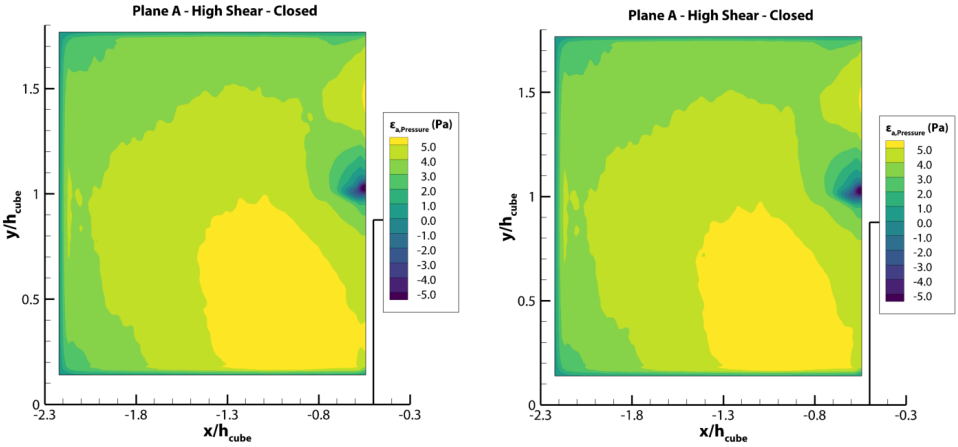
**(b)** Identity nudging

**(c)** BLUE nudging

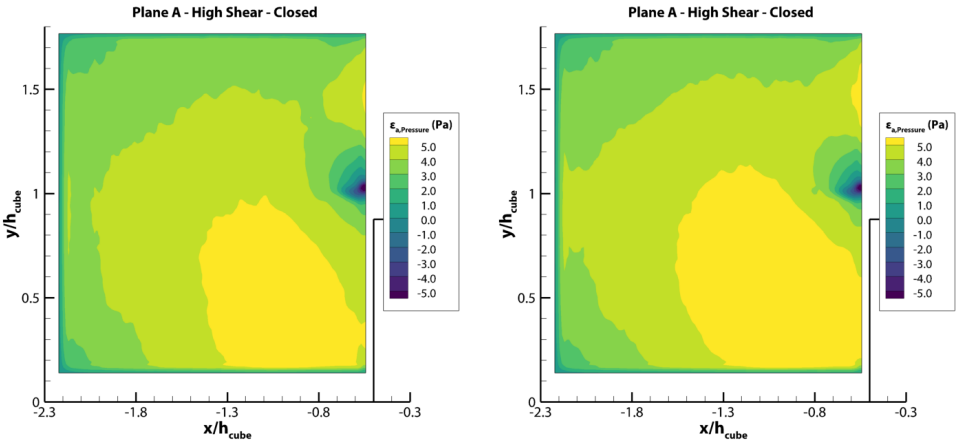
**(d)** Classic KF

**(e)** Integrated KF

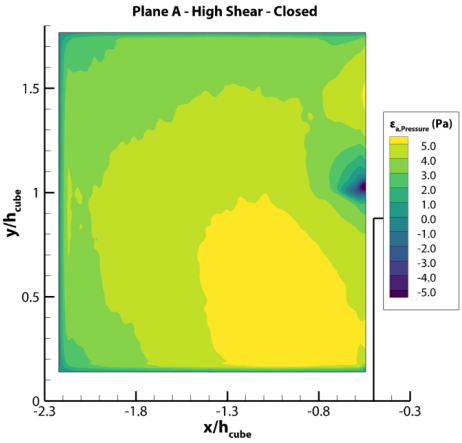
**Figure 5.71:** Relative velocity error contours for A-HS-Closed.




**(a)** Nominal Grid **(b)** Identity nudging

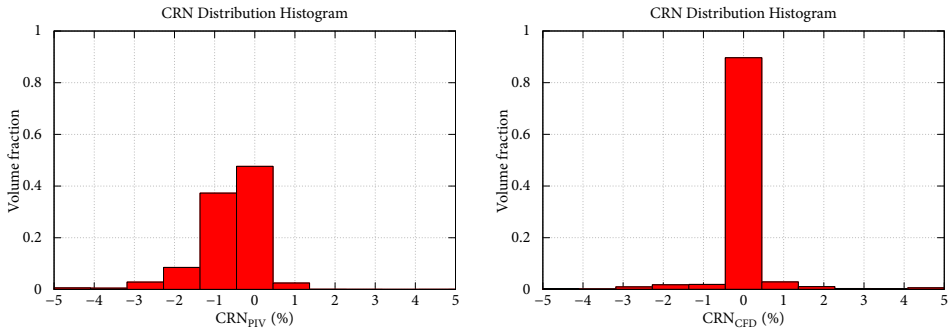



**(c)** BLUE nudging **(d)** Classic KF




**(e)** Integrated KF

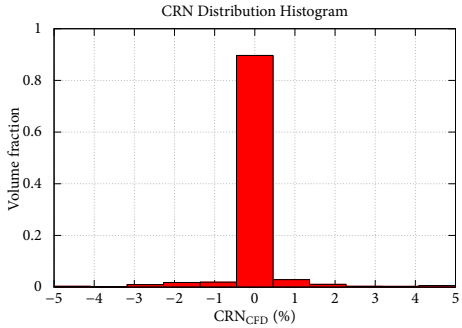
**Figure 5.72:** Absolute pressure error contours for A-HS-Closed.



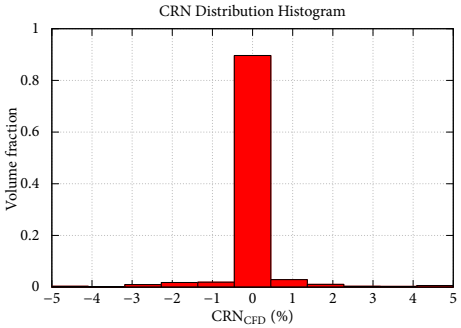

**(a)** PIV



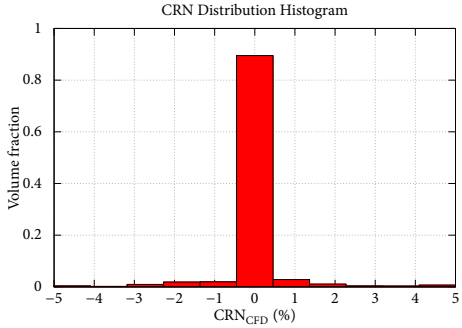

**(b)** CFD



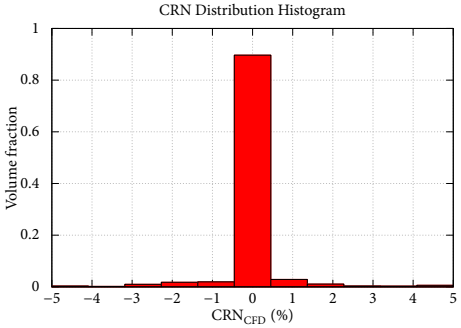

**(c)** Identity nudging




**(d)** BLUE nudging

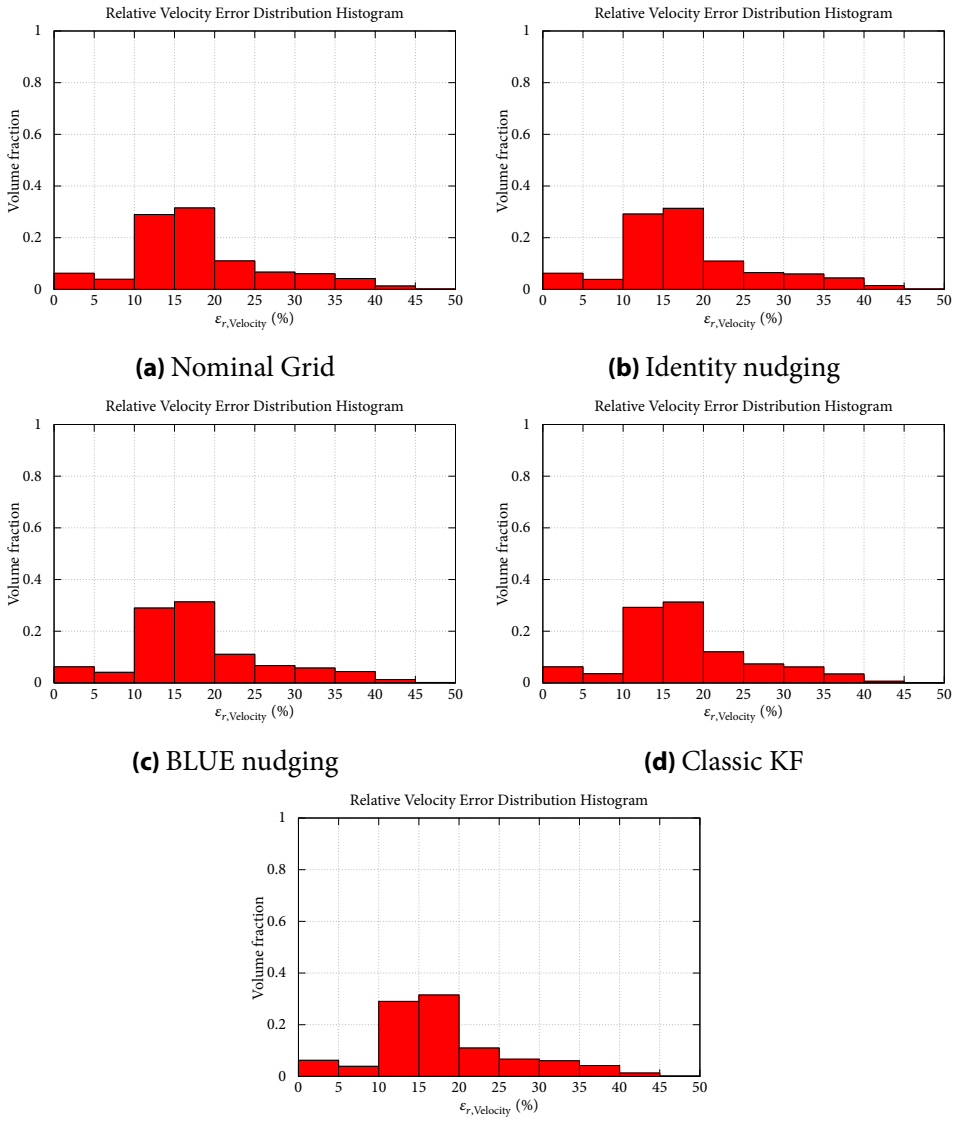



**(e)** Classic KF




**(f)** Integrated KF

**Figure 5.73:** Distribution of normalized continuity residual field for A-HS-Closed.




**(a)** Nominal Grid

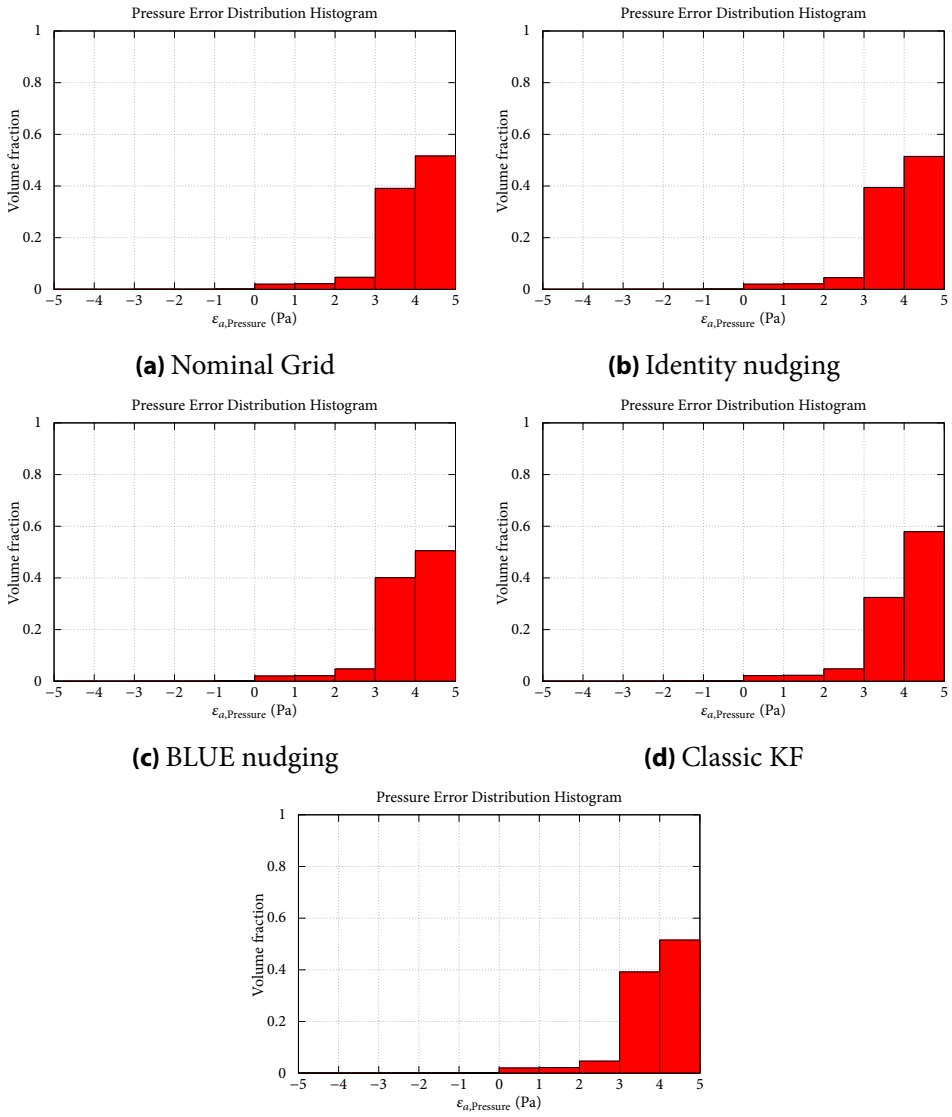
**(b)** Identity nudging

**(c)** BLUE nudging

**(d)** Classic KF

**(e)** Integrated KF

**Figure 5.74:** Distribution of relative velocity error field for A-HS-Closed.




**(a)** Nominal Grid

**(b)** Identity nudging

**(c)** BLUE nudging

**(d)** Classic KF

**(e)** Integrated KF

**Figure 5.75:** Distribution of absolute pressure error field for A-HS-Closed.

# Chapter 6

# Summary and Conclusions

## 6.1 Summary

In this Diploma Thesis, the application of Data Assimilation methods in Computational Fluid Dynamics applications was investigated.

