Spring 1-1-2013

Fast CFD Simulation Method for Indoor Environment Modeling

Haidong Wang
University of Colorado at Boulder, haidong.wang@colorado.edu

FAST CFD SIMULATION METHOD FOR INDOOR ENVIRONMENT MODELING

By

Haidong Wang
B.S., Tianjin University, 2006
M.S., Tianjin University, 2008

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirement for the degree of
Doctor of Philosophy
Department of Civil, Environmental and Architectural Engineering
2013

This thesis entitled:

Fast CFD Simulation Method for Indoor Environment Modeling

written by Haidong Wang

has been approved for the Department of Civil, Environmental and Architectural Engineering

____

Zhiqiang (John) Zhai

____

Moncef Krarti

Date______________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

ABSTRACT

Wang, Haidong (Ph.D., Department of Civil, Environmental and Architectural Engineering)

Fast CFD Simulation Method for Indoor Environment Modeling

Thesis directed by Associate Professor Zhiqiang (John) Zhai

Computational Fluid Dynamics (CFD) has been becoming an important tool in building environment and airflow modeling. Well validated CFD model can provide more detailed and informative results than other computer models and physical experiment. However, the significant computational cost of CFD has always been the main factor that restricting its broader application.

This study reviews the various means of reducing the computational cost of CFD in literature. With considering the characteristics and result accuracy requirement in indoor environment quality (IEQ) study, the most important factors that affect the computing cost are identified: grid resolution, turbulence modeling and velocity-pressure decoupling algorithm. Theoretical analysis is conducted to find out the possibilities of optimizing grid resolution and simplifying turbulence modeling. Numerical experiments using measurement data in literature is conducted to validate the hypothesis from theoretical analysis.

Transient airflow simulation is a topic that draws not as much attention in this area, but is of great importance for building indoor environment study. Building is an open system interacting with the continuously changing weather condition. Most of the airflow and heat transfer phenomenon in building is transient with time-dependent boundary condition and dynamic thermal response. This study reviews the algorithms for transient airflow simulation, identifies the potential candidates and compares the performance of different algorithms. Based on the theoretical analysis of pros and cons of each algorithm, a new algorithm called semi-Lagrangian PISO (SLPISO) is brought out. Computer code is developed to validate and test the performance of SLPISO algorithm for typical indoor airflow case. The algorithm can significantly reduce the computational intensity of PISO and improves the accuracy of FFD, but for problems with velocity and pressure closely linked, relatively small time step has be to adopted in order to get accurate transient CFD solution.

DEDICATION

To my parents, Rukun Wang and Xiulian Wang, and my sister, Caihong Wang, for their care, mentoring, endless love and unconditional support.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the guidance and inspiration of my committee members, the support of my family and wife, and the encouragement of my friends.

I would like to express my deepest gratitude to my advisor, Dr. John Zhai, who provided inspiriting guidance, constant support, patience and encouragement during this challenging journey. This work would not have been possible without his dedication to supervising me and pointing me in the right direction. I have also appreciated his personal help during my study at the University of Colorado Boulder and his advice on my career development.

I would like to thank Dr. Harihar Rajaram for providing a new perspective to my research direction, especially for introducing the PISO algorithm to me. Thanks to all my dissertation committee members, Professor Michael Brandemuehl, Professor Moncef Krarti, Professor Sedat Biringen and Professor Harihar Rajaram for providing scientific advice and innovative ideas and approaches to this research.

My special thanks to Dr. Wangda Zuo for help on exploring the fundamentals, especially Fast Fluid Dynamics (FFD), and for providing the code of FFD from his dissertation and helping on programming.

My colleague and friend James McNeil deserves my sincere thanks. I am lucky enough to have him in the same research group with shared research interests; he is dedicated to research, full of enthusiasm and always ready to help. His attitude toward scientific research and smartness in solving problems set a good example for me in ways I will always remember.

My same thanks to my other colleagues Saleh Al-Saadi and Xiangli Li for the great help in everyday life during the journey of this dissertation, who are always ready to provide advice and help. It feels good to be in such a warm family. My thanks also go to my colleague Xiang Liu for his support and help during my early years of starting the Ph.D. study.

There is no vocabulary in this world to tell my parents about how much I love and appreciate them for raising me. I owe them everything in my life.

Last but not least, I would like to thank my wife Qiong Guo, who has always been encouraging me to finish this dissertation. Her care, love and encouragement are the source of my strength and success. I am blessed to get to know and fall in love with her in this beautiful campus and experience all the drama together.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

NOMENCLATURE

CFD	Computational Fluid Dynamics
PDE	Partial Differential Equation
IEQ	Indoor Environment Quality
RANS	Reynolds-Averaged Navier-Stokes
PISO	Pressure-Implicit with Splitting of Operators
SLPISO	Semi-Lagrangian PISO
FFD	Fast Fluid Dynamics (Operator Splitting CFD)
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
RMSE	Root Mean Square Error
NC	Natural Convection
FC	Forced Convection
MC	Mixed Convection
NV	Natural Ventilation
DV	Displacement Ventilation
DNS	Direct Numerical Simulation
LES	Large Eddy Simulation
DES	Detached-Eddy Simulation
QUICK	Quadratic Upstream Interpolation for Convective

	Kinematics
CDS	Central Differencing Scheme
ROM	Reduced Order Model
POD	Proper Orthogonal Decomposition
DOF	Degree of Freedom
∇ Gradient	$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}\right)$
$\nabla \cdot$ Divergence	$\nabla \cdot u = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z}$
∇^2 Laplacian	$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$

CHAPTER 1
INTRODUCTION

1.1 Background and Significance

Computational Fluid Dynamics (CFD) is the analysis of systems involving fluid flow, heat transfer and associated phenomena such as chemical reactions by means of computer-based simulation [1]. It uses numerical methods to solve the discretized formulation of the Navier-Stokes equation, which is a mathematical description of fluid flow and heat and mass transfer phenomena. CFD is a powerful tool that has been used in many aspects of scientific research and engineering investigation such as: aerospace vehicle and airplane design; atmospheric, ocean and weather modeling/prediction; indoor and outdoor environment quality study, and so on.

CFD has been playing an important role in building indoor environment quality (IEQ) study since its first application in this area decades ago [2]. Compared to the traditional methods of airflow study such as on-site measurement and full scale and reduced scale experimentation, the cost of CFD is relatively lower and the time to build up a CFD model is shorter than to set up an experiment. CFD can also provide more detailed information than physical experimentation and has better flexibility for changing configuration for parametric studies.

Well-validated CFD models can be applied to various areas of building environment research. Such models are useful for schematic building design [3, 4] and pollutant and contaminant transport [5, 6], as well as real-time building control and monitoring [7]. CFD can be integrated into hourly based building energy simulation models [8, 9] and zonal airflow models [10] to obtain more informative results and improve the accuracy of these simulations. Such integration has been applied to building energy simulation software such as Design Builder [11] and TAS [12] as well as multi-zone airflow and contaminant modeling software such as CONTAM [13].

However, the significant computing cost of CFD has become the main reason that restricts its broader and better application, for example, in building emergency management, early stage architectural design, and integration with building energy simulation software. In such situations, many different parameter values and component interactions may need to be evaluated in a timely manner and hence require real-time or faster-than-real-time results. The fast evolution of computer hardware and scientific computing technologies, such as parallel computing, cloud computing and CUDA (Compute Unified Device Architecture) programming with GPU (Graphical Processing Unit) [14], provide good alternative solutions for this problem. However, due to the high cost of equipment, maintenance and operation, it is still a luxury for building environment quality-related simulations and does not fundamentally provide a solution based on the current level of computer technology.

Moreover, most of the CFD models being applied to indoor environment quality research, if not all, focus on the steady-state simulation capability, without investigating the accuracy of transient simulation. Therefore, the performance of existing models, when being applied to studying the transition of airflow from one state to another, is not very well validated and lacks credibility. The well validated CFD models for steady-state calculation, although providing a decent prediction on the steady-state flow field, are not necessarily accurate on the intermediate result for the airflow transiting from one state to another. These commonly used CFD models for steady-state airflow simulation, once adapted to a transient problem, will require thousands of iterations within each time step, making the acquisition of CFD even more computationally intensive. However, the potential applications of a fast CFD technique, such as building emergency management and integrated building energy simulation, primarily depend on the accuracy of a time-dependent solution. Hence research on the transient features of CFD models and investigating the advanced techniques of fast transient simulation are also of great interest.

1.2 Research Approach

Fast simulation technologies in literature are reviewed and the most promising algorithms for the benefit of computing speed are identified, which provides an overview for the direction of this research. Theoretical analysis and exploration are conducted to pinpoint the innovative means of further reducing computing cost. Extensive numerical experiments making use of experimental data with typical building configurations in the literature are conducted to test and validate the theoretical findings and hypothesis.

The building indoor environment shares many unique common characteristics compared to other areas of application. Most indoor airflows are turbulent with a relatively low Reynolds number and heat transfer induced convection is usually weak. Based on the fundamental difference of driving force, indoor airflow can be categorized into three different mechanisms: natural convection (NC), in which airflow is driven by a temperature difference induced buoyancy force; forced convection (FC), where airflow is driven by external forces such as mechanical (fan) or natural (wind) forces; and mixed convection (MC), a combination of the two. These common characteristics and mechanisms provide the fundamental similarities. Proper strategies specifically designated to these characteristics thus can be developed to improve the speed of CFD models.

Investigation indicates that the grid resolution is one of the main factors that determine the computing speed. A coarse grid can save computing cost significantly. The computing time is proportional to the total grid number in CFD even without considering that more iteration will be required for utilization of the fine grid. Turbulence modeling is another important topic. Different turbulence modeling techniques will require different levels of grid resolution and time step size. Based on the objective of this research, simplified turbulence model is the main focus and will be primarily investigated.

The computer tools used in this research will include both commercial CFD software and self-developed computer code. The commercial code Phoenics [15] is used in this research. It provides good flexibility for manipulating grid sizes and distributions. The embedded turbulence models, algorithms and numerical schemes are well validated; CFD algorithms not available in commercial code are implemented using a C program. Such a supplementation offers a full coverage of computer tools to test hypotheses and new algorithms brought out through this research.

1.3 Objective and Outline

The objective of this study is to identify and explore computationally inexpensive CFD models, by means of adopting coarse grids, simplifying turbulence models and optimizing the algorithms for numerically solving Navier-Stokes equations. Research will focus on both steady-state airflow and transient airflow simulation. The projected outcome of this study is the provision of a guideline for building IEQ simulation with simplified CFD models, applicable for both steady state and transient simulation.

The research will mainly focus on the CFD simulation of airflow movement and heat transfer phenomenon occurs in typical indoor environment, specifically regular size single room. The velocity and temperature flow field simulation will be investigated.

Chapter 1 is a statement of the significance of such research. It reveals the urgent demand for fast CFD simulation. The general research approach and objective are introduced.

Chapter 2 thoroughly reviews the fast simulation techniques for indoor environment and airflow study in the literature. The potential candidates in the literature could be both building and indoor environment related and in other areas of study such as aerospace science and engineering, mechanical engineering, computer and numerical science, and so on. Differences and similarities of the applied areas of these different techniques to building and indoor environment will be analyzed to discover their possibilities.

Chapter 3 summarizes the fundamentals of CFD being used in this study, including governing equations, turbulence modeling and numerical techniques. This provides the background knowledge for theoretical accordance and further improvement that can possibly be made.

Chapter 4 presents the result of optimizing the trade-off between grid resolution and result accuracy by taking advantage of coarse grid. It is found that the grid resolution is the most direct factor that affects the computational cost of a CFD simulation. Theoretical analysis is conducted to investigate the possibility of adopting coarse grid to reduce computational cost. The hypothesis is validated under the regime of existing turbulence models of RANS simulation.

Chapter 5 further explores the coarse grid application coupled with simplified turbulence models. The complexity of a turbulence model is another factor that inflates the CFD computing cost. Simplified turbulence modeling is proposed and validated through extensive numerical experimentation.

Chapter 6 investigates the transient simulation features of existing models. Based on the performance of different strategies, it is found that the pressure-velocity decoupling algorithm is a key factor on the transient simulation speed. The chapter further proposes a new algorithm that takes advantage of different existing algorithms. Validation results show that this new algorithm is quite promising and can provide accurate and fast time-dependent results when applied to transient simulation.

Chapter 7 is the conclusion and discussion part of this research, focusing on the main contribution of this study and work that has been done. It also brings out possible directions that are worthy of further investigation for the benefit of IEQ study-oriented CFD speed advancement.

CHAPTER 2
LITERATURE REVIEW OF FAST AIR SIMULATION TECHNIQUES

2.1 Overview of Fast Airflow Simulation Techniques

The voice for fast simulation results has been leading to various means and efforts to make CFD modeling less computationally intensive. Some of the efforts are: development of simplified turbulence models such as zero-equation models [16]; reform of solution algorithms for pressure-velocity decoupling such as PISO [17] and projection methods [18]; utilization of coarse grid [19, 20]; employment of cutting edge computer hardware technology such as GPU [14] and parallel/multi-processor supercomputers. Although the rapid development of computer hardware provides more powerful computing capacity, it does not solve the problem fundamentally.

Large computer clusters with the capacity to provide enough speed for practical engineering use are usually expensive in terms of price and maintenance.

Aside from CFD models, network models such as multi-zone [21] and zonal [22] models are developed as a much faster algorithm than CFD in building airflow simulation. Multi-zone models treat each zone as a node and assume well-mixing for each zone, and thus they cannot provide the informative results that CFD does; the zonal model is an intermediate model between multi-zonal models and CFD and thus can provide more informative results than multi-zone models. However, to make a reasonable division of the domain, it is necessary to know the airflow patterns and contaminant concentrations in priori.

Reduced order modeling (ROM) [23] has the potential of quantitatively describing the dynamics of systems at a computational cost much lower than the original numerical model. ROMs provide a means by which system dynamics can be readily interpreted. Next, these different models applicable in IEQ studies will be reviewed.

2.1.1 Multi-Zone and Zonal Models

Multi-zone and zonal models are network airflow models. They purposely provide openings, flow paths and interconnections between individual zones to model the effect of interactions. Multi-zone indoor environment quality (IEQ) modeling has been available as a research and analysis tool for over 20 years [24].

Multi-zone building airflow analysis may be based on either nodal or port-plane idealizations of integrated building/HVAC systems. The nodal approach, which evolved through an adaptation of nodal methods of electric resistance network analysis, idealizes the building system as collections of zones and duct junctions and associated node pressures within them linked by discrete flow-limiting elements [25].

The fundamental basis of a multi-zone model is a combination of mass conservation and mechanical energy conservation (expressed as the Bernoulli equation). Take a certain zone, for example. Assume that there is air flow inlet and outlet, with the mass and mechanical energy conservation expressed as

Mass conservation

$$\frac{dm}{dt} = \rho_i A_i v_i - \rho_o A_o v_o \tag{2.1}$$

Mechanical Energy Conservation

$$\left(P_i + \frac{1}{2}\rho_i v_i^2 + \rho_i g z_i\right) = \int_{in}^{out} \frac{\partial v}{\partial t} ds + \left(P_o + \frac{1}{2}\rho_o v_o^2 + \rho_o g z_o\right) + \Delta P \tag{2.2}$$

Concentration

$$V\frac{dC}{dt} = C_i A_i v_i - C_o A_o v_o \tag{2.3}$$

The mass conservation equation applies to air mass as well as any species transport with airflow, so the concentration equation has the same format. The mechanical energy conservation expression is the transient version of the Bernoulli equation, where ΔP denotes the pressure loss/gain of the connection. For example, a duct-fan system enhancing the air flow will give a positive ΔP, while a crack with resistance will exert a negative ΔP to the equation. The temperature is usually not calculated [26] so the thermal energy equation is not included. Temperature is usually specified by schedules to each zone.

Equation sets of interactive zones eventually form linear algebraic equations. Most multi-zone programs use the Newton-Raphson method [27] or its variants to solve the system equations. It is fundamentally an iterative method. The convergence of this iterative process is sensitive to the initial guess.

Multi-zone models can efficiently simulate the whole building air change and mass transfer with very low computing intensity, plus the wide availability of prevalent multi-zone simulation software, such as AIRNET[28], CONTAM [29] and Commis [27], make it popular in various aspects of building environment research such as evaluation of ventilation systems [30], epidemic disease transmission [31], and contaminant source tracking [32].

Zonal models have been developed over the last 25 years. They are based on an approximate partitioning of a room into a number of subzones and have been used for modeling indoor and outdoor environments [22]. They are an intermediate approach between computational fluid dynamics and simple nodal models. They have the ability to take into account various phenomena ignored by one-node and multi-zone models, such as temperature and contaminant distributions, thermal integration with cold façade, draft, asymmetric thermal radiation, and cold or hot floor surfaces. Zonal models are a promising way to predict air movement in a room with respect to comfort conditions and gradients of temperature because they require extremely little computer time and could easily be included in multi-zone air movement models.

However, to make a reasonable division of the domain, this approach needs complementary information and models to define flows. It is necessary to know the airflow patterns and contaminant concentrations in priori. Ideally, zonal models can calculate the same level of details as CFD, provided that the subzone used in this model has a similar resolution as a CFD grid. But study [20] shows that CFD is a better suited method to predict airflow details; various formulations of zonal models cannot provide satisfactory predictions for isothermal flow. CFD can be integrated into multi-zone models to improvement prediction accuracy [10], which works better than zonal models.

2.1.2 Computational Fluid Dynamics

The Computational Fluid Dynamics (CFD) method numerically solves Navier-Stokes equations, which are a set of partial differential equations (PDEs) used to mathematically describe flow phenomena. CFD can be used to describe and predict the flow phenomena in a predefined geometry with initial and boundary conditions.

Discretization of PDEs is utilized to convert continuous equations sets into numerical equation systems. Two of the commonly used methods are Finite Volume Method (FVM) and Finite Difference Method (FDM). FVM discretization is based upon an integral form of the PDE to be solved for a given finite volume, e.g. a cell. The variables are typically places at cell centers, except for staggered grids, where vector variables are placed at face center. Conservation of mass, momentum, and energy are ensured at each cell/finite volume level. FVM is a common approach used in CFD codes [33]. FDM discretization is based upon the differential form of the PDE to be solved, where an approximate difference formula that can generally be derived from a Taylor series expansion is used to represent derivatives.

Most building indoor airflow is turbulent, and turbulence modeling is a key issue in most CFD simulations, including indoor airflow simulation. Turbulent flows are characterized by a wide range of length scales. Based on the scales of modeled turbulence, Direct Numerical Simulation (DNS) [34], Detached Eddy Simulation (DES), Large Eddy Simulation (LES) [35] and Reynolds-Averaged Navier-Stokes (RANS) [36]-based turbulence models are used in CFD. DNS is a simulation that resolves turbulent flow at the smallest time and length scales. This is so far too expensive for engineering application. LES solves for the large eddies in a calculation and implicitly accounts for the small eddies by using a subgrid-scale model. DES is a hybrid of LES and other models such as RANS [37] for near-wall regions. RANS-based models solve the statistically averaged format of the Navier-

Stokes equation [38], which represents the mean flow. RANS model can be further divided to eddy viscosity models according to Boussinesq approximation [39] and Reynolds Stress Model (RSM).

Despite the challenges associated with turbulence modeling, the RANS approach has become very popular in modeling airflows in enclosed environments due to its significantly small requirements on computer resources and user skills [40]. Different RANS models have different performance on different airflow mechanisms [41, 42], and none of them is superior over other models on all types of flows in the enclosed environment. Study shows that a RNG k-ε model [43] has the best overall performance among RANS models [40, 44]. Zero-equation models are the simplest eddy viscosity model based on a mixed length hypothesis [45] which solves zero PDE in addition to the RANS. The empirical formulation of the zero-equation model developed specifically for indoor environment modeling [46] has been well validated and widely used for different indoor environment studies with acceptable accuracy and significant reduction in saving time.

Most of the studies and simulations mentioned above, if not all, are in the interest of a steady-state situation. The mean flow values are not dependent on time. This is no doubt useful for studying the building indoor environment. However, in the potential applicable area of fast CFD simulation, such as hourly building energy modeling, building system control, and emergency management, it is also of great interest to understand the transitional process of airflow and/or concentration from one condition to another. Transient and steady-state simulation performance of CFD is related to the velocity-pressure decoupling algorithms.

A Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) [47] algorithm and different modified versions of such algorithms, such as SIMPLER (SIMPLE Revised) [48], SIMPLEC (SIMPLE Consistent) [49], SIMPLEM (SIMPLE modified) [50] and SIMPLEX (SIMPLE extrapolation) [51], are mainly used for steady-state simulation and have been successfully applied to popular commercial CFD codes such as Fluent, Phoenics, Star-CD and free code such as OpenFoam. They are applicable to transients simulation as well by splitting a simulation into each time step and executing iterations in each time step to obtain converged solutions.

Another group of algorithms designed for transient CFD simulation application include PISO (Pressure-Implicit with Splitting of Operators) [17] algorithm, projection method [52], and fractional time step methods [53], which require no iteration within the selected time step. For steady-state simulation, PISO has no advantage over SIMPLE in computing cost when scalar such as temperature is closely linked to velocity [54]. PISO can be much faster for calculating isothermal flow [55]; as for a velocity-temperature strongly coupled problem, PISO can be four times slower, and gives correct solutions only when the time step is small. An improved PISO for buoyancy driven flow [56] may stress this problem.

Foster et al. [57, 58] implemented the projection method [18] to simulate the 3D motion of hot, turbulent gas using a relatively coarse grid. Stam [59] proposed using semi-Lagrangian advection and fast Fourier transformation to speed up the computation to a real-time or faster-than-real-time level. Zuo et al. [60] gave this algorithm a name, Fast Fluid Dynamics (FFD), and first applied this operator splitting algorithm to indoor environment modeling, improved the sequence of operators, tested higher orders of differencing schemes, and evaluated the accuracy levels.

In general, CFD is a much more complex model than network models to simulate building ventilation and indoor environment problems, in addition to being more computationally intensive. But the quantitative description of flow and more detailed information makes it still popular in this area. Recent developments in computing hardware for flow simulations and methodological trials have already demonstrated the great potential of bringing CFD simulation to real-time and even faster than real-time levels.

2.1.3 Reduced Order Model

CFD models are computationally expensive to solve compared to lumped parameter models such as multi-zone and zonal models. They usually involve large scale computational systems resulting from discretization of partial differential equations (PDEs). For a 2-equation turbulence model in 3 dimensions, the finite volume method produces 7 degrees of freedom (DOF) per grid cell (*P, u, v, w, k, ε,* and *T*). All of the DOF solved for are necessary to model the airflow and heat transfer, but the key quantities of interest, i.e, velocity and temperature in the indoor environment study to investigate airflow and thermal characteristics of a room, use only a small portion of the total model DOF.

Reduced-order models (ROMs), allowing the systematic generation of cost-efficient representations of such large-scale systems, are generally models to provide quantitatively accurate descriptions of the dynamics of systems at a computational cost much lower than the original numerical model and to provide a means by which system dynamics can be readily interpreted [61]. In many unsteady CFD applications, a small number of inputs and outputs of interest can be identified, and computationally efficient reduced-order models can be obtained that preserve the desired input–output mapping [62].

For non-linear, time-dependent problems in CFD, ROMs are typically based on the proper orthogonal decomposition (POD) [63] combined with Galerkin projection [64]. There are many successful applications of this model in related areas such as building contaminant transport [65] and incompressible jet flow [66]. A successful application of this method on data center thermal management [67], as in Figure 1, was able to reduce numerical models containing 103-105 DOF down to less than 20 DOF while still retaining greater than 90% accuracy over the domain.

The POD uses principle component analysis to decompose large systems into a series of fundamental modes; the Galerkin projection method is to project the governing equations onto modal subspace.

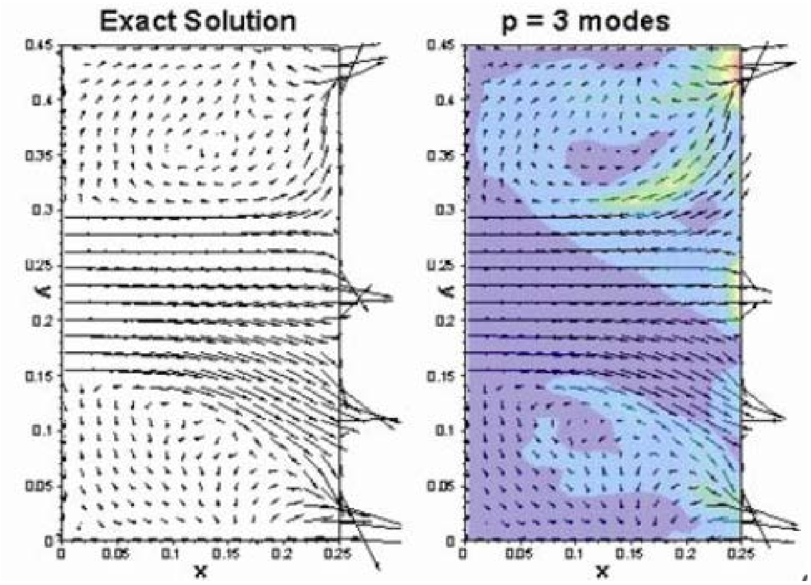


Figure 1. An example of successful application of reduced order modeling with POD to a flow system [67]. The contour indicates an error range of 0-0.08.

There are also two other mainly used reduced-order modeling approaches, called the Volterra theory of nonlinear systems [68] and Harmonic Balance (HB) formulation [69], which are prevalent approaches with many successful applications to fluid dynamics problems [61].

However, there are several challenges that need to be overcome before ROM methods can be routinely applied to practical problems in the following three categories: construction, generality, and accuracy assessment [61]. While most ROMs can operate in near real-time, their construction can however be computationally expensive as it requires accumulating a large number of system responses to input excitations. ROMs usually lack robustness with respect to parameter changes and therefore must often be rebuilt for each parameter variation [70]. Accuracy assessment is also an important topic dealing with the ability to evaluate the quality, inherently or a posteriori, of ROM solutions.

Specifically in indoor environment related areas, the ROMs serve building environment control purposes and building control systems well [71], but the character of ROMs and these challenges constitute a great barrier for its application in indoor environment simulation. A simple input-output relationship model does not provide informative data for thermal and airflow environment analysis.

2.1.4 Conclusive Summary

Different models and techniques in modeling indoor airflow are reviewed in this section to dig into the potentials of finding out a proper strategy for fast indoor environment airflow modeling. It shows that CFD is one of the most popular methods applied to indoor environment quality study. Various other types of models such as zonal and multi-zone models are dependent on CFD to provide higher levels of details in order to make better predictions. Despite the relatively high cost of computing time, which is being overcome by the rapid progress of both computer hardware technology and improvements on methodology and algorithms, CFD can provide the most informative data that is useful for various area applications of building indoor environment quality study.

2.2 Potentials of CFD Speed Advancement

2.2.1 Coarse Grid Trade-Off

While the number of floating-point operations is proportional to the number of grid points in CFD [72], grid resolution directly determines the CFD speed. Although a fine grid shall be employed in most CFD studies to obtain grid-independent solutions that can be analyzed further [73], information from all grid points used in a CFD (usually in the magnitude of millions) may not be necessary for the analyses. Computational results on a much coarser grid resolution, in most scenarios, will be adequate for macro or meso scale flow pattern analysis.

Coarse grid CFD, although slightly less accurate than that of very fine grids, which level out most discretization caused numerical error, has been applied to the problems [74-76] where accuracy is not the main concern. Not only can coarse grid decrease the number of floating point operation, it also cut the necessary iteration numbers to achieve convergence comparing to very fine grid.

However, the credibility of CFD simulation resulting with coarse grids has always been questioned due to the unknown error scales that coarse grids have brought. This leads to an urgent necessity of understanding and quantifying the trade-off between CFD grid resolution and simulation accuracy. Guidelines for selecting appropriate CFD grids based on the accuracy requirements of a simulation for realistic building indoor airflow conditions is also of great importance. In addition, grid distribution, assuming total grid points are the same, does not have significant impact on CFD speed, but will affect the accuracy of predicted results. The guidelines on optimization of grid distribution will also be helpful.

2.2.2 Turbulence Model Simplification

Turbulence treatment is another important factor that determines the speed of CFD. Different turbulence modeling strategies have different requirements on the resolutions of grid and time step points. RANS-based turbulence models are of great interest to this research, as they provide an economical way to simulate the mean property of flow. In most engineering applications such as IEQ modeling, time-averaged flow properties from RANS solutions can mostly provide adequate information for parametric analysis.

There is no single universally accepted turbulence model that works for all flows and all regimes; therefore, numerous turbulence models have been developed over the past decades. Studies show that specifically for enclosed environment modeling, RNG k-ε model overall has the best performance [40, 44]. However, these types of two-equation models have additional PDEs added to the governing equation set to calculate turbulence properties; hence, they increased the computational intensive. The turbulence property, specifically the turbulence viscosity here, is solely used for the closure of the PDE system, and is not a necessity of IEQ study under most circumstances.

This part of computational costs can be further deducted through using empirical algebraic equations to calculate the turbulent viscosity, such as zero-equation models [46]. It can be further simplified by using a constant to represent such a property, as long as such constant values can provide a good representation of turbulent viscosity properties. This provides the simplest version for modeling of turbulence. This constant viscosity model has been applied to some case studies [77, 78] and shows acceptable results for IEQ study. Nevertheless, the proper viscosity constant applicable to different circumstances, e.g., different flow mechanisms, Reynolds number, complexity of modeled space and grid resolution, is still unknown. A general rule of thumb for utilization of constant viscosity will be able to help improve the performance of such simplified turbulence models.

2.2.3 Velocity-Pressure Decoupling Algorithm

The previous section on CFD methods has already revealed that a velocity-pressure decoupling algorithm is critical for solving highly coupled Navier-Stokes equations. The SIMPLE algorithm and its variants are prevalent for commonly used CFD code, which requires iterations for steady state flow simulation, as well as transient flow simulation within each time step.

Within each iteration, numeric operation is conducted for a large data set that contains the information for every grid node. This makes SIMPLE algorithm and its variants, the mostly used algorithms for CFD methods, not computationally economical.

PISO is a non-iterative algorithm developed for transient flow simulation, and thus provides great potential for decreasing computational intensity. It literally reaches the exact solution of the discretized equation through two stages of predictor-corrector and is much faster than any iteration-based algorithm such as SIMPLE.

Besides the coupling between velocity and pressure, the velocity components along each direction are also high coupled in a momentum equation. Semi-Lagrangian scheme [79] used in FFD not only overcomes such coupling but also escapes the restriction of CFL condition [80] on a time step, which is especially beneficial for transient flow simulation. This will be discussed in the transient airflow simulation part.

2.3 Conclusions

This review reveals that CFD methods are the most accurate and informative way of predicting and simulating the airflow formulation in building environments. However, the computational intensive is the main reason that restricts its broader and better application in various areas.

The potentials for decreasing the computing cost of CFD methods are identified. It is found that the grid resolution, turbulence model and equation-solving algorithm are closely relevant to computing speed. Computing operation is directly related to the total grid number in a CFD model; therefore, coarse grid, although it brings in numerical truncation error and has impact on the accuracy of the CFD solution, can save computing cost significantly.

Turbulence modeling is another important factor that influences CFD speed. Simplified turbulence models may provide precise predictions for IEQ study, yet the detailed rule of thumb needs to be developed for the better utilization of these models.

Iteration impacts the computing speed of CFD. PISO and projection methods provide non-iterative alternatives to solve governing equations for velocity-pressure decoupling algorithms. Semi-Lagrangian advection scheme overcomes the restriction of the CFL condition and weakens the coupling between equations. All of these algorithms and schemes together will provide great potential for developing a fast version of the CFD model.

CHAPTER 3 FUNDAMENTALS OF COMPUTATIONAL FLUID DYNAMICS

To explore the potentials of computational speed advancement for CFD methods, it is important to understand the fundamentals. Navier-Stokes equation solution techniques will be covered in this chapter. According to the previous review, many factors are critically related to the objective of this research, which are: grid resolutions; turbulence models; iterations and velocity-pressure decoupling algorithms; and coupling between velocity components. This chapter will accordingly discuss the fundamentals of CFD, with an emphasis specifically on these above-mentioned aspects.

3.1 CFD and Turbulence Modeling

3.1.1 Governing Equations

CFD is fundamentally based on governing equations of fluid dynamics [81]. Governing equations (Navier-Stokes equations, N-S equations) are a set of PDEs that provide mathematical descriptions of flow and heat transfer phenomena. The continuity equation, momentum equation, energy equation (when temperature difference presents) and species transport equation (when concentration difference presents) constitute the full set of N-S equations.

Since in IEQ study, due to the relatively low air velocity, airflow is treated as incompressible, only incompressible format of N-S equation will be discussed. There are significant commonalities between these various equations. In typical indoor environments, velocity is low and air is considered to be incompressible. The general time-dependent governing equation, under Cartesian coordinates, expressed through Einstein notation [82] is:

$$\frac{\partial \phi}{\partial t} + \frac{\partial \left(u_j \phi \right)}{\partial x_j} = \Gamma_\phi \frac{\partial^2 \phi}{\partial x_j^2} + S_\phi \tag{3.1}$$

For mass conservation (continuity), momentum conservation, energy and species transport, the corresponding variables and coefficients in the general form of N-S equation are summarized in

Table 1.

Table 1. General form of governing equations for incompressible flow

Variable	Continuity	Momentum	Energy (temperature)	Species (concentration)
ϕ	1	u_i	T	C
Γ_ϕ	0	ν	α	D
S_ϕ	0	$-\frac{1}{\rho}\frac{\partial P}{\partial x_i} + S_{F,i}$	S_T	S_C

Here $i, j = 1, 2, 3$, u_i is the i_{th} component of the vector (velocity component along x, y and z direction), where T and C are temperature and species concentration. ν is kinematic viscosity of fluid, α is thermal diffusivity, and D is the diffusivity of species. P is the pressure and S_ϕ is the corresponding source term (vector component or scalar).

3.1.2 Turbulence Modeling

Turbulence is associated with the existence of random fluctuations in the fluid. This behavior can be exemplified by a typical point velocity measurement as a function of time at some location in the turbulent flow as in Figure 2. The velocity can be decomposed into a steady mean value $\bar{u}$ with a fluctuating component.