Firstly, a Fortran 95 computational code using the Finite Volume Method was developed in order to solve the transient 1-D compressible Euler equations. The code was used to solve the Sod Shock Tube case for different discretization schemes and varying grid resolution and compare the numerical results with the analytical solution.

Then, the computational code was extended by implementing a nudging Data Assimilation method. Two approaches were used; the first one nudged the numerical solution to the analytical solution, while the second one nudged the numerical solution to a numerical solution obtained from a finer grid. Afterward, the optimal nudging method was implemented, determining the gain matrix that minimizes the computational error through an assimilation time window. Finally, the adjoint equations to the transient 1-D Euler equations were devised and solved numerically to calculate the sensitivity derivatives used in the optimization loop.

Next, the calculation of pressure fields using PIV velocity fields was studied. A solver using the SIMPLE algorithm to correct the velocity fields and calculate the pressure fields was implemented in the OpenFOAM CFD toolbox. Additionally, a Poisson equation was used to calculate the pressure directly from the PIV velocity fields and compare the results to the ones produced from the SIMPLE algorithm. Then, the solver was verified in the atmospheric boundary layer flow around a wall-mounted cube. This flow was experimentally studied in previous work at the Laboratory for Innovative Environmental Technologies at the School of Mechanical Engineering at the National Technical University of Athens and the PIV data were available.

Finally, after the code validation and mesh refinement study, four DA methods, two based on nudging and two based on the KF, were implemented in the OpenFOAM-based solver and compared to each other.

## 6.2 Conclusions

By completing the studies conducted in this Diploma Thesis, the following conclusions are drawn:

- Application 1: 1-D Shock Tube Problem
  - The numerical solution could not approximate the analytical solution sufficiently.

  - Using the nudging method with the analytical solution improved the numerical solution.
  - Using the nudging method with a coarse and fine grid improved the coarse grid solution significantly, leading to results that are better than the ones obtained from grids with a resolution bigger around an order of magnitude.
  - Optimizing the gain matrix provided great value to the improvement of the nudging method, but a balance between computational cost and accuracy improvement must be reached.

- Application 2: Pressure Calculation from Stereo-PIV Data
  - The developed OpenFOAM-based solver using the SIMPLE algorithm managed to calculate pressure fields from PIV velocity fields
  - The results were independent of the mesh size, according to the mesh refinement study that was performed.
  - The use of DA methods, in general, improved the results, with the most improvement coming from the use of the classic KF method.

## 6.3 Future work

Based on the conclusion reached above and the relative discussion in the application chapters in this Thesis, the following future works are proposed:

- Application 1: 1-D Shock Tube Problem
  1. The nudging method can be turned into a Kalman Filtering method that uses the numerical solution in a finer gird as ground truth, under the assumption that the prediction errors are proportional to the time and space discretizations.
  2. A more robust and more fast optimization method can be used to determine the optimal gain matrix instead of the steepest descent method. For example, the steepest descent method with line search or the more advanced conjugate gradient method can be used.
  3. An investigation can be performed to determine the effect of the duration of the assimilation window as a fraction of the total simulation time, as well as the assimilation window opening, compared to the total simulation time. This can lead to a simple heuristic to minimize the computational cost for the optimization loop but still produce an optimized gain matrix.

- Application 2: Pressure Calculation from Stereo-PIV Data
  1. The developed OpenFOAM-based can be used to simulate the flow around the cube in a bigger computational domain using RANS. Initially, the results from the simulations on the large computational domain can be used to verify the the method used in the Thesis. Then, Data Assimilation methods can be used to take into account the experimental data, improving the CFD results near the cube, especially the cube wake. An investigation can be performed regarding the impact of the plane position on the improvement.

2. In addition, the effect of turbulence modeling can be investigated using different formulations for the data assimilation problem. For instance, data assimilation methods can be used only as a forcing term on the momentum equations or can also be used to augment the equations used by the turbulence models, as the Reynolds stresses are available from the PIV measurements.

# List of Figures

# List of Tables

# Εκτεταμένη Ελληνική Περίληψη

## 1 Εισαγωγή

Η Αφομοίωση Δεδομένων (ΑΔ) είναι μια περιοχή των μαθηματικών που έχει στόχο τον βέλτιστο συνδυασμό των αριθμητικών μοντέλων και των πειραματικών δεδομένων, με σκοπό την βελτίωση των αποτελεσματων των πρώτων, λαμβάνοντας υπόψιν τις αβεβαιότητες και των δύο.

Υπάρχουν δύο βασικές προσεγγίσεις της ΑΔ, κάνοντας χρήση μεθόδων λογισμού των μεταβολών και κάνοντας χρήση στατιστικών μεθόδων. Και στις δυό προσεγγίσεις στόχος είναι η εύρεση μιας βέλτιστη λύσης, η οποία είτε ελαχιστοποιεί μια κατάλληλη συνάρτηση κόστους (λογισμός των μεταβολών), είτε ελαχιστοποιεί την συνδιακύμανση των σφαλμάτων (στατιστική προσέγγιση). Αν και σε ορισμένες ειδικές περιπτώσεις οι δύο προσεγγίσεις είναι ταυτόσημες και καταλήγουν στην ίδια λύση, η στατιστική προσέγγιση μπορεί να προσφέρει μια πιο εμπλουτισμένη δομή πληροφορίας, η οποία περιέχει μια μέση τιμή και κάποια χαρακτηριστικά της μεταβλητότητας της. Οι προσεγγίσεις της ΑΔ έχουν χρησιμοποιηθεί εκτενώς στα πεδία της γεωφυσικής, της μετεωρολογίας, της ωκεανογραφίας κ.α., ενώ έχει αναπτυχθεί πληθώρα τεχνικών.

Μπορεί να ειπωθεί ότι η Πειραματική Ρευστοδυναμική (ΠΡ) και η Υπολογιστική Ρευστοδυναμική (ΥΡ) αναπτύσσονται ανεξάρτητα για μεγάλο χρονικό διάστημα. Κάποιες διαδραστικές προσπάθειες έχουν γίνει, για παράδειγμα, ο σχεδιασμός πειραμάτων με τη βοήθεια της ΥΡ, η διόρθωση πειραματικών αποτελεσμάτων με βάση αποτελέσματα ΥΡ ή επικύρωση/βαθμονόμιση μοντέλων ΥΡ μέσω της ΠΡ. Ωστόσο, τα τελευταία χρόνια έχουν γίνει περισσότερες προσπάθειες για την σύζευξη ΠΡ και ΥΡ, με στόχο την αμοιβαία βελτίωση τους.

Οι μετρήσεις και οι μετρητικές συσκευές μπορούν να χαρακτηρίσουν ροές σε πραγματικές συνθήκες, ωστόσο πολλές φορές είναι ανεπαρκείς λόγω των περιορισμών τους στο χρόνο και το χώρο, την έλλειψη ακρίβειας και το γεγονός ότι μερικές φορές ένα μετρητικό όργανο μπορεί να διαταράξει τη ροή. Σε αντίθεση με τις κλασσικές προσεγγίσεις ΠΡ, οι οποίες βασίζονται σε μετρητικές συσκευές υψηλής ακρίβειας και σχήματα υψηλής τάξης για τη μετεπεξεργασία των αποτελεσμάτων, ώστε να ξεπεραστούν αυτά τα προβλήματα, στην ΑΔ τα αριθμητικά μοντέλα που περιγράφουν τη ροή επαυξάνονται με τα πειραματικά δεδομένα. Σε σύγκριση με τις συμβατικές προσεγγίσεις ΥΡ, οι αβεβαιότητες μειώνονται κατά τη χρήση ΑΔ κάνοντας χρήση απευθείας των πειραματικών δεδομένων, αντί για την διερεύνηση της πληρότητας της μοντελοποίησης και των αρχικών ή/και οριακών συνθηκών.

Στόχος της παρούσας Διπλωματικής Εργασίας είναι η εφαρμογή τεχνικών ΑΔ σε προβλήματα ΥΡ. Μελετήθηκαν και επικυρώθηκαν δύο εφαρμογές, κάνοντας χρήση διαφορετικών τεχνικών ΑΔ και ενσωματώνοντας συνθετικά δεδομένα διαθέσιμα από την αναλυτική λύση του προβλήματος όπου αυτή υπάρχει, καθώς και πειραματικά δεδομένα.

# 2 Υπολογιστική Ρευστοδυναμική

Αρχικά, γίνεται μια σύντομη εισαγωγή στις μεθόδους ΥΡ που χρησιμοποιήθηκαν στην παρούσα Εργασία. Αναλυτικότερα, περιγράφονται οι εξισώσεις που διέπουν τη ροή (Euler και Navier-Stokes) καθώς και η διακριτοποίηση τους μέσω της μεθόδου των πεπερασμένων όγκων.

Η μέθοδος των πεπερασμένων όγκων [22] αποτελεί μια μέθοδο διακριτοποίησης μ.δ.ε., η οποία εφαρμόζει την ιδέα των όγκων ελέγχου για την περιγραφή φυσικών συστημάτων. Το υπολογιστικό χωρίο διακριτοποιείται σε πολλούς μικρούς όγκους πεπερασμένης διάστασης (κελιά) και οι μ.δ.ε. που περιγράφουν το σύστημα ολοκληρώνονται κατ'όγκο κάνοντας χρήση του θεωρήματος της απόκλισης για την μετατροπή ολοκληρωμάτων κατ'όγκο σε επιφανειακά ολοκληρώματα. Έτσι, οι όροι υπολογίζονται στις περιβάλλουσες επιφάνειες των κελιών, οδηγώντας στον υπολογισμό των διανυσματικών ροών διαμέσου αυτών των επιφανειών. Επειδή η διανυσματική ροή που εισέρχεται σε ένα κελί είναι ίδια με αυτή που εξέρχεται από τα γειτονικά του, η μέθοδος των πεπερασμένων όγκων είναι συντηρητική και μπορεί να εφαρμοστεί με ευκολία σε δομημένα αλλά και σε μη-δομημένα πλέγματα.

## 2.1 Εξισώσεις Euler

Οι εξισώσεις Euler αποτελούν ένα σύστημα ημι-γραμμικών υπερβολικών μερικών διαφορικών εξισώσεων (μ.δ.ε) και περιγράφουν την αδιαβατική και ατριβή ροή ρευστών [20, 21]. Πιο συγκεκριμένα, αποτελούν έκφραση της διατήρησης της μάζας, της ορμής και της ενέργειας κατά τη ροή ενός ρευστού στο χώρο. Η συντηρητική μορφή των εξισώσεων Euler για μη-μόνιμη συμπιεστή ροή παρουσιάζεται στην εξ. (2.1) και (2.3). Όπου **U** είναι το διάνυσμα των συντηρητικών μεταβλητών και **F** το διάνυσμα των αντίστοιχων διανυσματικών ροών τους. Ενώ, $\rho$ είναι η πυκνότητα του ρευστού, $u$ η ταχύτητα του ρευστού και $p$ η στατική πίεση του ρευστου. Στην παρούσα εργασία έγινε χρήση της συντηρητικής μορφής των εξισώσεων Euler, καθώς στη συμπιεστή ροή, όπου είναι πιθανή η εμφάνιση κυμάτων κρούσης και ασυνεχειών, μόνο οι συντηρητικές μεταβλητές είναι κατάλληλες για τον υπολογισμό των αριθμητικών ροών.