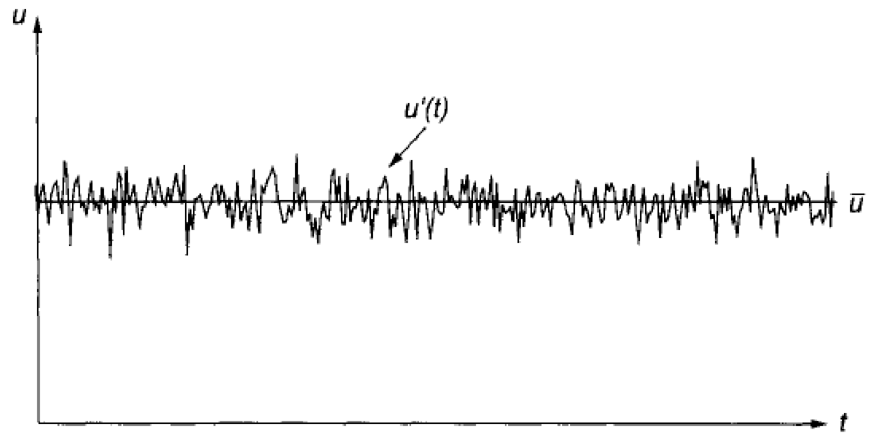


Figure 2. Velocity fluctuation with time at some point of turbulent flow[81]

Almost all room airflows are turbulent [83], which makes it a big challenge to resolve the time-dependent Navier-Stokes equations all the way down to the smallest scales. Very fine numerical resolution is required to capture all the details of indoor turbulent flow if Direct Numerical Simulation (DNS) is used. Since it is unlikely to be practical or inexpensive to simulate turbulence directly for typical indoor airflow, use of turbulence models to simulate the mean flow is popular. This simulation technique is normally referred as Large-Eddy Simulation (LES), Detached-Eddy Simulation (DES) and Reynolds Averaged Navier-Stokes (RANS) equations, based on the computing cost from high to low. As a relatively fast model, RANS-based turbulence models are widely used in building environment areas and are well-validated.

In the Navier-Stokes equation, the instant velocity component can be represented by a Reynolds-averaged (mean) velocity plus the fluctuating component. The final equation of such mean velocity has the same format as the original Navier-Stokes equation except that a Reynolds Stress term arises [84], which makes the mean velocity equation system not closed. For turbulence modeling solving the Reynolds-Averaged Navier-Stokes (RANS) equations [38], according to the Boussinesq approximation [39], the turbulence transfer mechanism can be modeled as an eddy viscosity or turbulent viscosity. The Eddy viscosity model is thus brought out, which uses extra equation(s) to make the equation system closed. The most commonly used mixed-length model, 1-equation model and 2-equation models all belong to the eddy viscosity model family. For incompressible flow, the basic idea of the eddy viscosity model can be summarized using Eq. (3.3). Eq. (3.2) is the original format of the Navier-Stokes equation. The governing equation of Reynolds averaged velocity is as Eq. (3.3). They are similar except that a turbulence viscosity ν_t (or eddy viscosity) is brought out to simulate the Reynolds stress term that arises from the time-average treatment:

$$\frac{\partial u_i}{\partial t} + \frac{\partial\left(u_j u_i\right)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + S_{F,i} \tag{3.2}$$

$$\frac{\partial \bar{U}_i}{\partial t} + \frac{\partial\left(\bar{U}_j \bar{U}_i\right)}{\partial x_j} = -\frac{\partial \bar{P}}{\partial x_i} + (\nu + \nu_t) \frac{\partial^2 \bar{U}_i}{\partial x_j^2} + S_{F,i} \tag{3.3}$$

where u represents velocity and $\bar{U}$ represents the Reynolds averaged velocity.

RANS-based turbulence models are very promising tools for indoor environment modeling since they provide informative results and are inexpensive [85]. There are many different RANS-based turbulence models developed for specific turbulence flows, but none of them is universally accepted as superior for all classes of problems.

Among RANS-based turbulence models, k-ε model is the most widely used and validated [86]. Many studies [87-89] proved that Renormalized Group (RNG) k-ε model [43] overall has the best performance in predicting indoor airflow. This is because it provides a low Reynolds number interpolation formula for the turbulent viscosity, which is valid for low to high Reynolds number flows. Two extra equations on turbulent kinetic energy (k) and dissipation (ε) are used to make the equations closed. The turbulence viscosity in the k-ε model family is modeled as a function of mean turbulent kinetic energy and dissipation as in Eq. (3.4):

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{3.4}$$

where C_μ is a constant empirically determined and validated by the specific turbulence model.

Transport equations of k and ε follow the same format as the other variables, therefore bringing in two extra PDEs to the equation system. Zero-equation models, which bring in no extra PDE to the equation system, use algebraic expression to model the turbulence viscosity. One important zero-equation model for enclosed environment modeling developed by Chen and Xu [46] is:

$$\nu_t = 0.03874UL \tag{3.5}$$

where 0.03874 is an empirical constant validated for different indoor airflows, U is the local mean velocity, and L is the distance to the nearest wall.

A further simplification of turbulence modeling can treat this eddy viscosity ν_t as a constant, which works well for some free shear flows. Meanwhile, the appropriate magnitude of such viscosity values is yet understood for IEQ modeling.

3.2 Numerical Viscosity

To numerically solve the governing equations, discretization is practiced. A continuous spatial domain is discretized, which introduces numerical errors to correctly approximate the exact solutions of governing equations. Finer discretization resolution leads to less numerical error. Theoretically, when refining grid resolutions, there is a point that further refinement will not (or only slightly) change numerical solutions. This is the minimum grid resolution upon which a grid-independent solution can be obtained. The truncation error caused by discretizing the continuous governing equations of flow is the fundamental basis of this phenomenon. To quantitatively evaluate the truncation error brought into numerical solutions of governing equations, numerical viscosity analysis is performed. All the following derivations are based on momentum equations of instant velocity components; however, the numerical viscosity result will be also applicable to RANS based governing equations, as well as temperature and concentration equations.

3.2.1 Upwind Differencing Scheme

First order upwind scheme is a very popular and unconditionally stable; however, the answer can be physically invalid in some situations. Using a two-dimensional (2D) steady-state incompressible flow as an example, the momentum equations are expressed as:

$$\begin{cases} u\dfrac{\partial u}{\partial x}+v\dfrac{\partial u}{\partial y}=-\dfrac{1}{\rho}\dfrac{\partial P}{\partial x}+v\left[\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}\right]+S_x \\ u\dfrac{\partial v}{\partial x}+v\dfrac{\partial v}{\partial y}=-\dfrac{1}{\rho}\dfrac{\partial P}{\partial y}+v\left[\dfrac{\partial^2 v}{\partial x^2}+\dfrac{\partial^2 v}{\partial y^2}\right]+S_y \end{cases} \tag{3.6}$$

where u and v are the velocity components along x and y direction, P is the pressure of the fluid, ρ and ν are the density and physical kinematic viscosity, and S_x and S_y are the external forces on the fluid along x and y direction. Using the upwind numerical scheme, assume u>0 and v>0, the 1st order term can be discretized as:

$$\begin{cases} \dfrac{\partial u}{\partial x}=\dfrac{u_{i,j}-u_{i-1,j}}{\Delta x} \\ \dfrac{\partial u}{\partial y}=\dfrac{u_{i,j}-u_{i,j-1}}{\Delta y} \\ \dfrac{\partial v}{\partial x}=\dfrac{v_{i,j}-v_{i-1,j}}{\Delta x} \\ \dfrac{\partial v}{\partial y}=\dfrac{v_{i,j}-v_{i,j-1}}{\Delta y} \end{cases} \tag{3.7}$$

The steady-state momentum equations can thus be discretized as:

$$\begin{cases} u\dfrac{u_{i,j}-u_{i-1,j}}{\Delta x}+v\dfrac{u_{i,j}-u_{i,j-1}}{\Delta y}=-\dfrac{1}{\rho}\dfrac{\partial P}{\partial x}+\nu\left[\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}\right]+S_x \\ u\dfrac{v_{i,j}-v_{i-1,j}}{\Delta x}+v\dfrac{v_{i,j}-v_{i,j-1}}{\Delta y}=-\dfrac{1}{\rho}\dfrac{\partial P}{\partial y}+\nu\left[\dfrac{\partial^2 v}{\partial x^2}+\dfrac{\partial^2 v}{\partial y^2}\right]+S_y \end{cases} \tag{3.8}$$

Using the Taylor series to express the parameter on adjacent cells:

$$\begin{cases} u_{i-1,j}=u_{i,j}-\Delta x\dfrac{\partial u_{i,j}}{\partial x}+\dfrac{\Delta x^2}{2}\dfrac{\partial^2 u_{i,j}}{\partial x^2}+O(\Delta x^3) \\ u_{i,j-1}=u_{i,j}-\Delta y\dfrac{\partial u_{i,j}}{\partial y}+\dfrac{\Delta y^2}{2}\dfrac{\partial^2 u_{i,j}}{\partial y^2}+O(\Delta y^3) \end{cases} \tag{3.9}$$

This yields:

$$\begin{cases} \dfrac{u_{i,j}-u_{i-1,j}}{\Delta x}=\dfrac{\partial u_{i,j}}{\partial x}-\dfrac{\Delta x}{2}\dfrac{\partial^2 u_{i,j}}{\partial x^2}-O(\Delta x^2) \\ \dfrac{u_{i,j}-u_{i,j-1}}{\Delta y}=\dfrac{\partial u_{i,j}}{\partial y}-\dfrac{\Delta y}{2}\dfrac{\partial^2 u_{i,j}}{\partial y^2}-O(\Delta y^2) \end{cases} \tag{3.10}$$

Similarly,

$$\begin{cases} v_{i-1,j}=v_{i,j}-\Delta x\dfrac{\partial v_{i,j}}{\partial x}+\dfrac{\Delta x^2}{2}\dfrac{\partial^2 v_{i,j}}{\partial x^2}+O(\Delta x^3) \\ v_{i,j-1}=v_{i,j}-\Delta y\dfrac{\partial v_{i,j}}{\partial y}+\dfrac{\Delta y^2}{2}\dfrac{\partial^2 v_{i,j}}{\partial y^2}+O(\Delta y^3) \end{cases} \tag{3.11}$$

And:

$$\begin{cases} \dfrac{v_{i,j}-v_{i-1,j}}{\Delta x}=\dfrac{\partial v_{i,j}}{\partial x}-\dfrac{\Delta x}{2}\dfrac{\partial^2 v_{i,j}}{\partial x^2}-O(\Delta x^2) \\ \dfrac{v_{i,j}-v_{i,j-1}}{\Delta y}=\dfrac{\partial v_{i,j}}{\partial y}-\dfrac{\Delta y}{2}\dfrac{\partial^2 v_{i,j}}{\partial y^2}-O(\Delta y^2) \end{cases} \tag{3.12}$$

Substituting the discretization terms in Eq. (3.8) with the Taylor series expansion (3.10) and (3.12), the momentum equations become:

$$\begin{cases} u\left(\frac{\partial u_{i,j}}{\partial x} - \frac{\Delta x}{2}\frac{\partial^2 u_{i,j}}{\partial x^2} - O(\Delta x^2)\right) + v\left(\frac{\partial u_{i,j}}{\partial y} - \frac{\Delta y}{2}\frac{\partial^2 u_{i,j}}{\partial y^2} - O(\Delta y^2)\right) = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + S_x \\ u\left(\frac{\partial v_{i,j}}{\partial x} - \frac{\Delta x}{2}\frac{\partial^2 v_{i,j}}{\partial x^2} - O(\Delta x^2)\right) + v\left(\frac{\partial v_{i,j}}{\partial y} - \frac{\Delta y}{2}\frac{\partial^2 v_{i,j}}{\partial y^2} - O(\Delta y^2)\right) = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right] + S_y \end{cases} \tag{3.13}$$

And further as:

$$\begin{cases} u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{u \cdot \Delta x}{2}\frac{\partial^2 u}{\partial x^2} + \frac{v \cdot \Delta y}{2}\frac{\partial^2 u}{\partial y^2} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + S_x + O(\Delta x^2) + O(\Delta y^2) \\ u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \frac{u \cdot \Delta x}{2}\frac{\partial^2 v}{\partial x^2} + \frac{v \cdot \Delta y}{2}\frac{\partial^2 v}{\partial y^2} + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right] + S_y + O(\Delta x^2) + O(\Delta y^2) \end{cases} \tag{3.14}$$

Since the coefficients $(\mathrm{u} \cdot \Delta\mathrm{x})/2$ and $(\mathrm{v} \cdot \Delta\mathrm{y})/2$ have the same effect as the physical viscosity of the fluid, it is called artificial viscosity [90] or numerical viscosity. The numerical viscosities for x and y directions are, respectively,

$$\begin{cases} \nu_x = u\frac{\Delta x}{2} \\ \nu_y = v\frac{\Delta y}{2} \end{cases} \tag{3.15}$$

If u<0, $\nu_x = -u\,\Delta x/2$, and thus $\nu_x = |u\,\Delta x/2|$. If v<0, $\mathrm{\nu_y} = -\mathrm{v}\,\Delta\mathrm{y}/2$, and thus $\nu_y = |v\,\Delta y/2|$. Similarly, for 3D cases, $\nu_z = |w\,\Delta z/2|$. The magnitude of numerical viscosity is proportional to the grid size, explaining the fact that refining grid improves the accuracy of a CFD simulation.

For the hybrid scheme, the upwind discretization is employed when the Péclet number is greater than 2 and the central differential scheme (CDS) is used for Pe ≤ 2 [91]. The detailed derivation of these terms is as follows.

3.2.2 Central Differencing Scheme (CDS)

CDS is also the scheme used by the Hybrid scheme when $Pe \leq 2$. The expression of Hybrid scheme is:

$$u_i = \begin{cases} u_{i-1} & Pe > 2 \;\; upwind\ scheme \\ \frac{1}{2}(u_{i-1} + u_{i+1}) & Pe \leq 2 \;\; central\ scheme \end{cases} \tag{3.16}$$

For CDS, assume u>0 and v>0; 1st order term can be discretized as:

$$\begin{aligned} \frac{\partial u}{\partial x} &= \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \\ \frac{\partial u}{\partial y} &= \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \\ \frac{\partial v}{\partial x} &= \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} \\ \frac{\partial v}{\partial y} &= \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} \end{aligned} \tag{3.17}$$

Thus the steady-state momentum equation can be discretized as:

$$\begin{aligned} u\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + v\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} &= -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + S_x \\ u\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} + v\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} &= -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right] + S_y \end{aligned} \tag{3.18}$$

Using the Taylor series to express the parameter on an adjacent cell:

$$\begin{aligned}
u_{i+1,j} &= u_{i,j} + \Delta x\frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^4) \\
u_{i-1,j} &= u_{i,j} - \Delta x\frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^4)
\end{aligned} \tag{3.19}$$

And

$$\begin{aligned}
u_{i,j+1} &= u_{i,j} + \Delta y\frac{\partial u_{i,j}}{\partial y} + \frac{\Delta y^2}{2}\frac{\partial^2 u_{i,j}}{\partial y^2} + \frac{\Delta y^3}{6}\frac{\partial^3 u_{i,j}}{\partial y^3} + O(\Delta y^4) \\
u_{i,j-1} &= u_{i,j} - \Delta y\frac{\partial u_{i,j}}{\partial y} + \frac{\Delta y^2}{2}\frac{\partial^2 u_{i,j}}{\partial y^2} + \frac{\Delta y^3}{6}\frac{\partial^3 u_{i,j}}{\partial y^3} + O(\Delta y^4)
\end{aligned} \tag{3.20}$$

Thus we have:

$$\begin{aligned}
\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} &= \frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^3) \\
\frac{u_{i,j} - u_{i,j-1}}{2\Delta y} &= \frac{\partial u_{i,j}}{\partial y} + \frac{\Delta y^2}{6}\frac{\partial^3 u_{i,j}}{\partial y^3} + O(\Delta y^3)
\end{aligned} \tag{3.21}$$

Similarly

$$\begin{aligned}
\frac{v_{i,j} - v_{i-1,j}}{2\Delta x} &= \frac{\partial v_{i,j}}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 v_{i,j}}{\partial x^3} + O(\Delta x^3) \\
\frac{v_{i,j} - v_{i,j-1}}{2\Delta y} &= \frac{\partial v_{i,j}}{\partial y} + \frac{\Delta y^2}{6}\frac{\partial^3 v_{i,j}}{\partial y^3} + O(\Delta y^3)
\end{aligned} \tag{3.22}$$

Substitute the discretization term by the Taylor series expansion:

$$\begin{aligned}
u\left(\frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^3)\right) + v\left(\frac{\partial u_{i,j}}{\partial y} + \frac{\Delta y^2}{6}\frac{\partial^3 u_{i,j}}{\partial y^3} + O(\Delta y^3)\right) &= -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + S_x \\
u\left(\frac{\partial v_{i,j}}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 v_{i,j}}{\partial x^3} + O(\Delta x^3)\right) + v\left(\frac{\partial v_{i,j}}{\partial y} + \frac{\Delta y^2}{6}\frac{\partial^3 v_{i,j}}{\partial y^3} + O(\Delta y^3)\right) &= -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right] + S_y
\end{aligned}$$

(3.23)

Further as:

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial x}-\frac{u\cdot\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3}-\frac{v\cdot\Delta y^2}{6}\frac{\partial^3 u}{\partial y^3}+\nu\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]+S_x+O(\Delta x^3)+O(\Delta y^3)$$
$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial y}-\frac{u\cdot\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3}-\frac{v\cdot\Delta y^2}{6}\frac{\partial^3 u}{\partial y^3}+\nu\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right]+S_y+O(\Delta x^3)+O(\Delta y^3) \tag{3.24}$$

To make it the same format as physical viscosity, the numerical viscosities for x and y directions are expressed as coefficients of the second derivative term, respectively as:

$$\nu_x=\frac{\frac{-u\cdot\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3}}{\frac{\partial^2 u}{\partial x^2}}$$
$$\nu_y=\frac{\frac{-v\cdot\Delta y^2}{6}\frac{\partial^3 u}{\partial y^3}}{\frac{\partial^2 u}{\partial y^2}} \tag{3.25}$$