Ο πεπερασμένος όγκος ενός 1Δ ομοιόμορφου πλέγματος παρουσιάζεται στο σχ. 2.1. Περιγράφεται ο όγκος που περιβάλει το $i$ – οστο κελί και οι κατακόρυφες γραμμές αναπαριστούν τις επιφάνειες που περιβάλλουν τον πεπερασμένο όγκο, $(i \pm 1/2)$. Τέλος, οι δείκτες $L$ και $R$ αναφέρονται στις διαφορετικές πλευρές των κοινών επιφανειών των κελιών, όπου επιλύονται τα προβλήματα Riemann.

Η εξ. (2.1) ολοκληρώνεται κατά μήκος του πεπερασμένου όγκου, όπως περιγράφεται στο σχ. 2.1. Προσεγγίζοντας την μέση τιμή στο $i$ – οστο κελί τη $n$ – οστη χρονική στιγμή και συνδυάζοντας τις εξ. (2.17) και (2.18), προκύπτει το σχήμα Godunov $1^{ης}$ τάξης [21, 23] όπως περιγράφεται στην εξ. (2.19).

### 2.1.1 Μέθοδος του Roe

Λόγω της διακριτοποίησης του πεδίου, εμφανίζονται τοπικές ασυνέχειες μεταξύ γειτονικών κελιών. Ένα πρόβλημα αρχικών τιμών, το οποίο περιγράφεται από συντηρητικές εξισώσεις και έχει αρχικά σταθερές συνθήκες με μια τοπική ασυνέχεια, ονομάζεται πρόβλημα Riemann. Για τον υπολογισμό των διανυσματικών ροών διαμέσου των κοινών επιφανειών των κελιών πρέπει να επιλυθεί το πρόβλημα Riemann σε κάθε κοινή επιφάνεια. Καθώς το πρόβλημα Riemann είναι μη-γραμμικό, γίνεται χρήση επαναληπτικών επιλυτών με υψηλό υπολογιστικό κόστος [21, 23]. Για τη μείωση του υπολογιστικού κόστους γίνεται χρήση προσεγγιστικών επιλυτών του προβλήματος Riemann, πιο συγκεκριμένα της μεθόδου του Roe [24].

Στη μέθοδο του Roe, γίνεται χρήση μια προσεγγιστικής γραμμικοποιημένης Ιακωβιανής του συστήματος $\tilde{\mathbf{A}}$, η οποία είναι διαγωνοποιήσιμη και περιγράφεται στην εξ. (2.27). Για τον υπολογισμό του μητρώου $\tilde{\mathbf{A}}$ γίνεται χρήση των αριστερών και δεξιών ιδιοδιανυσμάτων της πραγματικής Ιακωβιανής του συστήματος [21], υπολογισμένα για τις σταθμισμένες κατά Roe μεταβλητές όπως αυτές περιγράφονται στις εξ. (2.28)–(2.31). Ενώ η διανυσματική ροή διαμέσου των κοινών επιφανειών των κελιών σύμφωνα με τη μέθοδο του Roe [24], δίνεται από την εξ. (2.34).

### 2.1.2 Σχήμα Ανακατασκευής των Μεταβλητών

Από την εξ. (2.34), προκύπτει η ανάγκη ενός σχήματος ανακατασκευής των μεταβλητών από τα κέντρα των κελιών όπου αποθηκεύονται στις κοινές τους επιφάνειες,για τον υπολογισμό των διανυσματικών ροών. Το πιο απλο σχήμα είναι το ανάντι σχήμα, το οποίο χρησιμοποιεί τις μεταβλητές των κέντρων των κελιών εκατέρωθεν της κοινής επιφάνειας ως έχουν, έχει ακρίβεια $1^{ης}$ τάξης και περιγράφεται στην εξ. (2.35).

Επιπλέον, μέσω τμηματικά συνεχών προσεγγίσεων για κάθε κελί και του αναπτύγματος Taylor στη γειτονιά του κέντρου του κάθε κελιού, μπορεί να διατυπωθεί ένα σχήμα $2^{ης}$ τάξης, όπως φαίνεται στην εξ. (2.36).

Τέλος, το σχήμα MUSCL [26] κάνει χρήση περιοριστών κλίσεων για την αποφυγή ταλαντώσεων στις περιοχές όπου εμφανίζονται ασυνέχειες ή κύματα κρούσης για το σχήμα $2^{ης}$ τάξης. Το σχήμα MUSCL περιγράφεται στην εξ. (2.37), ενώ ένας τυπικός περιοριστής κλίσεων, ο συμμετρικός περιοριστής του van-Leer [27], περιγράφεται στην εξ. (2.39), όπου $r_*$ είναι ο λόγος κλίσεων στις κοινές επιφάνειες των κελιών.

## 2.2 Εξισώσεις Navier-Stokes

Οι εξισώσεις Navier-Stokes (NS) είναι ένα σύστημα παραβολικών μ.δ.ε, το οποίο περιγράφει την ιξώδη ροή και περιγράφει τη διατήρηση μάζας και ορμής για τα Νευτώνεια ρευστά. Για την συμπιεστή ροή, προστίθεται και μια εξίσωση κατάστασης η οποία συνδέει την πίεση, την θερμοκρασία και την πυκνότητα του ρευστού. Οι εξισώσεις NS για μη-μόνιμη ασυμπίεστη ροή παρουσιάζονται στις εξ. (2.11) και (2.12), με $\tau_{ij}$ τον τανυστή των ιξωδών τάσεων, όπως περιγράφεται στην εξ. (2.13).

Έπειτα πραγματοποιείται μια σύντομη περιγραφή της διακριτοποίησης των εξισώσεων NS μέσω της μεθόδου των πεπερασμένων όγκων, με κεντρο-κυψελική διατύπωση και για ομόθετα πλέγματα. Στην κεντρο-κυψελική διατύπωση, όλες οι μεταβλητές κατάστασης (πίεση και ταχύτητα) αποθηκευονται στο κέντρο των κελιών.

Η χρήση της κεντρο-κυψελικής διατύπωσης μαζί με ομόθετα πλέγματα επιλέχθηκε καθώς είναι η πιο συνηθισμένη προσέγγιση για επιλύτες ασυμπίεστης ροής και έχει υλοποιηθεί στο λογισμικό ΥΡ ανοιχτού κώδικα OpenFOAM [28–30], το οποίο χρησιμοποιείται στη δεύτερη εφαρμογή της Εργασίας.

Ένας πεπερασμένος όγκος (κελί) ενός 2Δ ορθογωνικού πλέγματος φαίνεται στο σχ. 2.2. Το κέντρο του κελιού σημειώνεται με $P$, και οι υπόλοιποι δείκτες αναφέρονται στα γειτονικά κελιά με βάση τη σχετική τους θέση ως προς το $P$. Οι συνεχείς γραμμές αναπαριστούν τις επιφάνειες που περιβάλλουν τους πεπερασμένους όγκους, ενώ οι διακεκομμένες αναπαριστούν το πλέγμα πάνω στο οποίο βρίσκονται τα κέντρα των κελιών. Η εξίσωση διατήρησης της ορμής ολοκληρώνεται κατά τον πεπερασμένο όγκο $V_P$, όπως φαίνεται στην εξ. (2.43) για μόνιμη ροή και διανυσματική μορφή. Έτσι, τα ολοκληρώματα όγκου, μετατρέπονται σε επιφανειακά ολοκληρώματα μέσω του θεωρήματος της απόκλισης (θεώρημα Gauss) και μέσω σχημάτων ανακατασκευής από τα κέντρα

των κελιών στις επιφάνειες που τα περιβάλλουν, διατυπώνεται η τελική διακριτοποιημένη μορφή των εξισώσεων NS [28–32].

Η τελική μορφή των ολοκληρωμάτων των όρων πηγής, κλίσεων, μεταφοράς και διάχυσης παρουσιάζονται στις εξ. (2.44), (2.45), (2.47) και (2.49) αντίστοιχα. Ενώ τα σχήματα σχήματα ανακατασκευής των όρων κλίσεων, μεταφοράς και διάχυσης παρουσιάζονται στις εξ. (2.46), (2.48) και (2.50) αντίστοιχα για ορθογωνικά πλέγματα.

### 2.2.1 Αλγόριθμος SIMPLE

Μια ενδιαφέρουσα συμπεριφορά των εξισώσεων NS είναι ότι για δεδομένο πεδίο πίεσης, το πεδίο ταχύτητας που προκύπτει από την εξίσωση διατήρησης της ορμής, δεν ικανοποιεί απαραίτητα την εξίσωση διατήρηση της μάζας. Η πιο συνηθισμένη αντιμετώπιση αυτής της συμπεριφοράς είναι η χρήση επαναληπτικών σχημάτων πρόβλεψης-διόρθωσης, τα οποία οδηγούν σε πεδία πίεσης και ταχύτητας τα οποία ικανοποιούν τις εξισώσεις διατήρησης της μάζας και της ορμής. Στην παρούσα Εργασία, έγινε χρήση του αλγορίθμου SIMPLE [34].

Αρχικά, γίνεται μια αρχική υπόθεση για τα πεδία πίεσης και ταχύτητας $p^*$ και $\mathbf{u}^*$ αντίστοιχα. Το πεδίο ταχύτητας προκύπτει από την εξίσωση ορμής, η οποία παρουσιάζεται στην ημι-διακριτοποιημένη μορφή της στην εξ. (2.56). Έπειτα, υποθέτοντας ότι τα πεδία πίεσης και ταχύτητας αποτελούνται από ένα κομμάτι πρόβλεψης και ένα κομμάτι διόρθωσης, όπως φαίνεται στις εξ. (2.57) και (2.58), υποθέτοντας ότι οι μη διαγώνιοι όροι του πεδίου διόρθωσης ταχύτητας είναι αμελητεοι και αντικαθιστώντας την εξ. (2.62) στην εξίσωση διατήρησης της μάζας, προκύπτει η εξίσωση διόρθωσης της πίεσης εξ. (2.63). Ενώ η σχέση μεταξύ διόρθωσης πίεσης και ταχύτητας δίνεται από την εξ. (2.60).

Έτσι, επιλύοντας την εξ. (2.63) προκύπτει το πεδίο διόρθωσης πίεσης $p'$ και αντίστοιχα αυτό της διόρθωσης ταχύτητας. Έπειτα, διορθώνονται τα πεδία πίεσης και ταχύτητας και τους εφαρμόζεται υποχαλάρωση μέσω των εξ. (2.64) και (2.65), ενώ στη συνέχεια χρησιμοποιούνται ως αρχική εκτίμηση για την επόμενη επανάληψη. Επομένως, πραγματοποιείται μια επαναληπτική διαδικασία στο τέλος της οποία τα πεδία διόρθωσης θα είναι μηδενικά και τα τελικά πεδία πιεσης και ταχύτητας θα ικανοποιούν τις εξισώσεις διατήρησης μάζας κια ορμής.

## 2.3 Μοντελοποίηση της Τύρβης - RANS

Για την απευθείας επίλυση τυρβώδων ροών απαιτείται πολύ πυκνή χωρική και χρονική διακριτοποίηση για την επίλυση των εξισώσεων NS, ώστε να προσδιοριστούν οι στοχαστικές μεταβολές που παρατηρούνται στα πεδία πίεσης και ταχύτητας και χαρακτηρίζουν το φαινόμενο της τύρβης. Για τη μείωση της απαιτούμενης υπολογιστικής ισχύος, γίνεται μοντελοποίηση της επίδρασης της τύρβης. Στην παρούσα Εργασία, έγινε χρήση της μεθόδου RANS, δηλαδή της στάθμισης κατά Reynolds των εξισώσεων NS για μόνιμη ροή. Στην RANS τα πεδία αποσυνθέτονται σε μια σταθερή συνιστώσα (μέσο πεδίο) και σε μια μεταβαλλόμενη συνιστώσα με μηδενική μέση τιμή. Έτσι, αντικαθιστώντας αυτή την υπόθεση στις εξισώσεις NS, προκύπτουν οι εξισώσεις για το μέσο πεδίο, όπως φαίνεται στις εξ. (2.71) και (2.72), με $\bar{p}$ και $\bar{\mathbf{u}}$ τα μέσα πεδία πίεσης και ταχύτητας αντίστοιχα. Οι RANS εξισώσεις για το μέσο πεδίο διαφέρουν από τις εξισώσεις NS μόνο κατά τον όρο $-\rho\overline{\mathbf{u}'\mathbf{u}'}$, που ονομάζεται τανυστής τάσεων Reynolds και αποτελεί τη μέση τιμή του γινομένου των διαταραχών. Συνήθως, ο όρος των τάσεων Reynolds μοντελοποιείται, ωστόσο στην παρούσα Εργασία, λόγω της ύπαρξης πειραματικών δεδομένων των τάσεων Reynolds για την εφαρμογή όπου η ροή είναι τυρβώδης, μέσω προηγούμενων εργασιών στο Εργαστήριο Τεχνολογικών Καινοτομιών Προστασίας Περιβάλλοντος της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ [41–44], δεν έγινε χρήση κάποιου μοντέλου τύρβης για την επίλυση των RANS.

# 3 Αφομοίωση Δεδομένων

Έπειτα, γίνεται μια σύντομη θεωρητική εισαγωγή στο πεδίο της Αφομοίωσης Δεδομένων (ΑΔ). Στην εξ. (3.1) παρουσιάζεται η στοχαστική θεώρηση ενός δυναμικού συστήματος, με $\mathbf{x}$ το διάνυσμα κατάστασης του μοντέλου, $\mathbf{M}$ τον πίνακα που αναπαριστά τη δυναμική του μοντέλου, ενώ το διάνυσμα $\mathbf{w}_k$ αναπαριστά το σφάλμα μοντελοποίησης με το αντίστοιχο μητρώο συνδιακύμανσης $\mathbf{Q}_k$. Ενώ στην εξ. (3.2) ορίζεται στοχαστικά το διάνυσμα παρατηρήσεων που συσχετίζονται με το φαινόμενο που περιγράφει το δυναμικό σύστημα, με $\mathbf{H}$ ένα μητρώο το οποίο αντιστοιχεί τον χώρο του μοντέλου στον χώρο των παρατηρήσεων και το διάνυσμα $\mathbf{v}_k$ αναπαριστά τα σφάλματα μέτρησης/παρατήρησης με το αντίστοιχο μητρώο συνδιακύμανσης $\mathbf{R}_k$.

Η μοντελοποίηση των $\mathbf{w}_k$ και $\mathbf{v}_k$ αποτελεί ένα σημαντικό κομμάτι της ΑΔ και στις μεθόδους που αναπτύχθηκαν στην παρούσα Εργασία, θεωρήθηκε ότι είναι ανεξάρτητα μεταξύ τους και ακολουθούν κανονικές κατανομές.

## 3.1 Στατιστικές Μέθοδοι

### 3.1.1 Βέλτιστη Παρεμβολή - BLUE

Η βασική ιδέα των στατιστικών μεθόδων ΑΔ είναι ο προσδιορισμός μιας κατάστασης (ανάλυση), η οποία είναι γραμμικός συνδυασμός της κατάστασης του μοντέλου και των παρατηρήσεων και τα σφάλματα υπολογισμού της έχουν την ελάχιστη συνδιακύμανση [1]. Έτσι, ο ορισμός της κατάστασης ανάλυσης φαίνεται στην εξ. (3.18), και το μητρώο βέλτιστου κέρδους $\mathbf{K}$ προκύπτει από το θεώρημα Gauss-Markov και φαίνεται στην εξ. (3.19). Ο εκτιμητής που αναπτύχθηκε ονομάζεται αποτελεσματικότερος αμερόληπτος γραμμικός εκτιμητής (BLUE), διότι ελαχιστοποιεί την συνδιακύμανση των σφαλμάτων (*βέλτιστος - αποτελεσματικότερος*), υποθέτει ότι η μέση τιμή των σφαλμάτων υπολογισμού της κατάστασης ανάλυσης είναι μηδενική (*αμερόληπτος*) και τέλος αποτελεί *γραμμικό* συνδυασμό της κατάστασης που υπολογιζεται από το μοντέλο και των παρατηρήσεων.

### 3.1.2 Φίλτρο Kalman

Ακολούθως, περιγράφεται η μέθοδος του φίλτρου Kalman [5], το οποίο αποτελεί ένα σχήμα πρόβλεψης-διόρθωσης και στοχεύει στην ελαχιστοποίηση της συνδιακύμανσης των σφαλμάτων της κατάστασης ανάλυσης σε κάθε χρονικό βήμα.

Στο βήμα πρόβλεψης γίνεται εκτίμηση της κατάστασης μέσω του δυναμικού μοντέλου και ανανεώνεται το μητρώο συνδιακύμανσης των σφαλμάτων, όπως φαίνεται στην εξ. (3.23). Ενώ στο βήμα διόρθωσης αρχικά υπολογίζεται το μητρώο κέδρους Kalman μέσω της εξ. (3.24) και στη συνέχεια ανανεώνονται η κατάσταση μέσω της εξ. (3.25), καθώς και το μητρώο συνδιακύμανσης των σφαλμάτων μέσω της εξ. (3.26).

## 3.2 Μέθοδοι Λογισμού των Μεταβολών

Η άλλη προσέγγιση σην ΑΔ βασίζεται στον λογισμό των μεταβολών και πιο συγκεκριμένα στην ελαχιστοποίηση μιας συνάρτησης κόστους η οποία εμπλέκει τα μητρώα συνδιακύμανσης των σφαλμάτων του μοντέλου και των παρατηρήσεων, το διάνυσμα κατάστασης του μοντέλου και το διάνυσμα των παρατηρήσεων. Επομένως, απαιτείται η χρήση μεθόδων βελτιστοποίησης, οι οποίες απαιτούν τον υπολογισμό της κλίσης της συνάρτησης κόστους, συνήθως ακολουθώντας την συζυγή προσέγγιση.

### 3.2.1 3D-Var

Για μόνιμα προβλήματα η πιο συνηθισμένη μέθοδος είναι η 3D-VAR, με τη συνάρτηση κόστους να ορίζεται στην εξ. (3.27). Ουσιαστικά η συνάρτηση κόστους της μεθόδου αποτελεί το σταθμισμένο άθροισμα των αποκλίσεων της υπολογιζόμενης κατάστασης από το μοντέλο και τις παρατηρήσεις, λαμβάνοντας υπόψιν τις αβεβαιότητες. Επιπρόσθετα, μπορεί να δειχθεί [1], ότι οι εκφράσεις του BLUE και της μεθόδου 3D-VAR είναι ισοδύναμες.

### 3.2.2 4D-Var

Επιπρόσθετα, σε μη-μόνιμα προβλήματα, πρέπει να ληφθεί υπόψιν η δυναμική του μοντέλου. Έτσι, γίνεται χρήση της μεθόδου 4D-VAR. Η συνάρτηση κόστους ορίζεται από τις εξ. (3.34)–(3.37), ενώ ο υπολογισμός των $\mathbf{M}_k^{\mathrm{T}}$ and $\mathbf{H}_k^{\mathrm{T}}$ γίνεται μέσω της συζυγούς μεθόδου. Τέλος, οι μέθοδοι 3D-Var και 4D-Var παρουσιάζονται συνοπτικά στο σχ. 3.2.

### 3.2.3 Μέθοδος Nudging

Τέλος, περιγράφεται μια απλούστερη μέθοδος, η οποία ονομάζεται nudging (ώθηση) ή Νευτώνεια χαλάρωση και έχει σκοπό την ώθηση του δυναμικού μοντέλου στις παρατηρήσεις, εισαγοντας έναν όρο ανάδρασης, ο οποίος είναι ανάλογος της διαφοράς των αποτελεσμάτων του μοντέλου και των παρατηρήσεων στο δυναμικό μοντέλο [48], όπως φαίνεται στην εξ. (3.39).

Η μέθοδος nudging μπορεί να υλοποιηθεί ευκολότερα σε σχέση με τις άλλες προσεγγίσεις, επεκτείνοντας τον υπολογιστικό κώδικα που επιλύει το δυναμικό μοντέλο. Ωστόσο, η βελτίωση που επιφέρει στα αποτελέσματα εξαρτάται από την επιλογή του μητρώου κέρδους $\mathbf{K}$. Μια προσέγγιση είναι η επιλογή του μητρώου του εκτιμητή BLUE, ενώ μια άλλη προσέγγιση με εφαρμογή σε μη-μόνιμα προβλήματα είναι η μέθοδος του βέλτιστου nudging [51, 52], όπου αναζητείται το μητρώο κέρδους $\mathbf{K}$, το οποίο ελαχιστοποιεί μια συνάρτηση κόστους που περιλαμβάνει το σφάλμα μεταξύ αριθμητικών αποτελεσμάτων και παρατηρήσεων κατά τη διάρκεια ενός χρονικού διαστήματος της προσομοίωσης.

# 4 Εφαρμογή 1: Ροή σε Κρουστικό Σωλήνα

Στην πρώτη εφαρμογή της Εργασίας, έγινε εφαρμογή μεθόδων ΑΔ στη διάδοση ασυνέχειας πίεσης και πυκνότητας για 1Δ ροή εντός κρουστικού σωλήνα απείρου μήκους (πρόβλημα του Sod). Αυτό το πρόβλημα χρησιμοποιείται ευρέως για την επιβεβαίωση κωδίκων ΥΔ [53, 54], ενώ υπάρχει διαθέσιμη αναλυτική λύση. Αρχικά, δημιουργήθηκε παράλληλος, μέσω του OpenMP [55], υπολογιστικός κώδικας σε Fortran 95, βασισμένος στον κώδικα που παρουσιάζεται στο μάθημα Υπολογιστική Ρευστομηχανική του 7$^{ου}$ εξαμήνου της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ.

Έπειτα, έγινε επιβεβαίωση του υπολογιστικού κώδικα και διερεύνηση της χρήσης διαφορετικών τάξεων των σχημάτων ανακατασκευής των συντηρητικών μεταβλητών από τα κέντρα των κελιών στις κοινές επιφάνειες των κελιών. Κατόπιν, υλοποιήθηκε η μέθοδος nudging κάνοντας χρήση της αναλυτικής λύσης ή αριθμητικών λύσεων σε πυκνότερα πλέγματα. Τέλος, έγινε βελτιστοποίηση του μητρώο κέδρους για την περίπτωση της μεθόδου nudging με την αναλυτική λύση.

## 4.1 Αριθμητική και Αναλυτική Λύση

Η περίπτωση που υλοποιήθηκε αφορά τη 1Δ μετάδοση της ασυνέχειας πυκνότητας και πίεσης ενός αρχικά ακίνητου ατριβούς ιδανικού αερίου εντός ενός σωλήνα απείρου μήκους, όπως φαίνεται στο σχ. 4.1. Το φαινόμενο περιγράφεται από τις 1Δ εξισώσεις Euler για συμπιεστό ρευστό εξ. (2.1) και (2.3).

### 4.1.1 Αναλυτική Λύση

Το σύστημα που περιγράφεται στην εξ. (2.3) είναι ένα σύστημα ημι-γραμμικών υπερβολικών μ.δ.ε. και η μετάδοση της πληροφορίας περιγράφεται από τρεις χαρακτηριστικές γραμμές που αντιστοιχούν σε ένα κύμα αποτόνωσης, μια ασυνέχεια επαφής και ένα κύμα κρούσης, όπως φαίνεται στο σχ. 4.2. Η χρονική εξέλιξη αυτών των χαρακτηριστικών χωρίζει το πεδίο του ρευστού σε 5 ξεχωριστές περιοχές, όπως φαίνεται στο σχ. 4.3, όπου μπορούν να υπολογιστούν αναλυτικά οι καταστάσεις του ρευστού [21, 23, 53].

Αναλυτικότερα, οι διακριτές καταστάσεις υπολογίζονται από τις εξ. (4.1)–(4.7), ενώ η κατάσταση εντός του κύματος αποτόνωσης (ισεντροπική διαδικασία), περιγράφεται από τις εξ. (4.8)–(4.10). Αντίστοιχα, οι θέσεις των ασυνεχειών συναρτήσει του χρόνου προσδιορίζονται μέσω των εξ. (4.11)–(4.13) και (4.15). Ενδεικτικά στο σχ. 4.4 παρουσιάζονται οι κατανομές των πρωταρχικών μεταβλητών $\mathbf{W} = [\rho \quad u \quad p]^{\mathrm{T}}$ τη χρονική στιγμή $t = 0.42\ s$ για $x \in [-5, 5]$.

### 4.1.2 Αριθμητική Λύση

Ο υπολογιστικός κώδικας που υλοποιήθηκε, χρησιμοποιεί το σχήμα Godunov για τη χρονική ολοκλήρωση των 1Δ συμπιεστών εξισώσεων Euler σε συντηρητική μορφή, με τον προσεγγιστικό επιλύτη του Roe για την επίλυση του προβλήματος Riemann. Ενώ για την ανακατασκευή των συντηρητικών μεταβλητών από τα κέντρα των κελιών στις κοινές επιφάνειες των κελιών, υλοποιήθηκαν σχήματα 1$^{ης}$ και 2$^{ης}$ τάξης (MUSCL [26]), με την επιλογή της χρήσης του συμμετρικού περιοριστή κλίσεων του van Leer [27].

Η αριθμητική επίλυση πραγματοποιήθηκε σε χωρίο μήκους $L = 10\text{m}$, με $x \in [-5, 5]\ \text{m}$ και την ασυνέχεια να εμφανίζεται στο $x_0 = 0$. Στα σύνορα χρησιμοποιήθηκαν οριακές συνθήκες συμμετρίας, ενώ ο συνολικός χρόνος προσομοίωσης επιλέχθηκε έτσι ώστε τα κύματα αποτόνωσης και κρούσης να μη φτάσουν στα σύνορα του υπολογιστικού χωρίου.