For u<0

$$\nu_x=\frac{\frac{u\cdot\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3}}{\frac{\partial^2 u}{\partial x^2}} \tag{3.26}$$

And for v<0

$$\nu_y = \frac{\frac{v \cdot \Delta y^2}{6} \frac{\partial^3 u}{\partial y^3}}{\frac{\partial^2 u}{\partial y^2}} \tag{3.27}$$

Thus a general expression of the numerical viscosity term is (on U equation):

$$\begin{gathered} \nu_x = -\frac{\frac{|u| \cdot \Delta x^2}{6} \frac{\partial^3 u}{\partial x^3}}{\frac{\partial^2 u}{\partial x^2}} \\ \nu_y = -\frac{\frac{|v| \cdot \Delta y^2}{6} \frac{\partial^3 u}{\partial y^3}}{\frac{\partial^2 u}{\partial y^2}} \end{gathered} \tag{3.28}$$

Apply the same approach to z direction

$$\nu_z = -\frac{\frac{|w| \cdot \Delta z^2}{6} \frac{\partial^3 u}{\partial z^3}}{\frac{\partial^2 u}{\partial z^2}} \tag{3.29}$$

So a general expression of numerical viscosity of CDS is:

$$\nu_i = -\frac{\frac{|u_i| \cdot \Delta x_i^2}{6} \frac{\partial^3 u_i}{\partial x_j^3}}{\frac{\partial^2 u_i}{\partial x_j^2}} \tag{3.30}$$

3.2.3 QUICK Scheme

The QUICK (Quadratic upstream interpolation for convective kinematics) [92] scheme is a third order differencing scheme. With QUICK scheme, the convection term can be discretized as:

$$\frac{\partial u}{\partial x} = \frac{3u_{i+1,j} + 3u_{i,j} - 7u_{i-1,j} + u_{i-2,j}}{8\Delta x} \tag{3.31}$$

Using the Taylor series to express the parameter on an adjacent cell:

$$\begin{aligned} u_{i+1,j} &= u_{i,j} + \Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^4) \\ u_{i-1,j} &= u_{i,j} - \Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^4) \\ u_{i-2,j} &= u_{i,j} - 2\Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{4\Delta x^2}{2}\frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{8\Delta x^3}{6}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^4) \end{aligned} \tag{3.32}$$

Thus

$$\frac{3u_{i+1,j} + 3u_{i,j} - 7u_{i-1,j} + u_{i-2,j}}{8\Delta x} = \frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{24}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^3) \tag{3.33}$$

Similarly

$$\frac{3u_{i,j+1} + 3u_{i,j} - 7u_{i,j-1} + u_{i,j-2}}{8\Delta x} = \frac{\partial u_{i,j}}{\partial y} + \frac{\Delta y^2}{24}\frac{\partial^3 u_{i,j}}{\partial y^3} + O(\Delta y^3) \tag{3.34}$$

Substitute the discretization term by the Taylor series expansions:

$$u\left(\frac{\partial u_{i,j}}{\partial x} + \frac{\Delta x^2}{24}\frac{\partial^3 u_{i,j}}{\partial x^3} + O(\Delta x^3)\right) + v\left(\frac{\partial u_{i,j}}{\partial y} + \frac{\Delta y^2}{24}\frac{\partial^3 u_{i,j}}{\partial y^3} + O(\Delta y^3)\right) = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] + S_x \tag{3.35}$$

Further as:

$$u\frac{\partial u_{i,j}}{\partial x} + v\frac{\partial u_{i,j}}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] - u\frac{\Delta x^2}{24}\frac{\partial^3 u_{i,j}}{\partial x^3} - v\frac{\Delta y^2}{24}\frac{\partial^3 u_{i,j}}{\partial y^3} + S_x + O(\Delta x^3) + O(\Delta y^3) \tag{3.36}$$

To make it the same format as physical viscosity, the numerical viscosities for x and y directions are expressed as coefficients of the second derivative term, respectively as:

$$\nu_x = \frac{\frac{-u \cdot \Delta x^2}{24} \frac{\partial^3 u}{\partial x^3}}{\frac{\partial^2 u}{\partial x^2}}$$
$$\nu_y = \frac{\frac{-v \cdot \Delta y^2}{24} \frac{\partial^3 u}{\partial y^3}}{\frac{\partial^2 u}{\partial y^2}} \tag{3.37}$$

For u<0

$$\nu_x = \frac{\frac{u \cdot \Delta x^2}{24} \frac{\partial^3 u}{\partial x^3}}{\frac{\partial^2 u}{\partial x^2}} \tag{3.38}$$

And for v<0

$$\nu_y = \frac{\frac{v \cdot \Delta y^2}{24} \frac{\partial^3 u}{\partial y^3}}{\frac{\partial^2 u}{\partial y^2}} \tag{3.39}$$

Thus a general expression of the numerical viscosity term is (on U equation):

$$\nu_x = -\frac{\frac{|u| \cdot \Delta x^2}{24} \frac{\partial^3 u}{\partial x^3}}{\frac{\partial^2 u}{\partial x^2}}$$
$$\nu_y = -\frac{\frac{|v| \cdot \Delta y^2}{24} \frac{\partial^3 u}{\partial y^3}}{\frac{\partial^2 u}{\partial y^2}} \tag{3.40}$$

Apply the same approach to the 3rd direction

$$\nu_z = -\frac{\frac{|w| \cdot \Delta z^2}{24} \frac{\partial^3 u}{\partial z^3}}{\frac{\partial^2 u}{\partial z^2}} \tag{3.41}$$

So a general expression of the numerical viscosity of CDS is:

$$\nu_i = -\frac{\frac{|u_i| \cdot \Delta x_i^2}{24} \frac{\partial^3 u_i}{\partial x_j^3}}{\frac{\partial^2 u_i}{\partial x_j^2}} \tag{3.42}$$

A summary of numerical viscosity and truncation error from different numerical schemes is shown in Table 2.

Table 2. Truncation error of discretization for different numerical schemes

Scheme	Numerical Viscosity	Truncation error	Higher Order Term (H.O.T)	Note
Upwind	$\left\lvert u_i \frac{\Delta x_i}{2} \right\rvert$	$\left\lvert u_i \frac{\Delta x_i}{2} \right\rvert \frac{\partial^2 u_i}{\partial x_j^2}$	$O(\Delta x^2)$	Same as Hybrid when $Pe > 2$
CDS	$-\frac{\frac{\lvert u_i \rvert \cdot \Delta x_i^2}{6} \frac{\partial^3 u_i}{\partial x_j^3}}{\frac{\partial^2 u_i}{\partial x_j^2}}$	$-\frac{\lvert u_i \rvert \cdot \Delta x_i^2}{6} \frac{\partial^3 u_i}{\partial x_j^3}$	$O(\Delta x^3)$	Same as Hybrid when $Pe \leq 2$
Quick	$-\frac{\frac{\lvert u_i \rvert \cdot \Delta x_i^2}{24} \frac{\partial^3 u_i}{\partial x_j^3}}{\frac{\partial^2 u_i}{\partial x_j^2}}$	$-\frac{\lvert u_i \rvert \cdot \Delta x_i^2}{24} \frac{\partial^3 u_i}{\partial x_j^3}$	$O(\Delta x^3)$	

A grid independent solution theoretically requires the numerical viscosity to be much smaller than the turbulence viscosity, so that the effect of grid-induced-error is fully eliminated from the “solved” governing equations. Since numerical viscosity is determined by both grid size and local velocities, it is difficult to find a generally good grid size that can always meet the requirement of grid independency for various simulation conditions. Nevertheless, the numerical viscosity provides an important aspect to check theoretically whether grid independency is reached or not.

3.3 Velocity-Pressure Decoupling Algorithm

3.3.1 SIMPLE Algorithm

The most prevalent strategy of solving such coupling problems in CFD is the algorithms in the SIMPLE family. It is an acronym for Semi-Implicit Method for Pressure-Linked Equations [47]. There are different modified variants on this algorithm, such as SIMPLER (SIMPLE Revised) [48], SIMPLEC (SIMPLE Consistent) [49], SIMPLEM (SIMPLE modified) [50] and SIMPLEX (SIMPLE extrapolation) [51], but the general strategy of predictor-corrector iteration is the same. An initial guess of velocity and pressure fields is used as the starting point of an iteration process. The guessed pressure field is used to solve the momentum equation. A pressure correction equation deduced from the continuity equation is then solved to update the pressure field. The new pressure field is then used to solve the momentum equation. As the iteration proceeds, the guessed field is improved and finally approaches the exact solution of the discretized equation.

The general procedure of the SIMPLE algorithm is

1. Guess initial pressure field p^*

2. Solve momentum equation to find velocity field u_i^*

3. Solve pressure correction equation to find pressure corrector p'. The pressure correction equation is from considering mass conservation (continuity equation).

4. Correct pressure and velocity

5. Repeat steps 2-4 with new pressure and velocity fields

The details of the SIMPLE algorithm will not be covered here since it can be found in any CFD textbook. A unified formulation of the SIMPLE algorithms family can be found at [93]. The modified SIMPLE algorithms usually consume more computation since extra procedures are involved, but the convergence rate is sped up due to such improvement. It is reported [94] that the computing time is reduced by 30-50% in using the SIMPLER algorithm.

3.3.2 PISO Algorithm

Different from the prevalence and success of SIMPLE family algorithms in a commercial software package, the PISO (Pressure-Implicit with Splitting of Operators) algorithm [17] is an efficient but somewhat underrated approach for pressure-velocity decoupling of the Navier-Stokes equation. It was developed originally for non-iterative computation of unsteady flow and successfully adapted for the iterative solution of steady-state problems. Considering the great potential of dealing with transient flow calculation without involving iteration, the next paragraphs will introduce the general procedure of the PISO algorithm for its possible application in real-time CFD.

Using calculus notation to express the general format of the momentum equation and continuity equation

$$\frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nu\nabla^2 u - \frac{1}{\rho}\nabla p + S \tag{3.43}$$

$$\nabla \cdot u = 0 \tag{3.44}$$

here u is used to represent the general velocity field. With first order time step discretization, the momentum and continuity equations can be written as:

$$\frac{u^{n+1} - u^n}{\Delta t} = -(u \cdot \nabla)u + \nu\nabla^2 u - \frac{1}{\rho}\nabla p + S \tag{3.45}$$

$$\nabla \cdot u^{n+1} = 0 \tag{3.46}$$

With some intermediate predictors and corrector steps (notate as 1, 2, 3… superscript; n and n+1 on superscript represent initial and final value over a time step), the PISO algorithm is developed in this paper.

Step 1

With initial flow field values of u^n and p^n, the intermediate field value u^1 can be obtained using an implicit scheme as Eq. (3.47)

$$\frac{u^1 - u^n}{\Delta t} = -(u^1 \cdot \nabla)u^1 + \nu\nabla^2 u^1 - \frac{1}{\rho}\nabla p^n + S \tag{3.47}$$

Since this is using p^n instead of p^{n+1}, u^1 will not satisfy the continuity equation.

Step 2

A second-step intermediate approximation of velocity is obtained from assuming an updated velocity field u^2 and pressure field p^1. Eq. (3.47) becomes

$$\frac{u^2 - u^n}{\Delta t} = -(u^1 \cdot \nabla)u^1 + \nu \nabla^2 u^1 - \frac{1}{\rho}\nabla p^1 + S \tag{3.48}$$

Eq. (3.48) subtract Eq. (3.47) yields

$$\frac{u^2 - u^1}{\Delta t} = -\frac{1}{\rho}(\nabla p^1 - \nabla p^n) \tag{3.49}$$

Take divergence for both sides of Eq. (3.49), together with the continuity equation $\nabla u^2 = 0$, such that velocity increment Eq. (3.49) yields the pressure increment equation to solve $p^1 - p^n$ field

$$\nabla^2 p^1 - \nabla^2 p^n = \frac{\rho}{\Delta t}\nabla \cdot u^1 \tag{3.50}$$

Step 3

An updated pressure field (or pressure increment field) can be plugged into Eq. (3.48) or Eq. (3.49) to update the velocity field and get the velocity field u^2.

Step 4

A replication of Step 2 is conducted using the updated result from Step 3 u^2 and initial value u^n, assuming a most updated pressure field p^{n+1}, and yields an updated field

$$\frac{u^{n+1}-u^{n}}{\Delta t}=-(u^{2}\cdot \nabla)u^{2}+\nu\nabla^{2}u^{2}-\frac{1}{\rho}\nabla p^{n+1}+S \tag{3.51}$$

Eq. (3.51) subtract Eq. (3.48) yields

$$\frac{u^{n+1}-u^{2}}{\Delta t}=-\frac{1}{\rho}(\nabla p^{n+1}-\nabla p^{1})+[-(u^{2}\cdot \nabla)u^{2}+\nu\nabla^{2}u^{2}]-[-(u^{1}\cdot \nabla)u^{1}+\nu\nabla^{2}u^{1}] \tag{3.52}$$

Take divergence for both sides of Eq. (3.52), together with the continuity equation $\nabla u^{n+1}=0$ and $\nabla u^{2}=0$, such that velocity increment Eq. (3.52) yields the pressure increment equation to solve $p^{n+1}-p^{1}$ field

$$\nabla^{2}p^{n+1}-\nabla^{2}p^{1}=\nabla\cdot[-(u^{2}\cdot \nabla)u^{2}+\nu\nabla^{2}u^{2}]-[-(u^{1}\cdot \nabla)u^{1}+\nu\nabla^{2}u^{1}] \tag{3.53}$$

Step 5

An updated pressure field (or pressure increment field) can be plugged into Eq. (3.51) or Eq. (3.52) to update the velocity field and approximate the exact solution u^{n+1} and p^{n+1}. This is proven to be sufficient for most practical purposes.

The computational sequence of the PISO algorithm can be summarized as Figure 3.

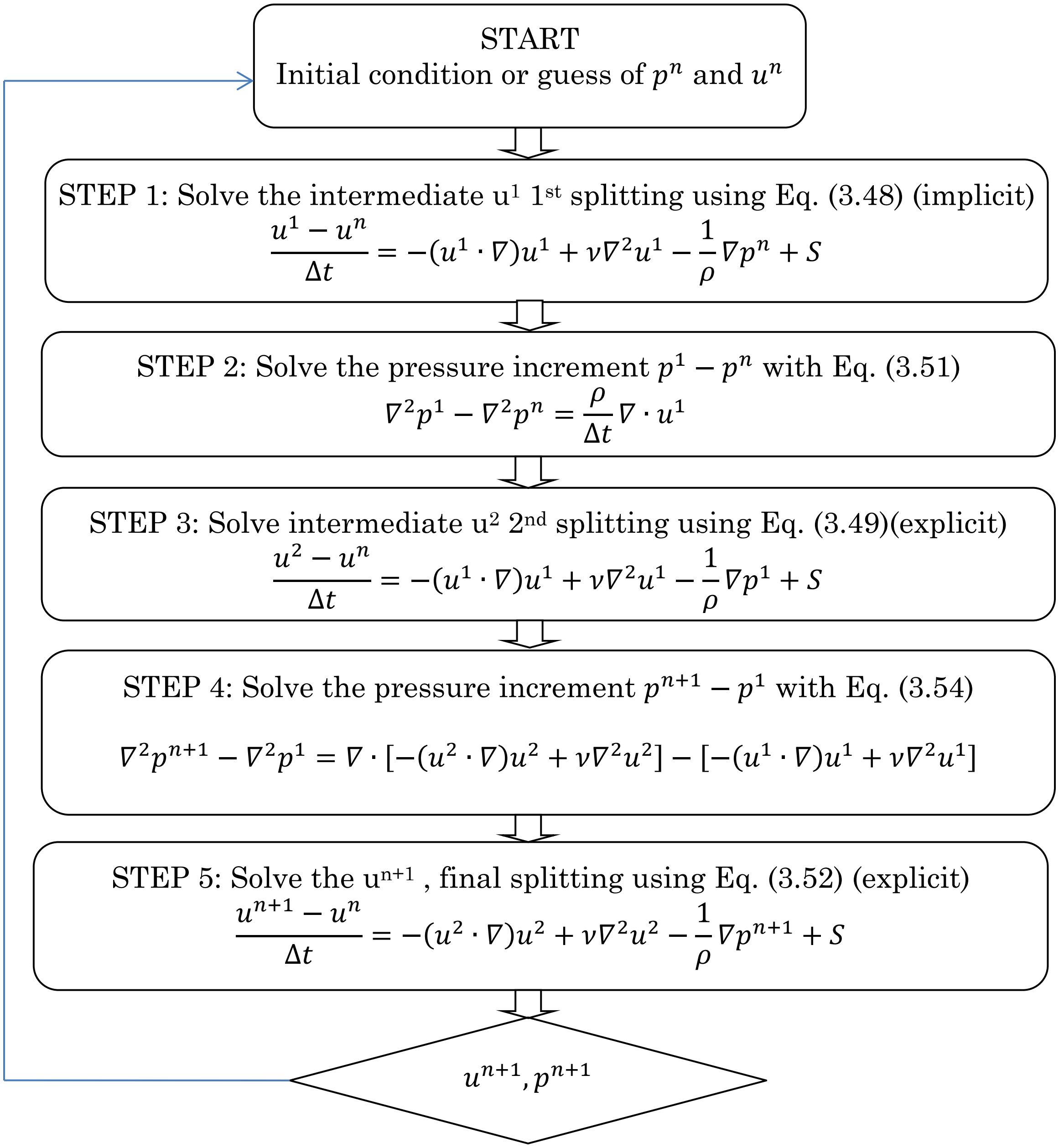

START
Initial condition or guess of p^n and u^n
STEP 1: Solve the intermediate u^1 1st splitting using Eq. (3.48) (implicit)
$\frac{u^1 - u^n}{\Delta t} = -(u^1 \cdot \nabla)u^1 + \nu\nabla^2 u^1 - \frac{1}{\rho}\nabla p^n + S$
STEP 2: Solve the pressure increment $p^1 - p^n$ with Eq. (3.51)
$\nabla^2 p^1 - \nabla^2 p^n = \frac{\rho}{\Delta t}\nabla \cdot u^1$
STEP 3: Solve intermediate u^2 2nd splitting using Eq. (3.49)(explicit)
$\frac{u^2 - u^n}{\Delta t} = -(u^1 \cdot \nabla)u^1 + \nu\nabla^2 u^1 - \frac{1}{\rho}\nabla p^1 + S$
STEP 4: Solve the pressure increment $p^{n+1} - p^1$ with Eq. (3.54)
$\nabla^2 p^{n+1} - \nabla^2 p^1 = \nabla \cdot [-(u^2 \cdot \nabla)u^2 + \nu\nabla^2 u^2] - [-(u^1 \cdot \nabla)u^1 + \nu\nabla^2 u^1]$
STEP 5: Solve the u^{n+1} , final splitting using Eq. (3.52) (explicit)
$\frac{u^{n+1} - u^n}{\Delta t} = -(u^2 \cdot \nabla)u^2 + \nu\nabla^2 u^2 - \frac{1}{\rho}\nabla p^{n+1} + S$
u^{n+1}, p^{n+1}

Figure 3. Sequence of operations for PISO algorithm

3.3.3 Projection Method

The projection method can be traced back to 1965 [52], in which a numerical algorithm is proposed for a staggered grid to calculate the time-dependent numerical solution of fluid flow. Foster et al. [57, 58] implemented such transient calculation with computers to simulate the 3D motion of hot, turbulent gas using a relatively coarse grid. An explicit discretization scheme was used, which raised stability issues when the time step is large. Therefore, when Stam [59] proposed using an implicit scheme, which is unconditionally stable, to animate the fluid-like flow physically; it was called stable fluids. Nevertheless, the most important contribution of stable fluids is not the stability but the idea of operator splitting and different techniques to speed up the computation, such as semi-Lagrangian advection and fast Fourier transformation, which speeded up the computation to a real-time or faster-than-real-time level. Zuo et al. [60] applied this operator splitting algorithm to indoor environment modeling, improved the sequence of operators, tested higher order of differencing schemes, and evaluated the accuracy level.

The Navier-Stokes Eq. (3.43) can be considered as a force to balance the equation of fluid in a control volume. The left-hand side can be considered as the acceleration of fluid in a certain control volume. The right-hand side, which is the sum up of several forces including advection (convection), diffusion, pressure and external force such as gravity and buoyancy from left to right, can be applied to the fluid sequentially to calculate the acceleration. Following the order of sequence suggested in [59, 95, 96], assign some intermediate velocity field u^1, u^2 and u^3 between time step $t+\Delta t$ and t, so the stable fluid solver is implemented as follows:

$$Force\colon \frac{u^1-u^n}{\Delta t}=S \tag{3.54}$$

$$Advection\colon \frac{u^2-u^1}{\Delta t}=-(u\cdot\nabla)u \tag{3.55}$$

$$Diffusion\colon \frac{u^3-u^2}{\Delta t}=\nu\nabla^2 u \tag{3.56}$$

$$Projection\colon \frac{u^{n+1}-u^3}{\Delta t}=-\frac{1}{\rho}\nabla p \tag{3.57}$$

In Eq. (3.57), since the pressure field is unknown for each time step, a projection method [18] is used as in Eq. (3.58), which is the divergence of Eq. (3.57) for both sides:

$$\nabla\cdot\left(\frac{u^{n+1}-u^3}{\Delta t}\right)=\nabla\cdot\left(-\frac{1}{\rho}\nabla p\right) \tag{3.58}$$

In this equation, as required by mass conservation equation (3.44), the final solution should fulfill the continuity equation as

$$\nabla\cdot u^{n+1}=0 \tag{3.59}$$

This yields a pressure calculation equation from intermediate result u^3 (Poisson equation)

$$\nabla^2 P=\frac{\rho}{\Delta t}\nabla\cdot u^3 \tag{3.60}$$

In Eq. (3.55), the advection of fluid is a non-linear expression, making it necessary to use iterative methods to solve. It is also noted that in Eq. (3.56), an implicit algorithm can be used for diffusion to ensure stability.

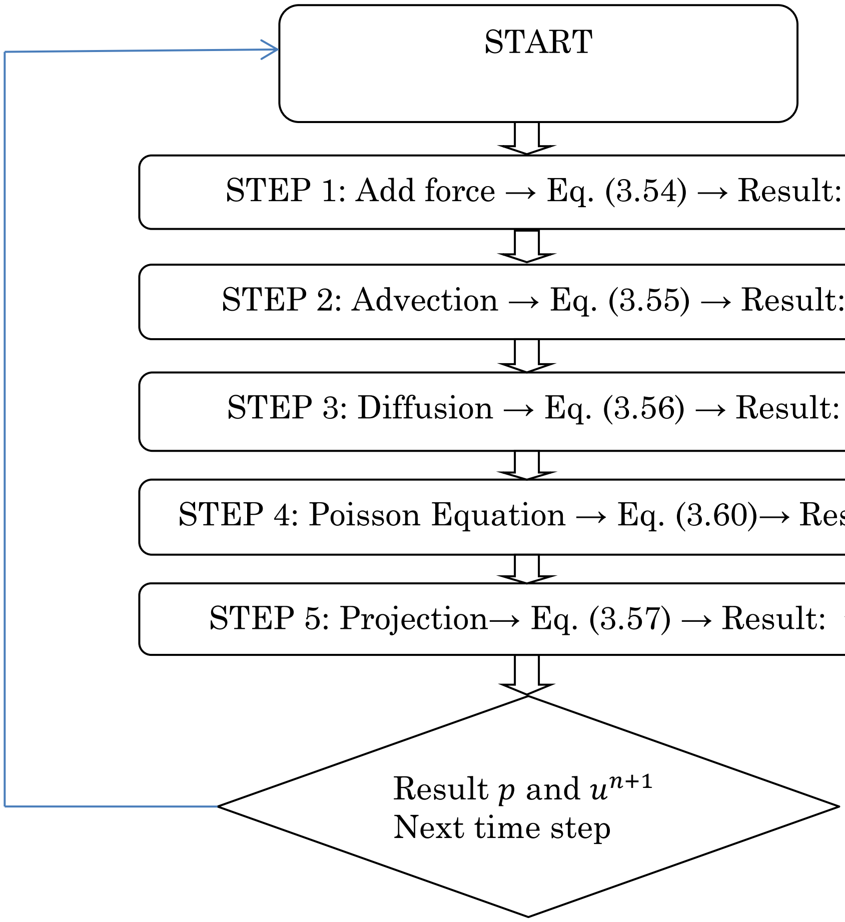

START
STEP 1: Add force → Eq. (3.54) → Result: u^1
STEP 2: Advection → Eq. (3.55) → Result: u^2
STEP 3: Diffusion → Eq. (3.56) → Result: u^3
STEP 4: Poisson Equation → Eq. (3.60)→ Result: p
STEP 5: Projection→ Eq. (3.57) → Result: u^{n+1}
Result p and u^{n+1}
Next time step

Figure 4. Sequence of Operation of FFD algorithm

The general sequence of FFD is as Figure 4. Zuo [97] optimized the sequence of each operator, applied/validated the algorithm in typical building airflow and utilized mass conservation correction to improve the accuracy. Numerical diffusion was reduced through using a higher order discretization scheme. The improved sequence of each operator is

Advection → Diffusion + Force → Projection

The result shows that accuracy of FFD needs further improvement for real building application. One issue and possible problem in this algorithm is that the pressure field for each time step is solely dependent on the velocity field and is independent from previous time step.

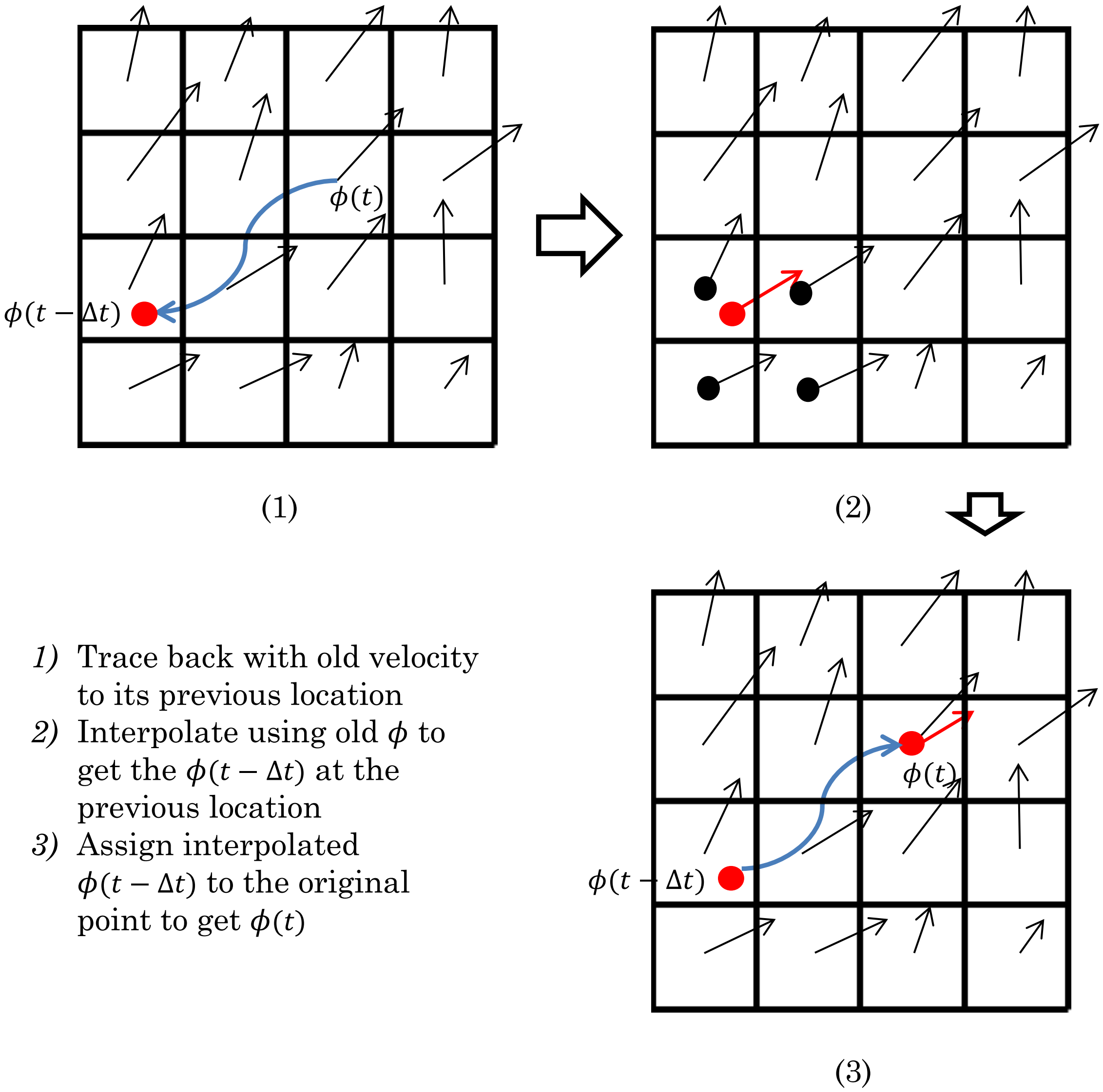
3.4 Semi-Lagrangian Advection

A fast CFD method requires the algorithm to be unconditionally stable. An implicit algorithm has no CFL restriction; however, in solving a momentum equation numerically, the advection term in Eq. (3.55) is fundamentally different from others since it ends up being a non-linear. A semi-Lagrangian scheme [79] is a good option for solving this problem. The idea of a semi-Lagrangian scheme was originated from advection of scalar, but it can be directly applied to vector.

The Lagrangian approach – named after French mathematician Lagrange – treats the continuum as a particle system. Each point in the fluid is labeled as a separate particle. From the perspective of such particles, the observed value of ϕ (density, temperature, etc.) will remain the same with the lapse of time.

The semi-Lagrangian scheme can be illustrated as in Figure 5. With an existing velocity field, it can trace back to its original location at previous time step

$t - \Delta t$. Such location will not necessarily drop on the exact grid point, so use the ϕ value surrounding it to interpolate the value at this location. The ϕ value on this specific particle Δt ago is such interpolated value. So we can directly assign such values to the grid points from which we are tracing back.



1) Trace back with old velocity to its previous location
2) Interpolate using old ϕ to get the $\phi(t - \Delta t)$ at the previous location
3) Assign interpolated $\phi(t - \Delta t)$ to the original point to get $\phi(t)$

Figure 5. Procedure of semi-Lagrangian scheme

Since there is no CFL condition restriction, the time step and grid size used in a semi-Lagrangian scheme is usually large, which introduces large truncation error. Higher order schemes can be used to improve the accuracy in interpolation of a semi-Lagrangian scheme.

3.5 Evaluation of Simulation Result Accuracy

The validation of each identified case is to find the grid-independent grid resolution and establish the relationship between cell number used and prediction accuracy. In the CFD study, predicted parameters of different grid resolutions are usually plotted along a line and compared with each other to examine the difference to see if a grid convergence has been achieved. This method is usually subjective and relies much on the observation and judgment of the CFD investigator. The best way to avoid dispute on whether a grid-independency has been achieved is to establish an objective criteria.

The idea of a grid independent study is to check the difference between two or more data sets and judge how close they are to each other. The simplest way to evaluate the distance between two data sets is to use a single number. Roache [98] recommended GCI as an estimator of three times the error of a numerical solution and exact solution of governing equations as:

$$GCI = \frac{3|\varepsilon|}{r^p - 1} \tag{3.61}$$

where $\varepsilon = (\phi_1 - \phi_2)/\phi_2$ is the relative prediction difference between coarse and fine grid; ϕ_1 and ϕ_2 are the prediction of coarse and fine grid respectively; $r = 2$ for grid doubling (or halving); and p is the order of the numerical scheme. For a second order scheme $GCI = |\varepsilon|$ and it is obvious for any two grid resolutions with any numerical scheme, GCI is a coefficient multiplies the prediction difference. The first question that arises from this index is the variable and location for the data point selection in a real practice. For indoor environment modeling, the airflow and thermal environment are usually the most important concerns, which also determine the field of other variables such as species concentration. The location of data points is recommended to be uniformly distributed in the computational domain, which can be representative and avoid the bias for selecting only one point.

To sum up the error for all data points within investigation, the Euclidean norm (2-norm) is a frequently used estimator [99]. Such estimator is implemented previously [100] and proven to be a good index for error estimation in the grid independent study of indoor environment modeling. The average 2-norm estimator is expressed as:

$$\frac{1}{n}\left(\sum_{i=1}^{n}|\varepsilon(i)|^2\right)^{\frac{1}{2}} = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{[\phi_1(i) - \phi_2(i)]^2}{\phi_2(i)^2}\right)^{\frac{1}{2}} \tag{3.62}$$

where $\varepsilon(i) = \left(\phi_1(i) - \phi_2(i)\right)/\phi_2(i)$ is the relative error on point i, and n is the number of total points under investigation. Practically, this estimator faces a problem when the variable $\phi(i)$ is a very small number, making the estimator $\varepsilon(i)$ a huge number. To avoid such incidence, the average value of 2-norm of $\phi(i)$ on all locations is recommended as the normalization factor so that:

$$\varepsilon(i) = \frac{\phi_1(i) - \phi_2(i)}{\frac{1}{n}\sqrt{\sum_{i=1}^{n} \phi_2(i)^2}} \tag{3.63}$$

This makes the GCI index similar, as a coefficient multiplies the normalized root mean square error criterion. We simply call this index as root mean square error (RMSE) for further analysis:

$$\begin{aligned} RMSE(\phi_1, \phi_2) &= \frac{3}{r^p - 1}\frac{1}{n}\left(\sum_{i=1}^{n} \frac{[\phi_1(i) - \phi_2(i)]^2}{\left[\frac{1}{n}\sqrt{\sum_{i=1}^{n} \phi_2(i)^2}\right]^2}\right)^{\frac{1}{2}} \\ &= \frac{3}{r^p - 1}\sqrt{\frac{\sum_{i=1}^{n}[\phi_1(i) - \phi_2(i)]^2}{\sum_{i=1}^{n} \phi_2(i)^2}} \end{aligned} \tag{3.64}$$

where $RMSE(\phi_1, \phi_2)$ is derived from GCI considering the prediction of different locations in a computational domain. The RMSE value provides a practically feasible criterion for indoor environment modeling grid independent study. In a strict grid independent study, the grid refinement factor is recommended to be greater than 1.3 [101], which means the total grid number of fine grid resolution should be at least 1.3 times that of coarse grid resolution. In this study, we used a refinement factor of 2, and since different orders of numerical scheme ranged from one to three, the average value of $p = 2$. The coefficient on the normalized root mean square error (RMSE) criterion ends up as 1. The difference between results of two neighboring grid-resolutions thus can be computed by the following index in a uniform format:

$$RMSE(\phi_1, \phi_2) = \sqrt{\frac{\sum_{i=1}^{n}[\phi_1(i) - \phi_2(i)]^2}{\sum_{i=1}^{n}\phi_2(i)^2}} \tag{3.65}$$

where $\phi_1(i)$ and $\phi_2(i)$ are the predictions of the same parameter at the same location of the different grid-resolutions.