Καθώς το σχήμα Godunov αποτελεί ένα ρητό σχήμα ολοκλήρωσης, λαμβάνεται υπόψιν η συνθήκη σύγκλισης CFL $\leq 1$ [21, 23] για τον προσδιορισμό της χωρικής και χρονικής διακριτοποίησης, όπως φαίνεται στην εξ. (4.16). Πιο συγκεκριμένα, επιλέχθηκαν 1000 σημεία για την διακριτοποίηση του σωλήνα, ισοδύναμα $\Delta x = 10^{-2}$m, ενώ επιλέχθηκε χρονικό βήμα $\Delta t = 10^{-4}$s, με τη συνθήκη σύγκλισης να ικανοποιείται, καθώς CFL $\approx 10^{-2} \ll 1$.

Οι κατανομές των πρωταρχικών μεταβλητών παρουσιάζονται στο σχ. 4.6 τη χρονική στιγμή $t = 0.42$s. Ο κώδικας μπορεί να παρακολουθήσει τη γενική συμπεριφορά της ροής, ωστόσο αποτυγχάνει να αποτυπώσει τη συμπεριφορά των ασυνεχειών, όπως φαίνεται στα σχ. 4.6 και 4.7. Ακόμη, η επίδραση της χωρικής διακριτοποίησης φαίνεται στα σχ. 4.8 και 4.9, για σχήμα $1^{ης}$ τάξης και ίδιο χρονικό βήμα η αύξηση της πυκνότητας του πλέγματος οδηγεί σε μεγαλύτερη ακρίβεια της τελικής λύσης.

Επιπλέον, για την ποσοτικοποίηση του σφάλματος μεταξύ αριθμητικής και αναλυτικής λύσης, έγινε χρήση της της ρίζας της μέσης τετραγωνικής απόκλισης (RMSE) εξ. (4.17), η χρονική εξέλιξη των RMSE των πρωταρχικών μεταβλητών για διαφορετική χωρική διακριτοποίηση φαίνεται στο σχ. 4.10. Παρατηρείται ασυμπτωτική σύγκλιση των RMSE, με μια ταλαντωτική συμπεριφορά γύρω από μια σταθερή τιμή, ενώ όπως αναμένεται η αύξηση της πυκνότητας του πλέγματος οδηγεί σε βελτιωμένη ακρίβεια.

## 4.2 Nudging με την Αναλυτική Λύση

Έπειτα, υλοποιήθηκε η μέθοδος nudging για τις εξισώσεις Euler σε συντηρητική μορφή. Αρχικά, κάνοντας χρήσης της αναλυτικής λύσης στα κέντρα των κελιών, όπως φαίνεται στην εξ. (4.18), με το νέο σχήμα ολοκλήρωσης να φαίνεται στην εξ. (4.19). Το μητρώο κέρδους $\mathbf{K}$ είναι $3 \times 3$ και επιλέγεται να είναι σταθερό και διαγώνιο για όλο το υπολογιστικό χωρίο. Ακόμη, καθώς όλες οι πρωταρχικές μεταβλητές έχουν τη ίδια τάξη μεγέθους, επιλεχθηκε το $\mathbf{K}$ να είναι μια βαθμωτή ποσότητα ($\mathbf{K} = K\mathbf{I_3}$), ενώ η τάξη μεγέθους του $K$ επιλέχθηκε μετά από δοκιμές.

Στα σχ. 4.11 και 4.12, φαίνεται η αριθμητική λύση τη χρονική στιγμή $t = 0.42$s για διαφορετικές τιμές του $K$ κάνοντας χρήση σχήματος $1^{ης}$ τάξης. Η βελτίωση είναι εμφανής ακόμη και για $K = 1$, καθώς τα αριθμητικά αποτελέσματα είναι πιο κοντά στην αναλυτική λύση. Επιπλέον, όπως φαίνεται στο σχ. 4.13, η χρονική εξέλιξη των RMSE των συντηρητικών μεταβλητών για διαφορετικές τιμές του $K$ τείνει σε μια ταλαντωτική συμπεριφορά γύρω από μία σταθερή μέση τιμή, η οποία μειώνεται σημαντικά με την χρήση της μεθόδου nudging, ενώ τα πλάτη αυτών των ταλαντώσεων αυξάνονται με την αύξηση του $K$.

## 4.3 Nudging με δύο Αριθμητικές Λύσεις

Έπειτα, ο υπολογιστικός κώδικας τροποποιήθηκε ώστε να μπορεί να επιλύει το πρόβλημα, κάνοντας χρήση δύο διαφορετικών χωρικών διακριτοποιήσεων. Αρχικά, η ροή επιλύεται δίχως nudging, όπως φαίνεται στην εξ. (4.20), για ένα πυκνό πλέγμα με NXF σημεία. Ακολούθως, η ροή επιλύεται σε ένα αραιότερο πλέγμα για ίδιο χρονικό βήμα, με NXC σημεία (NXC < NXF), εφαρμόζοντας τη μέθοδο nudging, με την λύση του πυκνού πλέγματος να θεωρείται ως η πραγματική κατάσταση, όπως περιγράφεται στην εξ. (4.21).

Έτσι, διερευνήθηκε η χρήση διαφορετικών ζευγών (NXC, NXF) και έγινε σύκγριση με τα αποτελέσματα δίχως τη μέθοδο nudging για πυκνότητα πλέγματος μια τάξη μεγέθους μεγαλύτερη από αυτή του ζεύγους (NX $= 10^4$). Η χρονική μεταβολή των RMSE των συντηρητικών μεταβλητών για διαφορετικά ζεύγη (NXC, NXF) παρουσιάζεται στα σχ. 4.14–4.16. Αρχικά, τα RMSE που προκύπτουν κατά τη χρήση δυό χωρικών διακριτοποιήσεων είναι χαμηλότερα σε σχέση με αυτά που προκύπτουν για ένα πολύ πυκνότερο πλέγμα (NX $= 10^4$). Ακόμη, στο τέλος

της προσομοίωσης, οι τιμές των RMSE μειώνονται καθώς η ανάλυση του πυκνού πλέγματος του ζεύγους αυξάνεται, ενώ παρατηρείται μια ταλαντωτική συμπεριφορά της οποίας το πλάτος αυξάνεται καθώς αυξάνεται η ανάλυση του πυκνού πλέγματος του ζεύγους.

## 4.4 Βελτιστοποίηση του Μητρώου Κέρδους

Τέλος, έγινε βελτιστοποίηση του μητρώου κέρδους με στόχο την ελαχιστοποίηση μιας συνάρτησης κόστους, η οποία εκφράζει την απόκλιση μεταξύ αριθμητικής και πειραματικής λύσης για ένα χρονικό διάστημα της προσομοίωσης. Αυτή η μέθοδος ονομάζεται βέλτιστο nudging [1, 51, 52]. Έτσι, γίνεται εύρεση του βέλτιστου μητρώου κέρδους $\mathbf{K}_{\text{optimal}}$ το οποίο ελαχιστοποιεί τη συνάρτηση κόστους για ένα χρονικό διάστημα $[t_1, t_2]$ και στη συνέχεια η προσομοίωση συνεχίζεται με $\mathbf{K} = \mathbf{K}_{\text{optimal}}$.

Η συνάρτηση κόστους $\mathbf{J}$ ορίζεται στην εξ. (4.22). Ενώ για την εύρεση του βέλτιστου $\mathbf{K}$, έγινε χρήση της μεθόδου της απότομης καθόδου, όπως περιγράφεται στην εξ. (4.23). Έτσι, προκύπτει η ανάγκη υπολογισμού των παραγώγων ευαισθησίας $\frac{\delta \mathbf{J}}{\delta \mathbf{K}}$ και κατά συνέπεια των $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}$. Για την αποφυγή του απευθείας υπολογισμού των $\frac{\delta \mathbf{U}}{\delta \mathbf{K}}$, έγινε χρήση της συνεχούς συζυγούς μεθόδου [57, 58]. Οι τελική μορφή των συζυγών εξισώσεων Euler παρουσιάζεται στην εξ. (4.31), οι οποίες επιλύονται ανάποδα στο χρονικό διάστημα $[t_1, t_2]$, μέσω του σχήματος της εξ. (4.35). Ενώ η τελική μορφή των παραγώγων ευαισθησίας παρουσιάζεται στην εξ. (4.36).

Έτσι, ο υπολογιστικός κώδικας επεκτάθηκε ώστε να επιλύει τις συζυγείς εξισώσεις και να εκτελεί ένα βρόγχο βελτιστοποίησης για το $\mathbf{K}$. Αρχικά, το χρονικό διάστημα βελτιστοποίησης επιλέχθηκε το $(\text{NT}_1, \text{NT}_2) = (500, 1500)$, ή ισοδύναμα $(t_1, t_2) = (0.05, 0.15)$ s με συνολικό χρόνο προσομοίωσης $t_{end} = 0.42$s. Επιπλέον, επιλέχθηκε βήμα $\eta = 10^4$ για τη μέθοδο της απότομης καθόδου, ενώ το κριτήριο σύγκρισης που επιλεχθηκε ήταν $|\mathbf{K}_{\text{new}} - \mathbf{K}_{\text{old}}| \leq 10^{-3}$, με το μέγιστο πλήθος επαναλήψεων του βρόγχου βελτιστοποίησης να είναι 15 επαναλήψεις.

Στο σχ. 4.17 παρουσιάζεται η εξέλιξη των RMSE πριν και μετά τη διαδικασία της βελτιστοποίησης για διαφορετική αρχικοποίησης του $\mathbf{K}$. Παρατηρείται ότι το βέλτιστο μητρώο κέρδους οδηγεί σε περίπου μια τάξη μεγέθους χαμηλότερες ελάχιστες τιμές των RMSE. Αυτό συμβαίνει διότι μέσω της βελτιστοποιησης 'γεμίζονται' τα μη διαγώνια στοιχεία του $\mathbf{K}$, μοντελοποιώντας καλύτερα τις πεπλεγμένες συνεισφορές στον όρο πηγής nudging των συντηρητικών μεταβλητών. Στο σχ. 4.18 παρουσιάζεται η εξέλιξη των RMSE κατά τη διάρκεια του βρόγχου βελτιστοποίησης. Η βελτίωση της ακρίβειας, τείνει να μειώνεται καθώς αυξάνονται οι επαναλήψεις, παρουσιάζοντας συμπεριφορά ασυμπτωτικής σύγκλισης, όπως φαίνεται και στο σχ. 4.19 για τις τρεις συνιστώσες της συνάρτηση κόστους.

# 5 Εφαρμογή 2: Υπολογισμός Πεδίων Πίεσης από Δεδομένα PIV

Στην δεύτερη εφαρμογή της Εργασίας, εφαρμόστηκαν μέθοδοι ΑΔ στον υπολογισμό πεδίων πίεσης από δεδομένα PIV για τη τη μόνιμη 2Δ τυρβώδη ροή γύρω από πακτωμένο κύβο εντός του ατμοσφαιρικού οριακού στρώματος. Ενώ η υλοποίηση έγινε με τη χρήση του παράλληλου λογισμικού ΥΡ ανοιχτού κώδικα OpenFOAM. Το OpenFOAM είναι ένας αντικειμενοστραφής υπολογιστικός κώδικας γραμμένος σε C++ και περιέχει έτοιμους επιλύτες καθώς και κλάσεις οι οποίες μπορούν να αποτελέσουν τα δομικά στοιχεία νέων επιλυτών. Στην παρούσα Εργασία έγινε επέκταση του επιλύτη simpleFOAM, ο οποίος κάνει χρήση του αλγορίθμου SIMPLE για την επίλυση μόνιμων ασυμπίεστων ροών.

Τα πεδία ταχυτήτων και τάσεων Reynolds είναι διαθέσιμα μέσω της μεθόδου PIV [61] από προηγούμενη πειραματική εργασία [41, 42], η οποία πραγματοποιήθηκε στο Εργαστήριο Τεχνολογικών Καινοτομιών Προστασίας Περιβάλλοντος της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ. Η μελέτη έγινε για πλήρως τυρβώδη ροή με τον αριθμό Reynolds στο ύψος του κύβου να είναι $\mathrm{Re} = 2.4 \cdot 10^4$ και λήφθηκαν μετρήσεις σε τρία επίπεδα στην κεντρική γραμμή του κύβου, όπως φαίνεται στο σχ. 5.1, για συνθήκες υψηλής (HS) και χαμηλής (LS) ατμοσφαιρικής διάτμησης, για κύβους με (open) και χωρίς (closed) ανοίγματα.

Για τον υπολογισμό του πεδίου πίεσης έγινε χρήση δύο μεθόδων, η πρώτη αφορά τη χρήση μιας εξίσωσης Poisson, ενώ η δεύτερη αφορά τη χρήση του αλγορίθμου SIMPLE για την ταυτόχρονη διόρθωση των πεδίων πίεσης και ταχύτητας, κάνοντας χρήση των δεδομένων PIV ως αρχικές και οριακές συνθήκες [44]. Έτσι, αφού τα πειραματικά δεδομένα μετατράπηκαν σε μορφή συμβατή με το OpenFOAM, οι δύο μέθοδοι υλοποιήθηκαν στο OpenFOAM και έγινε σύγκριση της απόδοσης τους. Έπειτα, έγινε μελέτη ανεξαρτησίας της λύσης από την πυκνότητα του πλέγματος. Τέλος, εφαρμόστηκαν και συγκρίθηκαν μέθοδοι ΑΔ (nudging και φίλτρο Kalman).