The definition of grid-independency is that the result of simulation is not affected by the density of the grid. Theoretically, keep increasing the grid number and there will be a critical point where continued increasing of the grid number will not change the result. However, in reality, the result will change all the way with the total grid number increase due to the round off error and convergent status, but there is a certain value under which a difference can be considered as the sign for reaching grid-independency.

Another application of normalized RMSE is to evaluate the performance of different CFD configurations. Using a profile plot is an efficient and visible approach to see the quality of the prediction of different CFD configurations. However, it is usually difficult to tell if one prediction is superior to the other when the two are close to each other. Also due to the complicity of CFD prediction, one result may have better prediction on one point but worse on another, especially when different measurement uncertainty is taken into consideration. When the comparison of CFD results and experimental data are in the whole flow field, it is usually difficult to draw a conclusion on the performance different CFD algorithms.

To avoid such problem, the normalized NRMSE value of the prediction to measurement is introduced to compare the prediction accuracy:

$$RMSE(P,M)=\sqrt{\frac{\sum_{i=1}^{n}\delta_{pm}\big(|P(i)-M(i)|-e(i)\big)^2}{\sum_{i=1}^{n}M(i)^2}} \tag{3.66}$$

$$\delta_{pm}=\begin{cases}1 & |P(i)-M(i)|>e(i)\\ 0 & |P(i)-M(i)|<e(i)\end{cases} \tag{3.67}$$

where $P(i)$and $M(i)$ are the prediction and measurement data sets in certain positions, respectively, and $e(i)$ is the uncertainty of test instrument in the experiment.

The basic idea of such a method is to see how far away the prediction deviates from the experimental data, taking into account the uncertainty in the experiment, and then normalized by the absolute value of the measurement data.

The normalized RMSE method can only give a general clue of the performance of a prediction, especially when the prediction is far away from the measurement. To perform a detailed analysis of the prediction, a profile comparison is still necessary. Therefore, the NRMSE value and profile plot together will be employed to show the performance of each prediction.

As stated above, a grid-independent study of this research will be conducted by increasing the total cell number on an order of 2, and the normalized RMSE value will be employed to compare the prediction difference between two neighboring grid-resolutions.

CHAPTER 4
COARSE GRID CFD AND NUMERICAL ERROR MINIMIZATION

4.1 Theoretical Analysis

4.1.1 Numerical Viscosity Analysis

As described in the previous Chapter, numerical viscosity is the fundamental cause of numerical error in a CFD model. The magnitude of numerical viscosity is proportional to the grid size for upwind scheme, explaining the fact that a coarse grid results in greater error. Refinement of the grid eliminates the impact of such discretized error through reducing numerical viscosity and thus improving the accuracy of a CFD simulation.

For turbulence modeling solving the RANS equations, the turbulence transfer mechanism can be modeled as an eddy viscosity or turbulent viscosity, which leads to an extra coefficient on the second derivative terms in the momentum equations. For a 2D steady-state turbulence flow, the momentum equations solved in CFD are in Reynolds averaged format as in Eq. (4.1):

$$\begin{cases} \bar{U}\dfrac{\partial \bar{U}}{\partial x} + \bar{V}\dfrac{\partial \bar{U}}{\partial y} = -\dfrac{1}{\rho}\dfrac{\partial \bar{P}}{\partial x} + \left[\nu_{physical} + \nu_t + \nu_x\right]\dfrac{\partial^2 \bar{U}}{\partial x^2} + \left[\nu_{physical} + \nu_t + \nu_y\right]\dfrac{\partial^2 \bar{U}}{\partial y^2} + S_x \\ \bar{U}\dfrac{\partial \bar{V}}{\partial x} + \bar{V}\dfrac{\partial \bar{V}}{\partial y} = -\dfrac{1}{\rho}\dfrac{\partial \bar{P}}{\partial y} + \left[\nu_{physical} + \nu_t + \nu_x\right]\dfrac{\partial^2 \bar{V}}{\partial x^2} + \left[\nu_{physical} + \nu_t + \nu_y\right]\dfrac{\partial^2 \bar{V}}{\partial y^2} + S_y \end{cases} \tag{4.1}$$

where U and V are Reynolds averaged velocity components; ν_{physical} is the physical viscosity written as ν previously. With these three viscosity terms on the diffusion term, it implies that turbulent flow behaves as a more viscous flow in nature, and the existence of numerical viscosity artificially aggravates such effect and factitiously makes the right hand side of the equation larger. Physical viscosity $\nu_{physical}$ can be neglected in analysis as it is usually one to two magnitudes smaller than ν_t for typical indoor environment flows.

$$\begin{cases} \bar{U}\dfrac{\partial \bar{U}}{\partial x} + \bar{V}\dfrac{\partial \bar{U}}{\partial y} = -\dfrac{1}{\rho}\dfrac{\partial \bar{P}}{\partial x} + [\nu_t + \nu_x]\dfrac{\partial^2 \bar{U}}{\partial x^2} + \left[\nu_t + \nu_y\right]\dfrac{\partial^2 \bar{U}}{\partial y^2} + S_x \\ \bar{U}\dfrac{\partial \bar{V}}{\partial x} + \bar{V}\dfrac{\partial \bar{V}}{\partial y} = -\dfrac{1}{\rho}\dfrac{\partial \bar{P}}{\partial y} + [\nu_t + \nu_x]\dfrac{\partial^2 \bar{V}}{\partial x^2} + \left[\nu_t + \nu_y\right]\dfrac{\partial^2 \bar{V}}{\partial y^2} + S_y \end{cases} \tag{4.2}$$

As numerical viscosity is proportional to grid size, the truncation error can be reduced by refinement of the grid until $\nu_x \ll \nu_t$ and $\nu_y \ll \nu_t$; therefore, the impact of numerical viscosity (i.e., grid size) is totally “eliminable.” This fundamentally provides an alternative to grid-independency verification.

Different discretizing schemes will cause different numerical viscosities. For higher order numerical schemes, the truncation error is different and the extra coefficient induced from the discretization is not on the diffusion term. For instance, the extra term induced from the second order scheme is related to the third order derivative of velocity.

Another important observation from this viscosity analysis is, in terms of the whole diffusion term, the truncation error (artificial diffusion) introduced by grid size can be expressed as in Eq. (4.3).

$$\begin{cases} [\nu_x]\dfrac{\partial^2\bar{U}}{\partial x^2}+[\nu_y]\dfrac{\partial^2\bar{U}}{\partial y^2} \\ [\nu_x]\dfrac{\partial^2\bar{V}}{\partial x^2}+[\nu_y]\dfrac{\partial^2\bar{V}}{\partial y^2} \end{cases} \tag{4.3}$$

From Eq. (4.3), the effect of numerical viscosity on total error is not only determined by the magnitude of itself but also determined by the corresponding second order derivatives it multiplies in the momentum equation. For example, for upwind scheme, the diffusion terms $(\partial^2 u)/(\partial x^2)$ and $(\partial^2 u)/(\partial y^2)$ might be magnitudes of difference for the same grid point. The weight factors of ν_x and ν_y are thus different, which leads to the potential of using a coarse grid in a certain direction without having a significant impact on the CFD solution.

4.1.2 Application of Normalized RMSE Value

In the previous section, normalized RMSE has already been introduced as a good alternative to evaluate CFD result accuracy. Eq. (4.4) and Eq. (4.5) will be useful for grid independency study in Section 4.2 and coarse grid result evaluation in Section 4.3.

$$RMSE(\phi_1, \phi_2) = \frac{3}{r^p - 1}\sqrt{\frac{\sum_{i=1}^{n}[\phi_1(i) - \phi_2(i)]^2}{\sum_{i=1}^{n}\phi_2(i)^2}} \tag{4.4}$$

where $\phi_1(i)$ and $\phi_2(i)$ are the predictions of the same parameter at the same location of the different grid-resolutions.

$$MSE(P, M) = \sqrt{\frac{\sum_{i=1}^{n}\delta_{pm}\left(|P(i) - M(i)| - e(i)\right)^2}{\sum_{i=1}^{n}M(i)^2}}$$

$$\delta_{pm} = \begin{cases} 1 & |P(i) - M(i)| > e(i) \\ 0 & |P(i) - M(i)| < e(i) \end{cases} \tag{4.5}$$

where $P(i)$ and $M(i)$ are the prediction and measurement data sets in certain positions, respectively, and $e(i)$ is the uncertainty of the test instrument in the experiment.

4.2 Grid-Independency Study

Grid independency is a necessity of using CFD solution for further scientific analysis. Theoretically, when the grid size approaches zero, the discretization error of CFD solution becomes zero. Traditionally in this area, profile of predicted value between different CFD grid resolutions (usually doubled each time) will be plotted to compared the difference. However, this method can be very subjective. Closeness between solutions can be very different at different locations of the profile; CFD practitioners have subjective judgments towards the same set of result.

In order to provide an objective criterion for grid-independency study specifically for IEQ research, a new method is brought out. According to the previous section, normalized RMSE and numerical viscosity analysis can both be applied to grid independency study. To evaluate the feasibility and performance of the proposed approach, particularly for indoor environment modeling, the normalized RMSE index and numerical viscosity analysis are employed for several typical indoor airflow predictions.

Since the RNG k-ε model has the overall best performance for indoor environment modeling among different RANS models [102, 103], this model was used for all the cases tested in this chapter. The test will be presented at the order of flow mechanisms of indoor airflow: natural convection, forced convection and mixed convection. Three representative numerical schemes (Upwind, Hybrid and QUICK scheme) are used. Although it is common to adopt non-uniform grids (or locally refined grids) for most CFD practices, here structured uniform grid distributions are employed, assuming CFD users have no prior knowledge on flow characteristics and thus no experience on local grid refinement. Indeed, uniform grid distribution makes it possible and easy to compare or verify numerical results against theoretical findings.

4.2.1 Natural Convection

Natural convection (NC) occurs when the flow is driven solely by a temperature-difference-induced buoyancy force. A two-dimensional (2D) experiment [104] conducted originally for CFD code validation purposes is used as an example to test and demonstrate the viability of the proposed method. The configuration, boundary condition and available measurement data for CFD model validation is shown in Figure 6. Two vertical walls were, respectively, heated and cooled to keep uniform temperatures, and the two horizontal walls have temperature distributions that can be discretized to ten small pieces, each of which has a uniform temperature. A clockwise-circulated flow inside the cavity was formed due to the temperature difference. Detailed measurements of temperature and velocity are available along the middle height and middle width lines. Along the mid-height line, vertical velocity component (v) and temperature data (t_1) are measured; while along the mid-width line, temperature data (t_2) is available from the experiment.

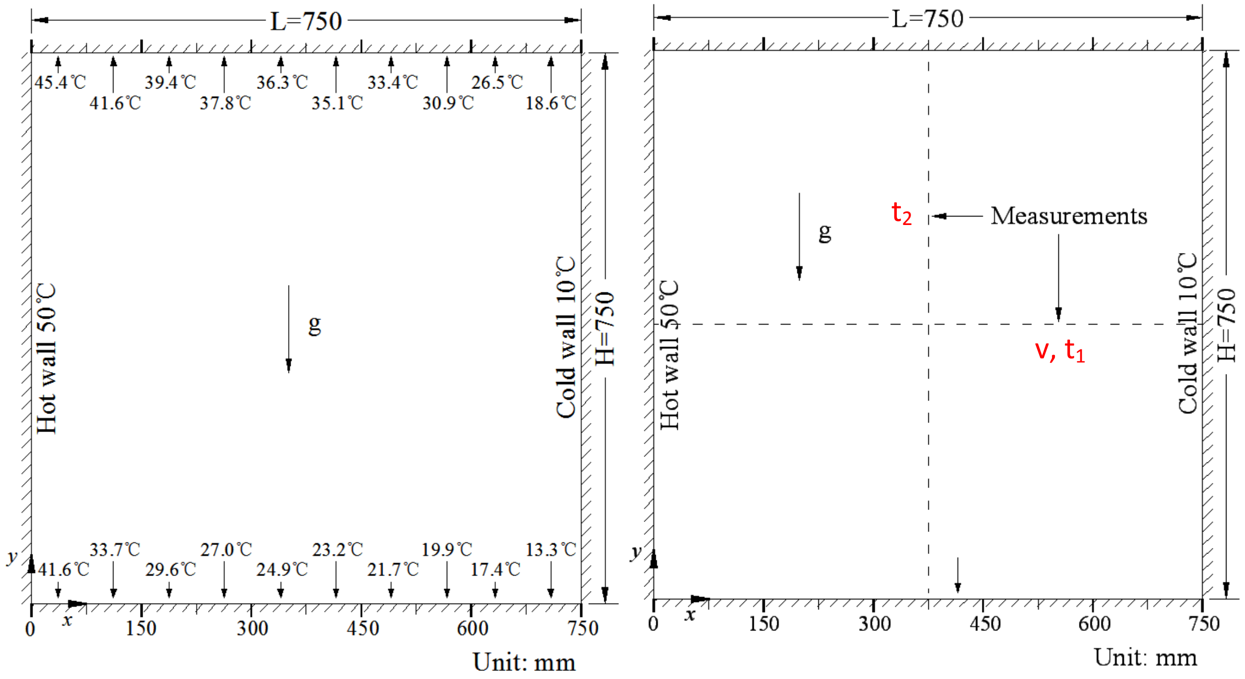


Figure 6. Configuration, boundary condition setup and measurements of NC Case

To ensure the comparability of difference in temperature with that in velocity, the absolute temperature has been normalized as below:

$$T = \frac{t - t_c}{t_h - t_c} \tag{4.6}$$

where t_h and t_c are the hot and cold wall temperature, and t is the predicted temperature at each measurement location. Velocity can also be normalized but it makes no difference to the RMSE value.

Table 3 shows the normalized RMSE result for this NC case. The grid resolution tested was doubled (r = 2) each time until reaching 1000×1000. Each modeling result was compared with that from the next finer grid. Note that no

RMSE value exists for grid #5, as this is the finest grid tested. The upwind scheme was used for this test (i.e., $p = 1$). According to Eq. (4.4), the RMSE for this NC case becomes:

$$RMSE(\phi_1, \phi_2) = 3\sqrt{\frac{\sum_{i=1}^{n}[\phi_1(i) - \phi_2(i)]^2}{\sum_{i=1}^{n}\phi_2(i)^2}} \tag{4.7}$$

Table 3. Grid resolutions and normalized RMSE results for the NC case

Grid index	Grid number (X*Y)	Normalized RMSE to next finer grid		
		V	*T1*	*T2*
#1	250*250	0.1825	0.0759	0.0392
#2	354*354	0.4886	0.1042	0.0681
#3	500*500	0.1369	0.0174	0.0186
#4	707*707	0.0541	0.0184	0.0157
#5	1000*1000	N/A	N/A	N/A

Figure 7 shows the profile comparison of predictions under different grid resolutions against the experimental data, a conventional approach to evaluating the grid independency of CFD modeling. Note that only portions of the profiles (near the left or bottom wall) were presented to highlight the differences among the simulations. The normalized RMSE values, however, compared all the data along the domain scales. Figure 7 reveals that results with grids #1 and #2 have a large discrepancy from those with the finer grids, as verified by the normalized RMSE values in Table 3. If using a 10% difference as the criterion to justify the grid independency, which is acceptable for most indoor applications, grid #4 is the grid that can provide the grid-independent solution for this NC case. This conclusion, however, may be difficult to draw if only based on the results in Figure 7. It is important to point out that the experimental data has nothing to do with the grid independency study. The grid-independent solution may not be the result that matches the experiment best as many other factors (e.g., turbulence model and boundary conditions) exist that will influence the accuracy of the simulation. On a separate note, most locations in experimental measurements are deliberatively chosen to cover the key flow zones and/or characteristics, thus providing great insight into selecting proper spatial points for the RMSE calculation and comparison.