## 5.1 Αλγόριθμος SIMPLE στο OpenFOAM

Η υλοποίηση του αλγορίθμου SIMPLE στο OpenFOAM διαφέρει σε σχέση με αυτή που περιγράφηκε προηγουμένως [28, 30]. Αρχικά, η υποχαλάρωση της ταχύτητας γίνεται με πεπλεγμένο τρόπο όπως φαίνεται στην εξ. (5.1). Ακόμη, δεν υπολογίζονται πεδία διόρθωσης πίεσης και ταχύτητας, αλλά νέα (ολόκληρα) πεδία πίεσης και ταχύτητας. Τέλος, για την αποφυγή του φαινομένου της σκακιέρας (checkerboard), γίνεται έμμεση εφαρμογή της παρεμβολής Rhie-Chow.

Έτσι, μετά την αρχική εκτίμηση των πεδίων $p$ και $\mathbf{u}$, εκκινείται μια επαναληπτική διαδικασία κατά την οποία αρχικά υπολογίζεται ένα πεδίο ταχύτητας το οποίο ικανοποιεί την εξίσωση ορμής μέσω της εξ. (5.2) και το αντίστοιχο πεδίο πίεσης, το οποίο ικανοποιεί την εξίσωση συνέχειας μέσω της εξ. (5.5). Έπειτα γίνεται διόρθωση των ογκομετρικών ροών μέσω της εξ. (5.6) και κατόπιν γίνεται άμεση χαλάρωση του πεδίου πίεσης. Τέλος, υπολογίζεται το νέο πεδίο ταχύτητας μέσω της εξ. (5.7). Έτσι, στη λήξη της επαναληπτικής διαδικασίας ικανοποιούνται οι εξισώσεις διατήρησης της μάζας και ορμής.

## 5.2 Δημιουργία Πλεγμάτων

Για την επίλυση των εξισώσεων NS με το OpenFOAM, έγινε χρήση των πλεγμάτων όπου είναι διαθέσιμα τα δεδομένα PIV, δηλαδή σε κάθε επίπεδο μέτρησης. Καθώς το OpenFOAM κάνει χρήση της κεντροκυψελικής διατύπωσης, έγινε παρεμβολή των μεγεθών (ταχύτητες και τάσεις Reynolds) από τους κόμβους των πλεγμάτων στα κέντρα των κελιών.

Έτσι, εξασφαλίστηκε ότι γίνονται μόνο παρεμβολές, αποφεύγοντας τις προεκβολές στα σύνορα. Επιπλέον, με αυτό την επιλογή τα πλέγματα που χρησιμοποιήθηκαν για την αριθμητική

επίλυση έχουν ακριβώς τις ίδιες διαστάσεις και σύνορα με αυτά της πειραματικής μεθόδου PIV, όπως φαίνεται στο σχ. 5.2.

## 5.3 Μέθοδοι Υπολογισμού Πεδίων Πίεσης από Δεδομένα PIV

Στη βιβλιογραφία [64], συναντώνται τέσσερις βασικές μεθοδολογίες υπολογισμού του πεδίου πίεσης για δεδομένο πεδίο ταχύτητας μέσω της μεθόδου PIV. Συνοπτικά: επίλυση της εξίσωσης Bernoulli, ολοκληρωματικές μέθοδοι, εξίσωση Poisson, επαναληπτικές μέθοδοι [44, 65]. Στην παρούσα Εργασία, εφαρμόστηκε η μέθοδος της επίλυσης μιας εξίσωσης Poisson, υποθέτοντας ότι τα πεδία ταχύτητας που μετρήθηκαν ικανοποιούν την εξίσωση συνέχειας, όπως φαίνεται στην εξ. (5.10). Καθώς και η επαναληπτική μέθοδος η οποία κάνει χρήση του αλγορίθμου SIMPLE για την ταυτόχρονη διόρθωση των πεδίων πίεσης και ταχύτητας, με τα πειραματικά πεδία ταχύτητας να αποτελούν την αρχική εκτίμηση για την έναρξη του αλγορίθμου SIMPLE.

Η ουσιαστική διαφορά των δυό μεθόδων έγκειται στην παραδοχή της πρώτης ότι ικανοποιείται η εξίσωση της συνέχειας, κάτι το οποίο δεν μπορεί να εξασφαλιστεί. Ωστόσο, και οι δύο μεθοδολογίες απαιτούν την επίλυση μια εξίσωσης Poisson, η οποία απαιτεί ιδιαίτερη μέριμνα ως προς την επιλογή των οριακών συνθηκών της πίεσης. Αναλυτικότερα, έγινε χρήση μηδενικών συνθηκών Dirichlet για την πίεση στις πλευρές των επιπέδων μέτρησης που δεν εφάπτονται στον κύβο και χρήση μηδενικών συνθηκών Neumann (μηδενική κλίση) για την πίεση στις πλευρές που εφάπτονται στον κύβο. Αντίστοιχα, για την ταχύτητα επιλέχθηκαν συνθήκες Dirichlet ίσες με τις πειραματικές τιμές και επιπλέον οι σταθερές τιμές επεκτάθηκαν σε μια ζώνη βάθους δύο κελιών εντός του υπολογιστικού χωρίου σύμφωνα με προηγούμενη δουλεία στο Εργαστήριο Τεχνολογικών Καινοτομιών Προστασίας Περιβάλλοντος της Σχολής Μηχανολόγων Μηχανικών του ΕΜΠ [44], όπως φαίνεται στο σχ. 5.3.

## 5.4 Επιλογές Προσομοίωσης

Ακόμη, όπως αναφέρθηκε παραπάνω δεν έγινε χρήση μοντέλου τύρβης, αλλά οι τάσεις Reynolds χρησιμοποιήθηκαν όπως προέκυψαν από τις πειραματικέ μετρήσεις. Ακόμη, για την επίλυση των γραμμικών συστημάτων έγινε χρήση επαναληπτικών μεθόδων. Πιο συγκεκριμένα, για την εξίσωση πίεσης χρησιμοποιήθηκε η προσταθεροποιημένη μέθοδος των συζυγών κλίσεων (PCG) με τον διαγώνιο μερικό προσταθεροποιητή Cholesky (DIC). Αντίστοιχα, για την εξίσωση ορμής χρησιμοποιήθηκε η προσταθεροποιημένη μέθοδος των δι-συζυγών κλίσεων (PBiCG) με τον διαγώνιο μερικό προσταθεροποιητή LU (DILU). Τέλος, έγινε χρήση των σχημάτων διακριτοποίησης όπως αυτά περιγράφονται στα υποκεφάλαια 2.2.2.1–2.2.2.4 και συνοψίζονται στον πιν. 5.1.

## 5.5 Μετρικές Αξιολόγησης

Ακόμη για την αξιολόγηση των αποτελεσμάτων έγινε χρήση της κατανομής του συντελεστή πίεσης $C_p$, όπως αυτός ορίζεται στην εξ. (5.11) και σύγκριση του με τα αντίστοιχα πειραματικά δεδομένα [41, 42]

Επιπλέον, για την αξιολόγηση της ποιότητας των δεδομένων PIV και των αριθμητικών αποτελεσμάτων έγινε χρήση του κανονικοποιημένου υπολοίπου συνέχειας CRN, όπως φαίνεται στην εξ. (5.12), με $\langle|\mathbf{u}_{\mathrm{PIV}}|\rangle$ το μέσο μέτρο το πεδίου ταχύτητας των δεδομένων PIV. Ακόμη, στην εξ. (5.13) ορίζεται το κανονικοποιημένο σχετικό σφάλμα υπολογισμού της ταχύτητας. Επιλέχθηκε, η κανονικοποίηση με το μέσο μέτρο της ταχύτητας, ώστε να αποφεύγονται οι απειρισμοί κοντά στα σημεία με πολύ χαμηλές τιμές ταχυτήτων. Επιπλέον, στην εξ. (5.14) ορίζεται

το απόλυτο σφάλμα υπολογισμού. Ωστόσο, οι παραπάνω μετρικές προφανώς είναι βαθμωτά πεδία, για αυτό το λόγο επιπλέον των σχημάτων των ισογραμμών τους, δημιουργήθηκαν τα ιστογράμματα της κατανομής τους εντός των επιπέδων. Στον οριζόντιο άξονα αποτυπώνονται οι κλάσεις των παραπάνω πεδίων, ενώ στον κατακόρυφο άξονα το αντίστοιχο κλάσμα όγκου όπως ορίζεται στην εξ. (5.15).

## 5.6 Ανεξαρτησία Πλέγματος

Τέλος, στην ΥΡ ένας από τους βασικότερους περιορισμούς ως προς την ποιότητα της τελικής λύσης είναι η ποιότητα του πλέγματος. Το πλέγμα που διατίθεται από το PIV είναι απολύτως ορθογωνικό, ωστόσο είναι σχετικά αραιό σε σχέση με ένα τυπικό πλέγμα ΥΡ. Για αυτό το λόγο έχει αρκετό ενδιαφέρον η μελέτη της επίδρασης της πυκνότητας του πλέγματος του CFD στην ποιότητα των τελικών λύσεων. Η πύκνωση του αρχικού πλέγματος έγινε μέσω της εφαρμογής refineMesh του OpenFOAM ενώ η παρεμβολή των δεδομένων έγινε μέσω της εφαρμογής mapFields. Στο OpenFOAM η εφαρμογή refineMesh διαιρεί το κάθε κελί του ορθογωνικού πλέγματος σε 4 μικρότερα. Έτσι, προκύπτουν ένα πυκνότερο (×4) και ένα πυκνότατο (×16) πλέγμα για κάθε περίπτωση που μελετήθηκε, όπως φαίνεται και στο σχ. 5.4. Επομένως, μετά την επίλυση της ροής πάνω σε αυτά τα εκλεπτυσμένα πλέγματα γίνεται σύγκριση των αποτελεσμάτων με αυτά των ονομαστικών πλεγμάτων.

## 5.7 Επίπεδο Α

Αρχικά, γίνεται εφαρμογή των μεθόδων στο επίπεδο Α. Το επίπεδο αυτό βρίσκεται ανάντη του κύβου με αποτέλεσμα να εμφανίζεται σημείο ανακοπής περίπου για $y \approx 0.7H_{cube}$. Όπως φαίνεται στις ισογραμμές του μέτρου της ταχύτητας στα σχ. 5.6, 5.15, 5.23 και 5.27, οι ταχύτητες που προκύπτουν από τον αλγόριθμο SIMPLE είναι κοντά με τα πειραματικά δεδομένα. Επιπλέον, όπως φαίνεται στις ισογραμμές του CRN στα σχ. 5.8, 5.17, 5.23 και 5.27, και στα ιστογράμματα της κατανομής του CRN στα σχ. 5.11, 5.20, 5.24 και 5.28, η μέθοδος του αλγορίθμου SIMPLE διορθώνει τα πεδία ταχύτητας ως προς την ικανοποίηση της συνέχειας σε σχέση με τα πειραματικά δεδομένα.

Επιπρόσθετα, όπως φαίνεται στις ισογραμμές του πεδίου πίεσης στα σχ. 5.7, 5.16, 5.23 και 5.27 και στην κατανομή $C_p$ στα σχ. 5.5 και 5.14, τα πεδία πίεσης που προέκυψαν από τον SIMPLE ήταν αρκετά κοντά στα πειραματικά δεδομένα και στην προηγούμενη υλοποίηση της μεθόδου σε άλλον υπολογιστικό κώδικα [44]. Αντιθέτως, η χρήση της εξίσωσης Poisson αποτυγχάνει στην πρόβλεψη των πεδίων πίεσης, καθώς όπως αναφέρθηκε παραπάνω τα αρχικά πεδία ταχύτητας δεν ικανοποιούν την εξίσωση συνέχειας.

Ακόμη, όπως φαίνεται στα σχ. 5.6, 5.7, 5.15, 5.16, 5.23 και 5.27, η εκλεπτυνση των πλεγμάτων δεν οδηγεί σε εμφανείς διαφορές στα πεδία πίεσης και ταχύτητας, ενώ εμφανίζεται βελτίωση στα πεδία CRN καθώς με την πύκνωση του πλέγματος μειώνεται το ποσοστό των κελιών όπου η ταχύτητα παραμένει σταθερή αό τις οριακές συνθήκες, όπως φαίνεται στα σχ. 5.8 και 5.17 και στα ιστογράμματα στα σχ. 5.11, 5.20, 5.24 και 5.28.

## 5.8 Επίπεδο Β

Έπειτα, γίνεται εφαρμογή των μεθόδων στο επίπεδο Β. Το επίπεδο αυτό βρίσκεται πάνω από τον κύβο με αποτέλεσμα να εμφανίζεται αποκόλληση της ροής στην κάτω επιφάνεια του επιπέδου. Αρχικά, όπως φαίνεται στις ισογραμμές του μέτρου ταχύτητας στα σχ. 5.32, 5.37, 5.41 και 5.45, οι ταχύτητες που προκύπτουν από τον αλγόριθμο SIMPLE είναι κοντά με τα πειραματικά δεδομένα.