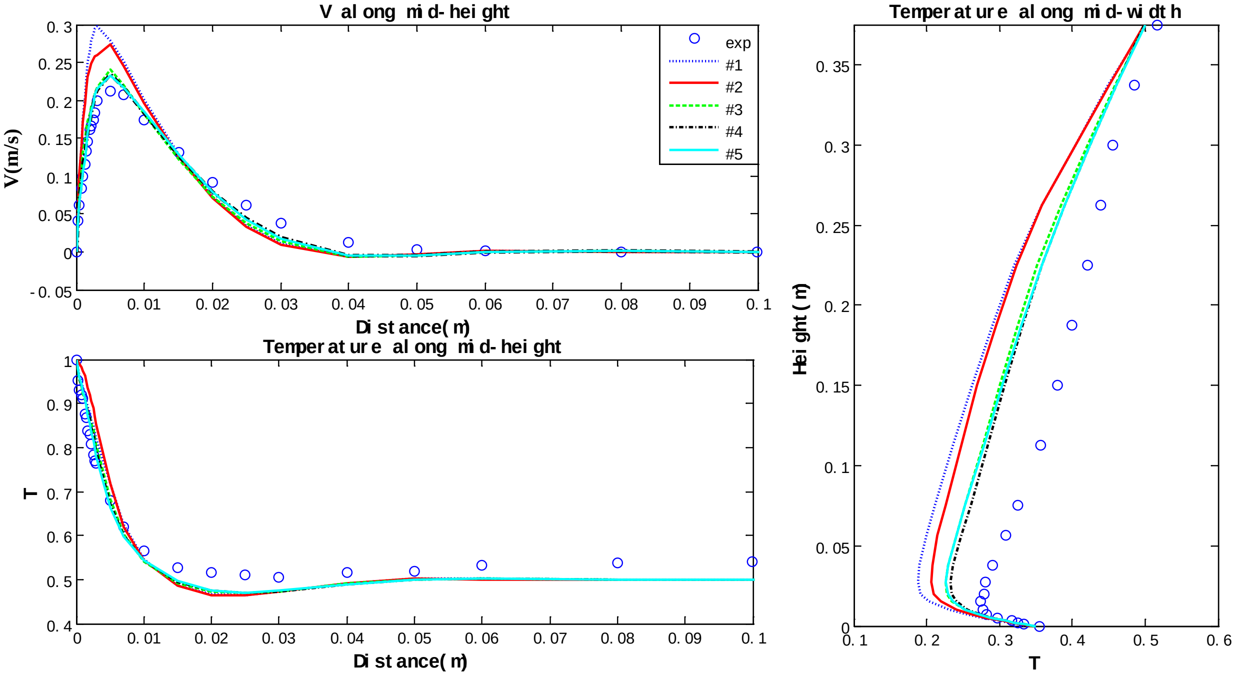


Figure 7. Profile comparison of predictions against measurements under different grid resolutions for the NC case

Figure 8 shows the contours of the calculated numerical viscosity from the results with grid #1 and grid #4. The very left contour is the turbulence viscosity distribution in the computational domain from the grid-independent solution (grid #4). The other four contours plot the calculated numerical viscosity with the upwind scheme for grids #1 and #4. For the original coarse grid #1, the numerical viscosity is comparable with the turbulent viscosity, especially in the critical locations near the walls, which results in the extra artificial diffusion in the flow and therefore affects the prediction. Grid #4, however, produces almost negligible numerical viscosity compared to the turbulent viscosity; thus further refinement of grids should not influence the simulation results, indicating the reach of a grid-independent solution. This viscosity analysis provides the fundamental verification on the mesh adequateness of grid #4.

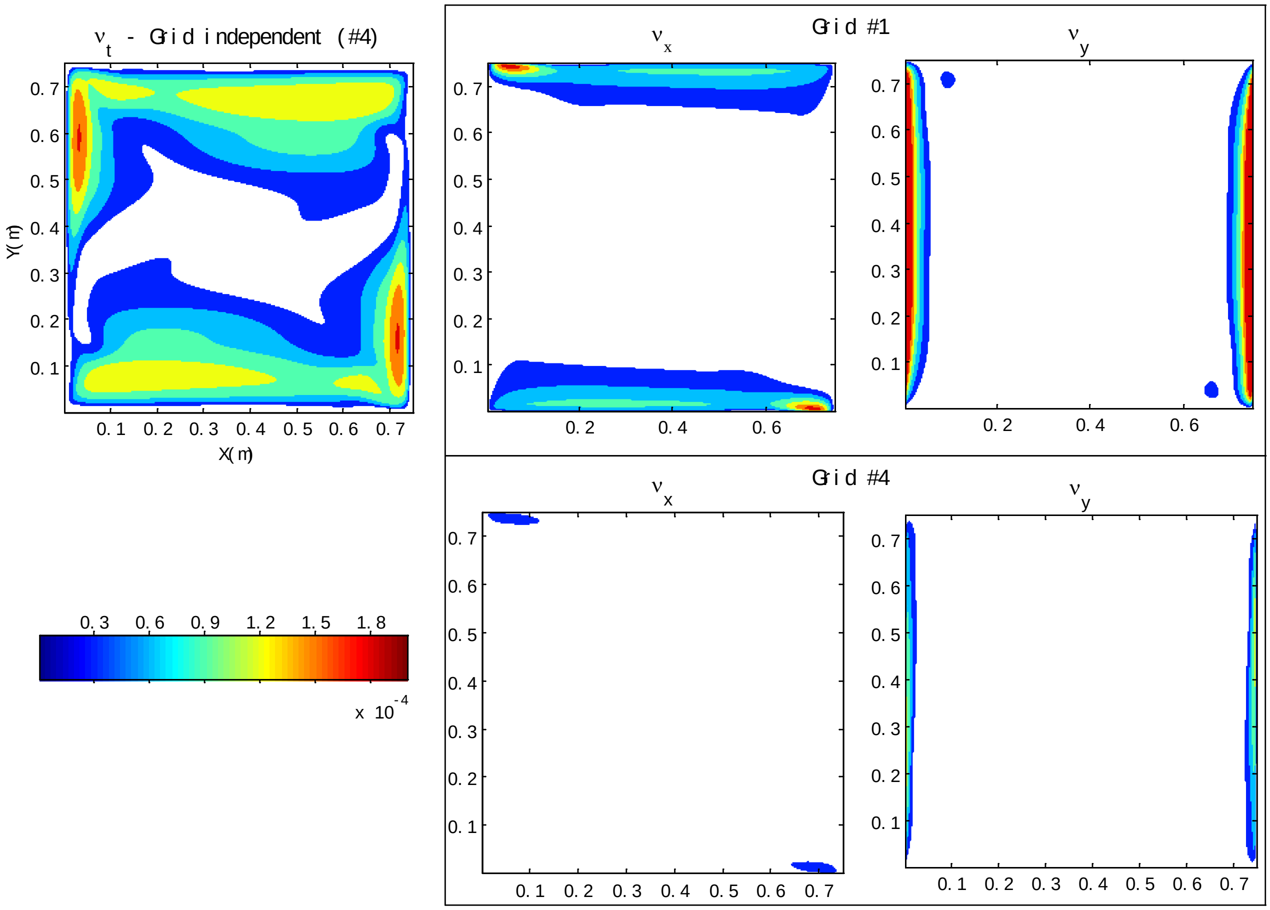


Figure 8. Comparison of numerical viscosity (grid-independent #4) with turbulence viscosity for the NC case

4.2.2 Forced Convection

Forced convection (FC) occurs when the air flow is solely driven by external forces such as mechanical force without the consideration of heat transfer (i.e., all the boundary conditions in the domain have the same temperature with no heat source or sink). This can be found in built environments where the space is mainly mechanically or wind ventilated with negligible temperature difference.

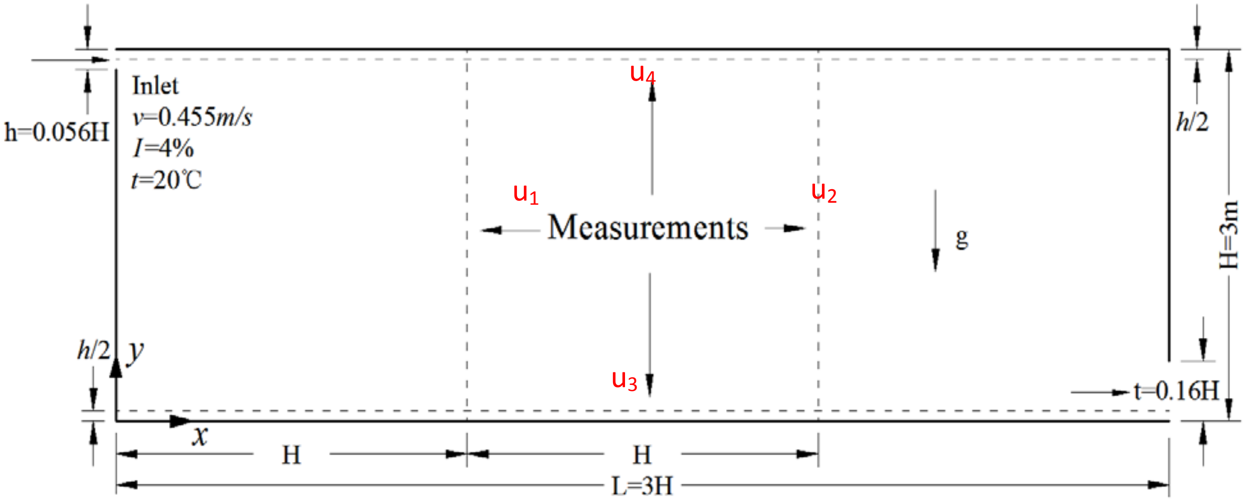


Figure 9. Configuration, boundary condition setup and measurements of FC Case

The widely used Nielsen's experiment [105] is employed in this study as Figure 9 shows the configuration of the experiment. The horizontal velocity component along four measurement lines is used for result validation.

Similar to the NC study, the total grid number was doubled each time in the FC simulation in order to check the prediction difference between two neighboring grid resolutions. This FC study used the hybrid scheme, so p = 1 (the upwind scheme) when $Pe > 2$ and $p = 2$ (CDS) when $Pe \leq 2$. When $p = 2$ is used, the normalized RMSE from Eq. (4.4) becomes:

$$RMS \quad (\phi_1, \phi_2) = \sqrt{\frac{\sum_{i=1}^{n}[\phi_1(i) - \phi_2(i)]^2}{\sum_{i=1}^{n}\phi_2(i)^2}} \tag{4.8}$$

The predicted velocity values are normalized by the inlet velocity for the comparison purpose:

$$U = \frac{u}{u_{inlet}} = \frac{u}{0.455} \tag{4.9}$$

Table 4 presents the grid independency study results based on the normalized RMSE criterion, which agrees well with the velocity profile comparison shown in Figure 10. Grid #1 results have a significant discrepancy from those on finer grids, especially with larger RMSE values for the velocities along two horizontal lines (*U3* and *U4*). Using the same criterion of 10%, the results indicate that grid #2 and above will be adequate for this FC case.

Figure 11 shows the contours of local turbulent and numerical viscosities from the grid-independent solution. A hybrid differencing scheme is used here. Therefore, for each grid point, the differencing scheme switches between upwind and CDS, and numerical viscosity is calculated accordingly. Note that for CDS, the numerical viscosities on the momentum equations of U (horizontal velocity component) and V (vertical velocity component) are different. The first contour in Figure 11 shows the turbulent viscosity as the reference, while the second and third groups show the estimated numerical viscosities of the hybrid scheme on the U and V momentum equations (a combination of numerical viscosities from upwind and CDS; the scheme used is determined by local Péclet number). The numerical viscosities on both equations are negligible compared to the turbulent viscosity.

This indicates that the numerical error has been eliminated by refining the grid and verifies the grid independency conclusion drawn by using the normalized RMSE method.

Table 4. Grid resolutions and normalized RMSE results for the FC case

		Normalized RMSE to next finer grid			
Grid index	Grid number (X*Y)	*U1*	*U2*	*U3*	*U4*
#1	106*36	0.0434	0.0778	0.1547	0.1932
#2	150*50	0.0454	0.0533	0.0588	0.0618
#3	212*71	0.0390	0.0212	0.0539	0.0604
#4	300*100	N/A	N/A	N/A	N/A

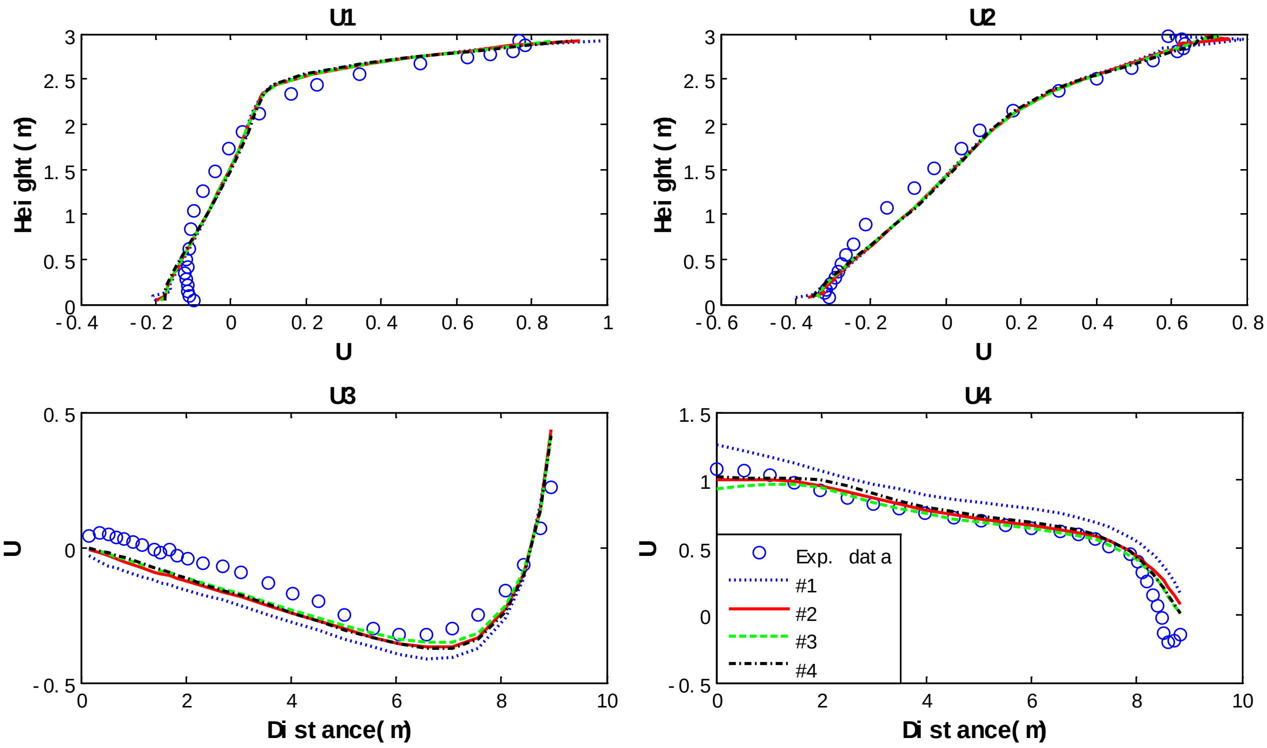


Figure 10. Profile comparison of predictions under different grid resolutions for the FC case

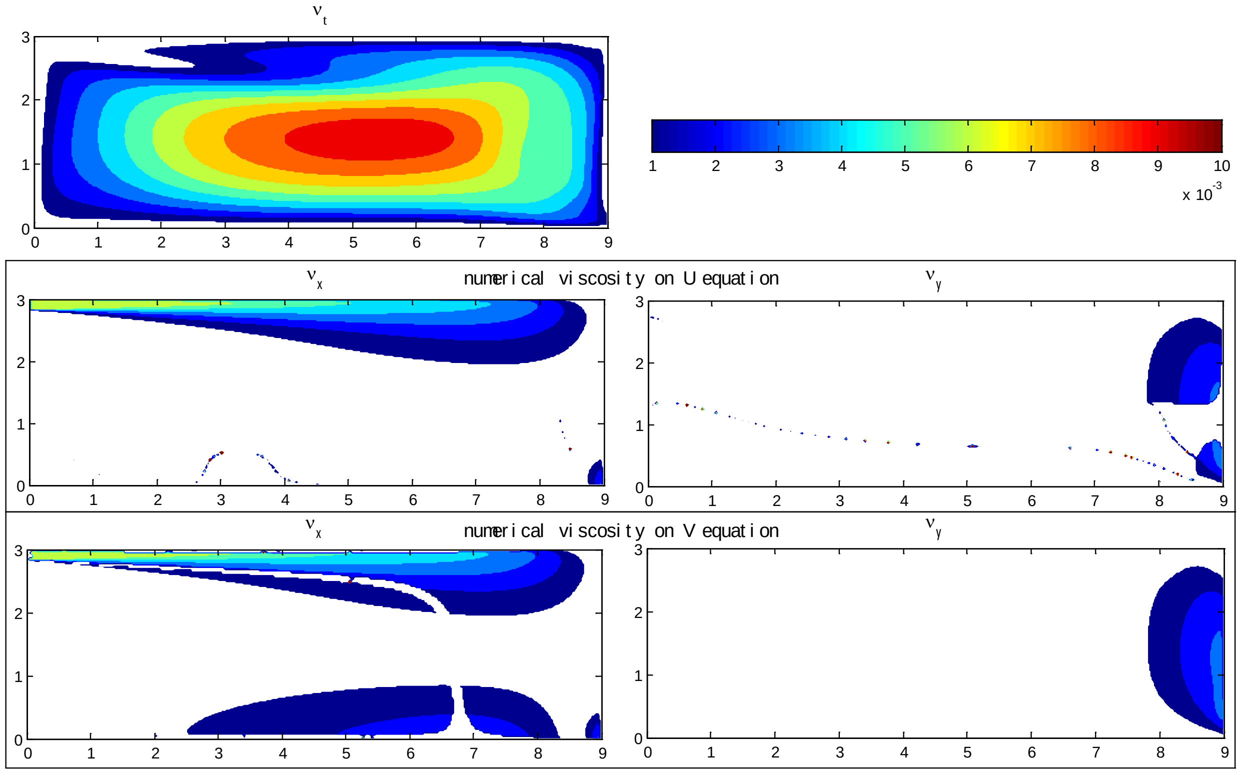


Figure 11. Comparison of numerical viscosity (grid-independent #3) with turbulence viscosity for the FC case

4.2.3 Mixed Convection

Mixed convection has the combined mechanisms of NC and FC. The flow is driven by both external force and temperature-difference-induced buoyancy force. The MC case studied is a slot ventilated cavity [106] as shown in Figure 12. The floor of the cavity is heated to a uniform temperature of 35.5°C. The inlet air and other walls were maintained at a cooler temperature of 15°C. Measurements were made along the mid-height and mid-width of the space. Along the horizontal line, the vertical velocity component (v) and temperature were measured; along the vertical line, the horizontal velocity component (u) and temperature were measured.