Επιπλέον, η εξίσωση Poisson και ο αλγόριθμος SIMPLE παράγουν παρόμοια πεδία πίεσης τα οποία είναι κοντά στα πειραματικά δεδομένα όπως φαίνεται στις ισογραμμές πίεσης στα σχ. 5.32, 5.37, 5.41 και 5.45 και στις κατανομές $C_p$ στα σχ. 5.31 και 5.36. Αυτή η συμπεριφορά οφείλεται στην καλύτερη ικανοποίηση της συνέχειας από τα πειραματικά δεδομένα όπως φαίνεται στι ισογραμμές CRN στα σχ. 5.32, 5.37, 5.41 και 5.45 και στα ιστογράμματα της κατανομής του CRN, στα σχ. 5.33, 5.38, 5.42 και 5.46. Τέλος, η πύκνωση των πλεγμάτων δεν επιδρά σημαντικά τα τελικά πεδία πίεσης και ταχύτητας, όπως φαίνεται στα σχ. 5.32, 5.37, 5.41 και 5.45.

## 5.9 Επίπεδο C

Τέλος, γίνεται εφαρμογή των μεθόδων στο επίπεδο C. Το επίπεδο αυτό βρίσκεται στον ομόρρου του κύβου με αποτέλεσμα να εμφανίζεται μια ζώνη ανακυκλοφορίας της ροής. Σημαντική είναι η παρατήρηση ότι η περίπτωση υψηλής διάτμησης με κλειστό κύβο έχει ένα εξαιρετικά αραιό πλέγμα, όπως φαίνεται στον πιν. 5.4, με αποτέλεσμα η διαθέσιμη πληροφορία ναι είναι πολύ πτωχή.

Αρχικά, από τις ισογραμμές των πεδίων ταχύτητας στα σχ. 5.50, 5.55, 5.59 και 5.63, φαίνεται ότι οι ταχύτητες που προκύπτουν από τον αλγόριθμο SIMPLE είναι κοντά με τα πειραματικά δεδομένα. Επιπλέον, όπως φαίνεται στις ισογραμμές των πεδίων πίεσης στα σχ. 5.50, 5.55, 5.59 και 5.63, τα πεδία πίεσης που προέκυψαν από τον SIMPLE ήταν αρκετά κοντά στα πειραματικά δεδομένα όπως φαίνεται και στην κατανομή των $C_p$ στα σχ. 5.49. Τέλος, η εκλεπτυνση των πλεγμάτων δεν επιδρά σημαντικά τα τελικά πεδία πίεσης και ταχύτητας, όπως φαίνεται στα σχ. 5.50, 5.55, 5.59 και 5.63.

## 5.10 Αφομοίωση Δεδομένων

Έπειτα, ο επιλύτης επεκτάθηκε με την εφαρμογή μεθόδων ΑΔ. Πιο συγκεκριμένα, υλοποιήθηκαν οι μέθοδοι nudging και του φίλτρου Kalman με διαφορετικές παραλλαγές.

### 5.10.1 Μέθοδος Nudging

Η μέθοδος nudging περιγράφηκε αναλυτικά στο υποκεφάλαιο 3.4. Οι τροποποιημένες εξισώσεις RANS με τον όρο nudging, παρουσιάζονται στις εξ. (5.16) και (5.17). Αυτή η τροποποίηση υλοποιήθηκε στο OpenFOAM, και υλοποιήθηκαν δύο διαφορετικές επιλογές για το μητρώο $\mathbf{K}$.

Η πρώτη επιλογή είναι το 'ταυτοτικό' nudging, όπου επιλέχθηκε $\mathbf{K} = \mathbf{I}$ δίχως να λαμβάνονται υπόψιν οι αβεβαιότητες είτε του μοντέλου είτε των πειραματικών δεδομένων.

Η δεύτερη επιλογή ήταν η χρήση του μητρώου $\mathbf{K}$ όπως αυτό προκύπτει από τον εκτιμητή BLUE. Το μητρώο συνδιακύμανσης των σφαλμάτων των παρατηρήσεων προκύπτει από τον τανυστή των τάσεων Reynolds, υποθέτοντας ότι δεν υπάρχει χωρική συσχέτιση μεταξύ των σφαλμάτων των μετρήσεων, το μητρώο προκύπτει διαγώνιο $\mathbf{R} = \sigma^2_{\mathrm{PIV},i}\delta_{ij}$. Ακόμη, στον BLUE, το μητρώο συνδιακύμανσης των σφαλμάτων πρόβλεψης $\mathbf{P}$ ταυτίζεται με το μητρώο συνδιακύμανσης των σφαλμάτων του μοντέλου $\mathbf{Q}$. Για την εκτίμηση του $\mathbf{Q}$, γίνεται η παραδοχή ότι είναι διαγώνιο και εξαρτάται μόνο από τη χωρική διακριτοποίηση [6] και την ελάχιστη τάξη των σχημάτων διακριτοποίησης και κατά συνέπεια $\mathbf{Q} = C\Delta x\mathbf{I}$, με $\Delta x$ το μέσο μήκος των κελιών.

### 5.10.2 Φίλτρο Kalman

Επιπλέον, υλοποιήθηκε η μέθοδος του φίλτρου Kalman, όπως περιγράφηκε στο υποκεφάλαιο 3.2.2. Αρχικά, καθώς το φίλτρο Kalman απαιτεί ένα μητρώο που αναπαριστά τη δυναμική των εξισώσεων NS, γίνεται η προσέγγιση $\mathbf{M} = \mathbf{A}^{-1}\mathbf{I}$ σύμφωνα μες τη βιβλιογραφία [6], όπου $\mathbf{A}$

είναι το μητρώο που περιέχει τους συντελεστές της διαγωνίου των διακριτοποιημένων εξισώσεων ορμής.

Υλοποιήθηκαν δύο διαφορετικές διατυπώσεις της μεθόδου. Στην πρώτη διατύπωση, την 'κλασσική' διατύπωση, όπου κάθε επανάληψη του αλγορίθμου SIMPLE θεωρείται το μοντέλο από το οποίο προκύπτει το πεδίο ταχύτητας το οποίο απαιτείται στο βήμα πρόβλεψη του φίλτρου Kalman.

Στη δεύτερη διατύπωση, την 'ενσωματωμένη' διατύπωση, συνδυάζονται τα δύο σχήματα πρόβλεψης-διόρθωσης (φίλτρο Kalman και αλγόριθμος SIMPLE) σε ένα [7], με μια μικρή τροποποίηση της εξίσωσης πίεσης του αλγορίθμου SIMPLE, όπως φαίνεται στην εξ. (5.24), εξασφαλίζοντας ότι η κατάσταση ανάλυσης θα ικανοποιεί την εξίσωση συνέχειας.

## 5.11 Αποτελέσματα

Έτσι, μετά την υλοποίηση όλων των παραπάνω στο περιβάλλον του OpenFOAM, η ροή επιλύθηκε ξανά. Για την αξιολόγηση των αποτελεσμάτων, έγινε χρήση του RMSE του μέτρου της ταχύτητας, της μέσης τιμής και τυπικής απόκλισης των πεδίων CRN και κανονικοποιημένου σχετικού σφάλματος ταχύτητας, καθώς οι διαφορές στα πεδία πίεσης και ταχύτητας είναι δυσδιάκριτες.

Στους πιν. 5.5–5.16 φαίνεται η απόδοση των διαφορετικών μεθόδων ΑΔ για τις διαφορετικές περιπτώσεις που εξετάστηκαν. Με εξαίρεση την περίπτωση του επιπέδου C με συνθήκες υψηλής διάτμησης για κλειστό κύβο, που φαίνεται στον πιν. 5.14, όπου το αρχικό πλέγμα είναι ιδιαίτερα αραιό σε σχέση με τις υπόλοιπες περιπτώσεις, η κλασσική διατύπωση του φίλτρου Kalman παρουσιάζει την καλύτερη συμπεριφορά σε σχέση με τις υπόλοιπες μεθόδους ΑΔ, καθώς επιτυγχάνει τις χαμηλότερες τιμές RMSE καθώς και τις χαμηλότερες μέσες τιμές και τυπικές αποκλισεις του CRN και του κανονικοποιημένου σχετικού σφάλματος ταχύτητας. Έτσι, αφενός καταφέρνει να μην απομακρύνεται ιδιαίτερα από τα πειραματικά δεδομένα, αφετέρου δεν οδηγεί σε λύσεις οι οποίες αποτυγχάνουν στη διατήρηση της συνέχειας, ενώ ταυτόχρονα μειώνει την μεταβλητότητα των πεδίων.

Ακόμη, όπως αναμενόταν, η μέθοδος nudging που χρησιμοποιεί το μητρώο κέρδους του εκτιμητή BLUE, αποδίδει καλύτερα σε σχέση με το 'ταυτοτικό' nudging, οδηγώντας σε χαμηλότερες μέσες τιμές και τυπικές αποκλίσεις. Τέλος, η ενσωματωμένη διατύπωση του φίλτρου Kalman, αν και αρκετά ενδιαφέρουσα, δεν παρουσιάζει σημαντική βελτίωση σε σχέση με τη μέθοδο του nudging.

# 6 Συμπεράσματα

Συνοψίζοντας, στην παρούσα Διπλωματική Εργασία εφαρμόστηκαν επιτυχώς μέθοδοι Αφομοίωσης Δεδομένων σε δύο προβλήματα Υπολογιστικής Ρευστοδυναμικής. Στην πρώτη εφαρμογή, την 1Δ ροή εντός κρουστικού σωλήνα, παρατηρήθηκε ότι η αριθμητική λύση δεν μπορούσε να προσεγγίσει ικανοποιητικά των αναλυτική λύση. Η χρήση της μεθόδου nudging είτε μέσω της αναλυτικής λύσης είτε μέσω της αριθμητικής λύσης σε ένα πυκνότερο πλέγμα, οδηγεί σε σημαντική βελτίωση της ακρίβειας των αποτελεσμάτων. Τέλος, η βελτιστοποίηση του μητρώου κέρδους **K** βελτιώνει τα αποτελέσματα της μεθόδου nudging, ωστόσο πρέπει να λαμβάνεται υπόψιν η εξισορρόπηση μεταξύ υπολογιστικού κόστους και της βελτίωσης που προκύπτει.

Στη δεύτερη εφαρμογή που εξετάστηκε, τον υπολογισμό πεδίων πίεσης από δεδομένα PIV, ο επιλύτης που αναπτύχθηκε στο περιβάλλον του OpenFOAM κατάφερε να παράξει πεδία πίεσης τα οποία είναι σε συνέπεια με τα πειραματικά δεδομένα, ενώ ταυτόχρονα, τα πεδία ταχύτητας βελτιώθηκαν ως προς την ικανοποίηση της εξίσωσης συνέχειας. Ακόμη, η πύκνωση του πλέγματος δεν έδειξε να επηρεάζει σημαντικά την απόδοση της μεθόδου. Τέλος, η χρήση μεθόδων ΑΔ βελτίωσε εν γένει τα αποτελέσματα, με τη μεγαλύτερη βελτίωση να προκύπτει από την χρήση της κλασσικής διατύπωσης του φίλτρου Kalman.

# Bibliography


[1] M. Asch, M. Bocquet, and M. Nodet, *Data Assimilation: Methods, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Dec. 2016. DOI: 10.1137/1.9781611974546.

[2] T. Hayase, "Numerical simulation of real-world flows", *Fluid Dynamics Research*, vol. 47, no. 5, p. 051 201, Sep. 2015, ISSN: 1873-7005. DOI: 10.1088/0169-5983/47/5/051201.

[3] D. P. G. Foures, N. Dovetta, D. Sipp, and P. J. Schmid, "A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction", *Journal of Fluid Mechanics*, vol. 759, pp. 404–431, Nov. 2014, ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2014.566.

[4] S. Symon, N. Dovetta, B. J. McKeon, D. Sipp, and P. J. Schmid, "Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil", *Experiments in Fluids*, vol. 58, no. 5, Apr. 2017. DOI: 10.1007/s00348-017-2336-8.

[5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 0021-9223. DOI: 10.1115/1.3662552.

[6] M. Meldi and A. Poux, "A reduced order model based on Kalman filtering for sequential data assimilation of turbulent flows", *Journal of Computational Physics*, vol. 347, pp. 207–234, Oct. 2017. DOI: 10.1016/j.jcp.2017.06.042.

[7] C. Introini, S. Lorenzi, A. Cammi, D. Baroli, B. Peters, and S. Bordas, "A Mass Conservative Kalman Filter Algorithm for Computational Thermo-Fluid Dynamics", *Materials*, vol. 11, no. 11, p. 2222, Nov. 2018. DOI: 10.3390/ma11112222.

[8] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics", *Journal of Geophysical Research*, vol. 99, no. C5, p. 10 143, 1994. DOI: 10.1029/94jc00572.

[9] H. Kato, A. Yoshizawa, G. Ueno, and S. Obayashi, "A data assimilation methodology for reconstructing turbulent flows around aircraft", *Journal of Computational Physics*, vol. 283, pp. 559–581, Feb. 2015, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2014.12.013.

[10] X. Gao, Y. Wang, N. Overton, M. Zupanski, and X. Tu, "Data-assimilated computational fluid dynamics modeling of convection-diffusion-reaction problems", *Journal of Computational Science*, vol. 21, pp. 38–59, Jul. 2017, ISSN: 1877-7503. DOI: 10.1016/j.jocs.2017.05.014.

[11] X.-L. Zhang, H. Xiao, G.-W. He, and S.-Z. Wang, "Assimilation of disparate data for enhanced reconstruction of turbulent mean flows", *Computers & Fluids*, vol. 224, p. 104 962, Jun. 2021, ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2021.104962.

[12] T. Hayase and S. Hayashi, "State Estimator of Flow as an Integrated Computational Method With the Feedback of Online Experimental Measurement", *Journal of Fluids Engineering*, vol. 119, no. 4, pp. 814–822, Dec. 1997, ISSN: 0098-2202. DOI: 10.1115/1.2819503.

[13] T. Hayase, K. Nisugi, and A. Shirai, "Numerical realization for analysis of real flows by integrating computation and measurement", *International Journal for Numerical Methods in Fluids*, vol. 47, no. 6-7, pp. 543–559, 2005. DOI: 10.1002/fld.829.

[14] N. J. Neeteson and D. E. Rival, "State observer-based data assimilation: A PID control-inspired observer in the pressure equation", *Measurement Science and Technology*, vol. 31, no. 1, p. 014 003, Oct. 2019, ISSN: 0957-0233. DOI: 10.1088/1361-6501/ab40d4.

[15] E. Saredi, N. Tumuluru Ramesh, A. Sciacchitano, and F. Scarano, "State observer data assimilation for RANS with time-averaged 3D-PIV data", *Computers & Fluids*, vol. 218, p. 104 827, Mar. 2021, ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2020.104827.

[16] M. Zauner, V. Mons, O. Marquet, and B. Leclaire, "Nudging-based data assimilation of the turbulent flow around a square cylinder", *Journal of Fluid Mechanics*, vol. 937, Mar. 2022. DOI: 10.1017/jfm.2022.133.

[17] V. Mons, J.-C. Chassaing, T. Gomez, and P. Sagaut, "Reconstruction of unsteady viscous flows using data assimilation schemes", *Journal of Computational Physics*, vol. 316, pp. 255–280, Jul. 2016. DOI: 10.1016/j.jcp.2016.04.022.

[18] L. Lei, D. R. Stauffer, and A. Deng, "A hybrid nudging-ensemble Kalman filter approach to data assimilation. part II: Application in a shallow-water model", *Tellus Series A-dynamic Meteorology and Oceanography*, vol. 64, no. 1, p. 18 485, May 2012, ISSN: 1600-0870. DOI: 10.3402/tellusa.v64i0.18485.

[19] T. Suzuki and F. Yamamoto, "Hierarchy of hybrid unsteady-flow simulations integrating time-resolved PTV with DNS and their data-assimilation capabilities", *Fluid Dynamics Research*, vol. 47, no. 5, p. 051 407, Sep. 2015, ISSN: 1873-7005. DOI: 10.1088/0169-5983/47/5/051407.

[20] L. Euler, "Principes généraux du mouvement des fluides", *Mémoires de l'académie des sciences de Berlin*, pp. 274–315, Jan. 1757. [Online]. Available: https://scholarlycommons.pacific.edu/euler-works/226.

[21] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, 3rd ed. Berlin Heidelberg: Springer-Verlag, 2009, ISBN: 9783540252023. DOI: 10.1007/b79761.

[22] P. W. McDonald, *The computation of transonic flow through two-dimensional gas turbine cascades*. American Society of Mechanical Engineers, 1971, vol. 79825.

[23] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Aug. 2002. DOI: 10.1017/cbo9780511791253.

[24] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes", *Journal of Computational Physics*, vol. 43, no. 2, pp. 357–372, Oct. 1981, ISSN: 0021-9991. DOI: 10.1016/0021-9991(81)90128-5.

[25] A. Harten, "On a class of high resolution total-variation-stable finite-difference schemes", *SIAM Journal on Numerical Analysis*, vol. 21, no. 1, pp. 1–23, Feb. 1984. DOI: 10.1137/0721001.

[26] B. van Leer, "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method", *Journal of Computational Physics*, vol. 32, no. 1, pp. 101–136, Jul. 1979. DOI: 10.1016/0021-9991(79)90145-1.

[27] B. van Leer, "Towards the ultimate conservative difference scheme. II. monotonicity and conservation combined in a second-order scheme", *Journal of Computational Physics*, vol. 14, no. 4, pp. 361–370, Mar. 1974. DOI: 10.1016/0021-9991(74)90019-9.

[28] H. Jasak, "Error analysis and estimation for the finite volume method with applications to fluid flows.", 1996.

[29] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques", *Computers in Physics*, vol. 12, no. 6, p. 620, 1998. DOI: 10.1063/1.168744.

[30] C. Greenshields and H. Weller, *Notes on Computational Fluid Dynamics: General Principles*. Reading, UK: CFD Direct Ltd, 2022.

[31] H. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics*, 2nd ed. Philadelphia, PA: Prentice Hall, Feb. 2007.

[32] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-16874-6.

[33] R. Courant, E. Isaacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences", *Communications on Pure and Applied Mathematics*, vol. 5, no. 3, pp. 243–255, Aug. 1952. DOI: 10.1002/cpa.3160050303.

[34] S. Patankar and D. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", *International Journal of Heat and Mass Transfer*, vol. 15, no. 10, pp. 1787–1806, Oct. 1972. DOI: 10.1016/0017-9310(72)90054-3.

[35] J. Boussinesq, "Essai sur la théorie des eaux courante", *Mémoires présentés par divers savants á l'Académie des Sciences XXIII*, vol. 1, pp. 1–23, 1877.

[36] W. P. Jones and B. E. Launder, "The prediction of laminarization with a two-equation model of turbulence", *International Journal of Heat and Mass Transfer*, vol. 15, no. 2, pp. 301–314, Feb. 1972, ISSN: 0017-9310. DOI: 10.1016/0017-9310(72)90076-2.

[37] D. C. Wilcox, "Reassessment of the scale-determining equation for advanced turbulence models", *AIAA Journal*, vol. 26, no. 11, pp. 1299–1310, 1988, ISSN: 0001-1452. DOI: 10.2514/3.10041.

[38] F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications", *AIAA Journal*, vol. 32, no. 8, pp. 1598–1605, 1994, ISSN: 0001-1452. DOI: 10.2514/3.12149.

[39] B. E. Launder, G. J. Reece, and W. Rodi, "Progress in the development of a reynolds-stress turbulence closure", *Journal of Fluid Mechanics*, vol. 68, no. 3, pp. 537–566, Apr. 1975. DOI: 10.1017/s0022112075001814.

[40] C. G. Speziale, S. Sarkar, and T. B. Gatski, "Modelling the pressure–strain correlation of turbulence: An invariant dynamical systems approach", *Journal of Fluid Mechanics*, vol. 227, pp. 245–272, Jun. 1991. DOI: 10.1017/s0022112091000101.

[41] M. Manolesos, Z. Gao, Z. Xing, M. Panos, and D. Bouris, "Experimental study of the flow past a cube with openings embedded in a turbulent boundary layer", in *10th International Symposium on Turbulence and Shear Flow Phenomena*, 2017.

[42] M. Manolesos, Z. Gao, and D. Bouris, "Experimental investigation of the atmospheric boundary layer flow past a building model with openings", *Building and Environment*, vol. 141, pp. 166–181, Aug. 2018, ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2018.05.049.

[43] N.-P. Pallas, "Υπολογισμός Πιέσεων από Δεδομένα PIV με Χρήση του Αλγορίθμου SIMPLE", M.S. thesis, School of Mechanical Engineering, National Technical University of Athens, Aug. 2020. DOI: https://doi.org/10.26240/heal.ntua.19777.

[44] N.-P. Pallas and D. Bouris, "Calculation of the pressure field for turbulent flow around a surface-mounted cube using the SIMPLE algorithm and PIV data", *Fluids*, vol. 7, no. 4, p. 140, Apr. 2022. DOI: 10.3390/fluids7040140.

[45] M. Fisher, "Background error covariance modelling", in *Seminar on Recent developments in data assimilation for atmosphere and ocean, 8-12 September 2003*, ECMWF, Shinfield Park, Reading: ECMWF, 2003, pp. 45–64. [Online]. Available: https://www.ecmwf.int/node/9404.

[46] T. Suzuki, "Reduced-order kalman-filtered hybrid simulation combining particle tracking velocimetry and direct numerical simulation", *Journal of Fluid Mechanics*, vol. 709, pp. 249–288, Aug. 2012. DOI: 10.1017/jfm.2012.334.

[47] R. N. Bannister, "A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics", *Quarterly Journal of the Royal Meteorological Society*, vol. 134, no. 637, pp. 1971–1996, Oct. 2008. DOI: 10.1002/qj.340.

[48] R. A. Anthes, "Data assimilation and initialization of hurricane prediction models", *Journal of the Atmospheric Sciences*, vol. 31, no. 3, pp. 702–719, Apr. 1974. DOI: 10.1175/1520-0469(1974)031<0702:daaioh>2.0.co;2.

[49] D. Auroux and J. Blum, "A nudging-based data assimilation method: The Back and Forth Nudging (BFN) algorithm", vol. 15, no. 2, pp. 305–319, 2008. DOI: 10.5194/npg-15-305-2008.

[50] C. Wunsch, *Discrete inverse and state estimation problems*, en. Cambridge, England: Cambridge University Press, Jun. 2006.

[51] X. Zou, I. M. Navon, and F. X. Ledimet, "An optimal nudging data assimilation scheme using parameter estimation", *Quarterly Journal of the Royal Meteorological Society*, vol. 118, no. 508, pp. 1163–1186, Oct. 1992. DOI: 10.1002/qj.49711850808.

[52] D. R. Stauffer and J.-W. Bao, "Optimal determination of nudging coefficients using the adjoint equations", *Tellus A*, vol. 45, no. 5, pp. 358–369, 1993, ISSN: 1600-0870. DOI: 10.1034/j.1600-0870.1993.t01-4-00003.x.

[53] G. A. Sod, "A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws", *Journal of Computational Physics*, vol. 27, no. 1, pp. 1–31, Apr. 1978, ISSN: 0021-9991. DOI: 10.1016/0021-9991(78)90023-2.

[54] N. Ji, "Comparison and Analysis of Different Numerical Schemes in Sod's One-dimensional Shock Tube Problems", *Journal of Physics: Conference Series*, vol. 1550, p. 032 049, May 2020. DOI: 10.1088/1742-6596/1550/3/032049.

[55] OpenMP Architecture Review Board, *OpenMP application program interface version 4.5*, Nov. 2015. [Online]. Available: https://www.openmp.org/wp-content/uploads/openmp-4.5.pdf.

[56] W. J. M. Rankine, "XV. On the thermodynamic theory of waves of finite longitudinal disturbance", *Philosophical Transactions of the Royal Society of London*, vol. 160, pp. 277–288, 1870. DOI: 10.1098/rstl.1870.0015.

[57] A. Jameson, "Aerodynamic design via control theory", *Journal of scientific computing*, vol. 3, no. 3, pp. 233–260, 1988.

[58] K. C. Giannakoglou and D. I. Papadimitriou, "Adjoint methods for shape optimization", in *Optimization and Computational Fluid Dynamics*, D. Thévenin and G. Janiga, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 79–108, ISBN: 978-3-540-72153-6. DOI: 10.1007/978-3-540-72153-6_4.

[59] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain", USA, Tech. Rep., 1994.

[60] D. P. Rodgers, "Improvements in multiprocessor system design", *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3, pp. 225–231, Jun. 1985. DOI: 10.1145/327070.327215.

[61] M. Raffel, C. E. Willert, F. Scarano, C. J. Kähler, S. T. Wereley, and J. Kompenhans, *Particle Image Velocimetry*. Springer International Publishing, 2018. DOI: 10.1007/978-3-319-68852-7.

[62] C. M. Rhie and W. L. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation", *AIAA Journal*, vol. 21, no. 11, pp. 1525–1532, Nov. 1983. DOI: 10.2514/3.8284.

[63] P. P. Walatka, P. G. Buning, L. Pierce, and P. A. Elson, "PLOT3D user's manual", Tech. Rep. NAS 1.15:101067, Mar. 1990.

[64] B. W. van Oudheusden, "PIV-based pressure measurement", *Measurement Science and Technology*, vol. 24, no. 3, p. 032 001, Jan. 2013. DOI: 10.1088/0957-0233/24/3/032001.

[65] E. Gunaydinoglu and D. F. Kurtulus, "Pressure–velocity coupling algorithm-based pressure reconstruction from PIV for laminar flows", *Experiments in Fluids*, vol. 61, no. 1, Nov. 2019. DOI: 10.1007/s00348-019-2831-1.