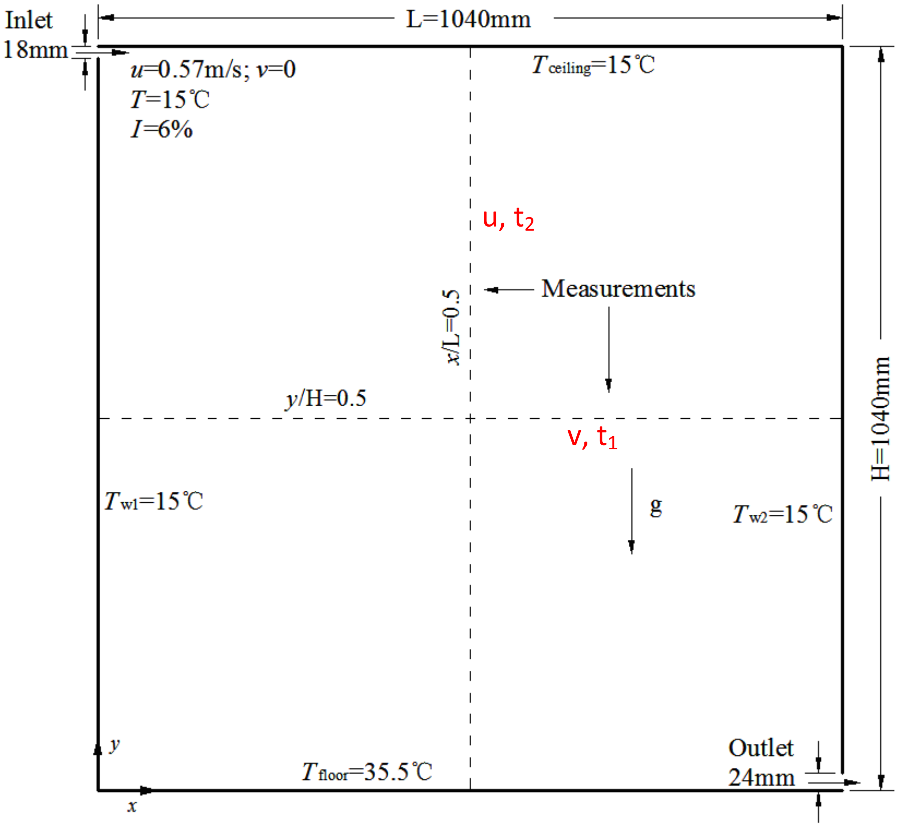


Figure 12. Configuration, boundary condition setup and measurements of MC Case

The QUICK scheme is used for this case. Eq. (4.10) and (4.11) are used for the normalization of measured variables (temperature and velocity).

$$T = \frac{t - t_w}{t_f - t_w} \tag{4.10}$$

$$U = \frac{u}{u_{inlet}}; \; V = \frac{v}{u_{inlet}} \tag{4.11}$$

where t_f and t_w are the floor and wall temperatures, and t is the predicted temperature on each point in the domain. Velocity can also be normalized by the inlet velocity $u = 0.57m/s$, but this will not influence the RMSE result.

Table 5 summarizes the grid independency study result using the normalized RMSE method by doubling the total grid number of each simulation. Figure 13 plots the profiles of measured and predicted velocities and temperatures. The main visible prediction difference occurs for the temperature (*T1*) distribution along the mid-height horizontal line, where grid #1 and #2 results have a large discrepancy compared to the finer grid results. This is clearly reflected by the normalized RMSE judgment matrix. Using the 10% criterion, one can identify that grid #3 and finer can be considered to be grid independent.

Table 5. Grid resolutions and normalized RMSE results for MC case

Grid index	Grid number (X*Y)	Normalized RMSE to next finer grid			
		V	*U*	*T1*	*T2*
#1	52*52	0.2217	0.2491	0.1521	0.0519
#2	74*74	0.0738	0.1207	0.1805	0.1346
#3	104*104	0.0697	0.0656	0.0515	0.0436
#4	147*147	0.0803	0.0663	0.0217	0.0288
#5	208*208	N/A	N/A	N/A	N/A

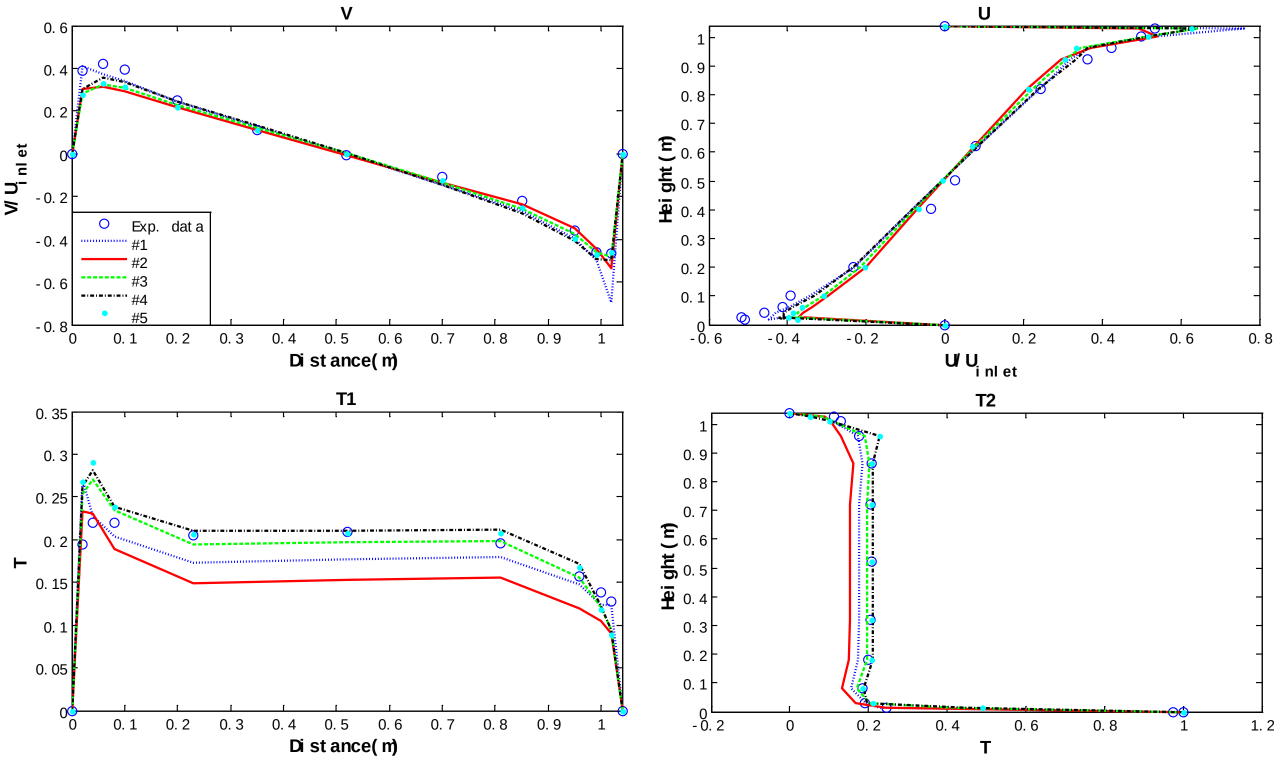


Figure 13. Profile comparison of predictions under different grid resolutions for the MC case

Figure 14 provides the comparison of the calculated turbulent viscosity with the numerical viscosity of the QUICK scheme from the grid-independent solution. The left contour shows the turbulent viscosity as the reference, and right contours are the numerical viscosity on the *U* (horizontal velocity) equation and the *V* (vertical velocity) equation, respectively. The result verifies that the numerical viscosity is much smaller than the turbulent viscosity in the whole computational domain. The finding agrees with the conclusion from the RMSE judgment.

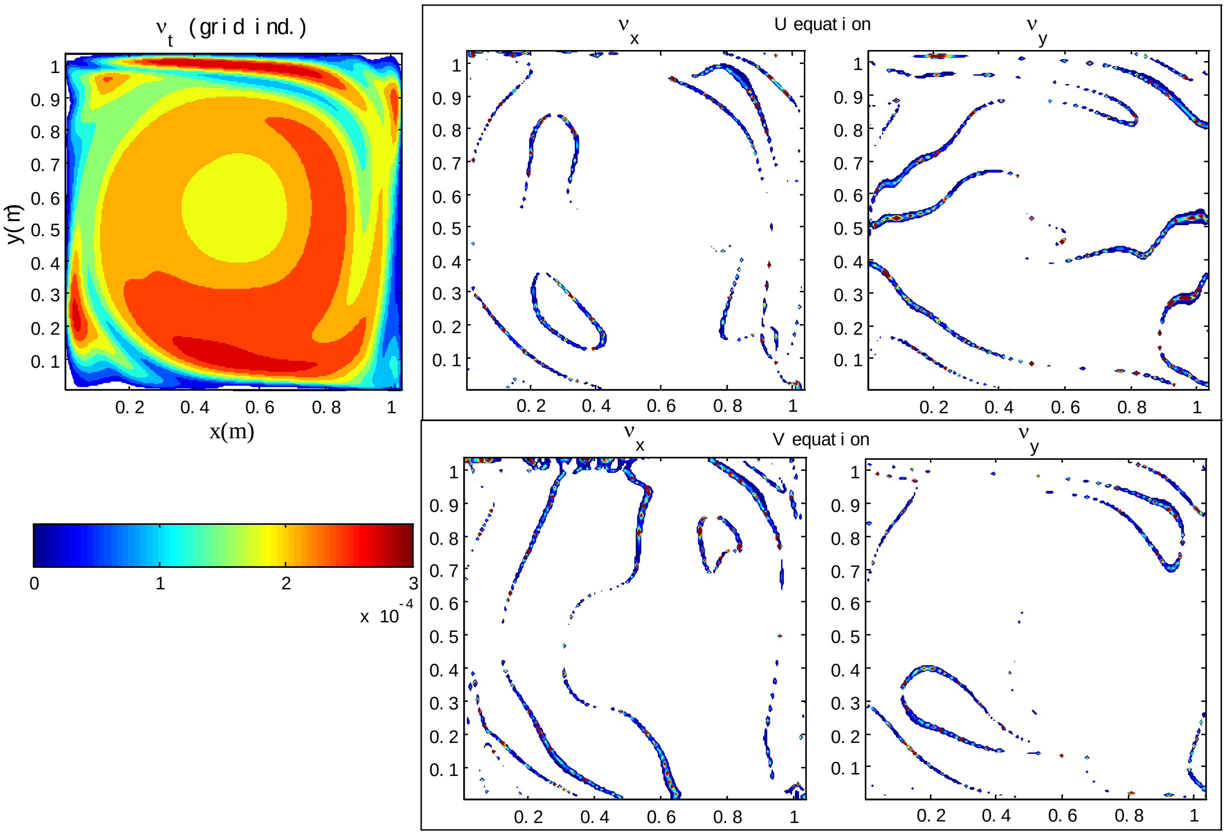


Figure 14. Comparison of numerical viscosity (grid independent #4) with turbulence viscosity for the MC case. Left – turbulent viscosity; Right – numerical viscosity on U equation (top) and V equation (bottom)

4.2.4 Three Dimensional Office

Three two-dimensional cases covered typical indoor environmental airflow mechanisms are used to validate the proposed method for grid independency study. In order to test the effectiveness of the proposed method for a complex indoor environment, a three-dimensional (3D) side-wall supply displacement ventilation case [107] was used. The room was 5.16m×3.65m×2.7m as shown in Figure 15. The displacement ventilation diffuser provided a ventilation rate of 4 ACH through the perforated front panel with a net area ratio of 10%. The equivalent air velocity through the front panel was 0.086m/s. The supply and exhaust air temperatures were, respectively, 17°C and 21.6°C. The detailed dimension and heat flux of each object in the room can be found in the ASHRAE research report 949 [107]. The experiment measured the air velocity and temperature along 9 vertical poles as shown in Figure 16.

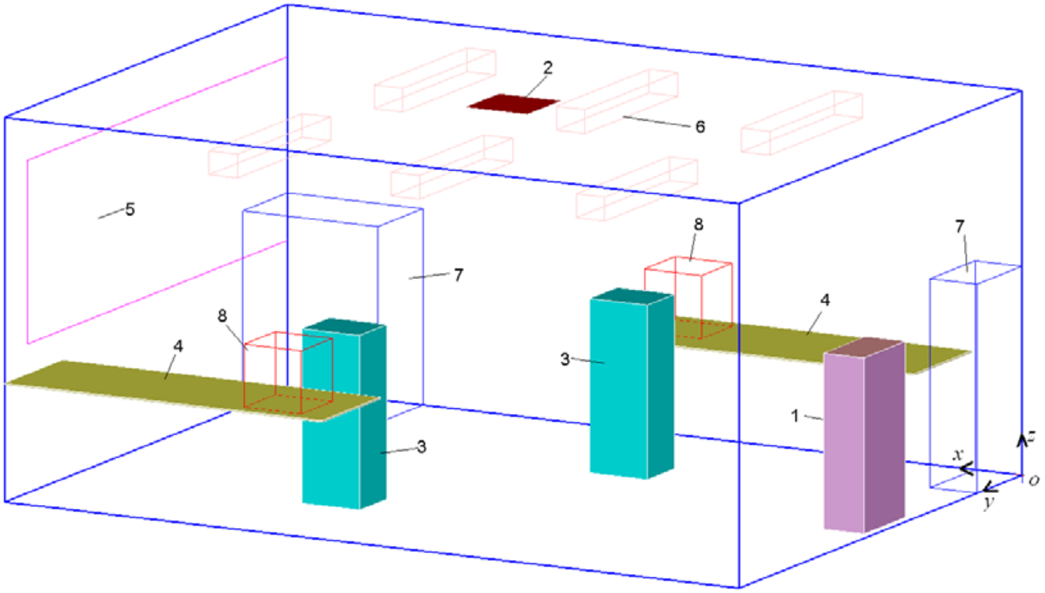


Figure 15. Configuration of displacement ventilation office case

(1. Inlet 2. Outlet 3. Person 4. Table 5. Window 6. Lamp 7. Cabinet 8. Computer)

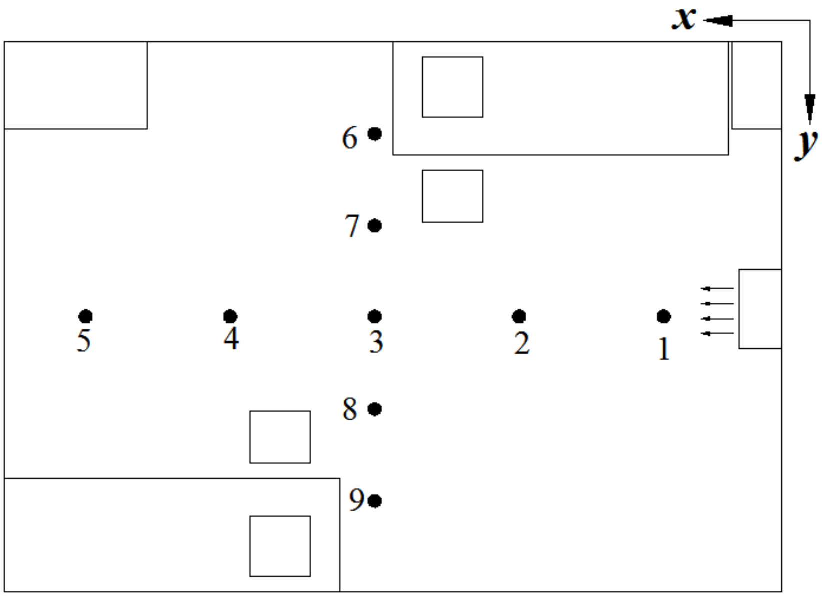


Figure 16: floor plan of the positions of parameter profile comparison, experimental data including velocity and temperature is available along vertical poles of P1 through P9.

The dimensionless temperature is defined as:

$$T = \frac{t - t_i}{t_o - t_i} \tag{4.12}$$

where t_i and t_o are the inlet and exhaust temperatures, and t is the predicted temperature on each mesh point in the domain in °C. Velocity can also be normalized by the equivalent inlet velocity $u = 0.068 m/s$, but it has no influence on the RMSE calculation.

Table 6 summarizes the grid study results using the normalized RMSE method by doubling the total grid number in each simulation. Figure 17 and Figure 18 present the profile plots of the dimensionless velocity and temperature, respectively. Grid #1, #2 and #3 results have a large discrepancy from those with the finer grids, as revealed by both the normalized RMSE table and the profile comparison.

Table 6. Grid resolutions and normalized RMSE results for the 3D office case

		Normalized RMSE to next finer grid			
Grid index	Grid number (X*Y*Z)	V-stream wise	V-cross section	T-stream wise	T-cross section
#1	62*42*27	0.1937	0.3481	0.0164	0.0276
#2	78*54*34	0.1828	0.1313	0.0083	0.0109
#3	98*68*43	0.1436	0.2089	0.0154	0.0142
#4	123*86*54	0.1019	0.1262	0.0082	0.0096
#5	158*108*68	N/A	N/A	N/A	N/A

It is noticed that beside the grid size, the RMSE value is also very sensitive to the simulation convergence status. It is common that convergence of simulation with a fine grid needs more efforts than that with a coarser grid due to computing instability issues and accumulation of numerical errors on all mesh points. For this 3D room case, although the relaxation factors for the simulation were carefully chosen for the finer grid than #5, the criterion of 10% can still not be achieved. As the RMSE difference between grids #4 and #5 is around 10%, grid #4 is considered to be grid independent. This 10% criterion can be used for the general guidance but can certainly be adjusted based on specific needs and conditions. The further numerical viscosity analysis shown in Figure 19 confirms that the difference between #4 and #5 is mostly from other numerical factors instead of the grid-induced error.

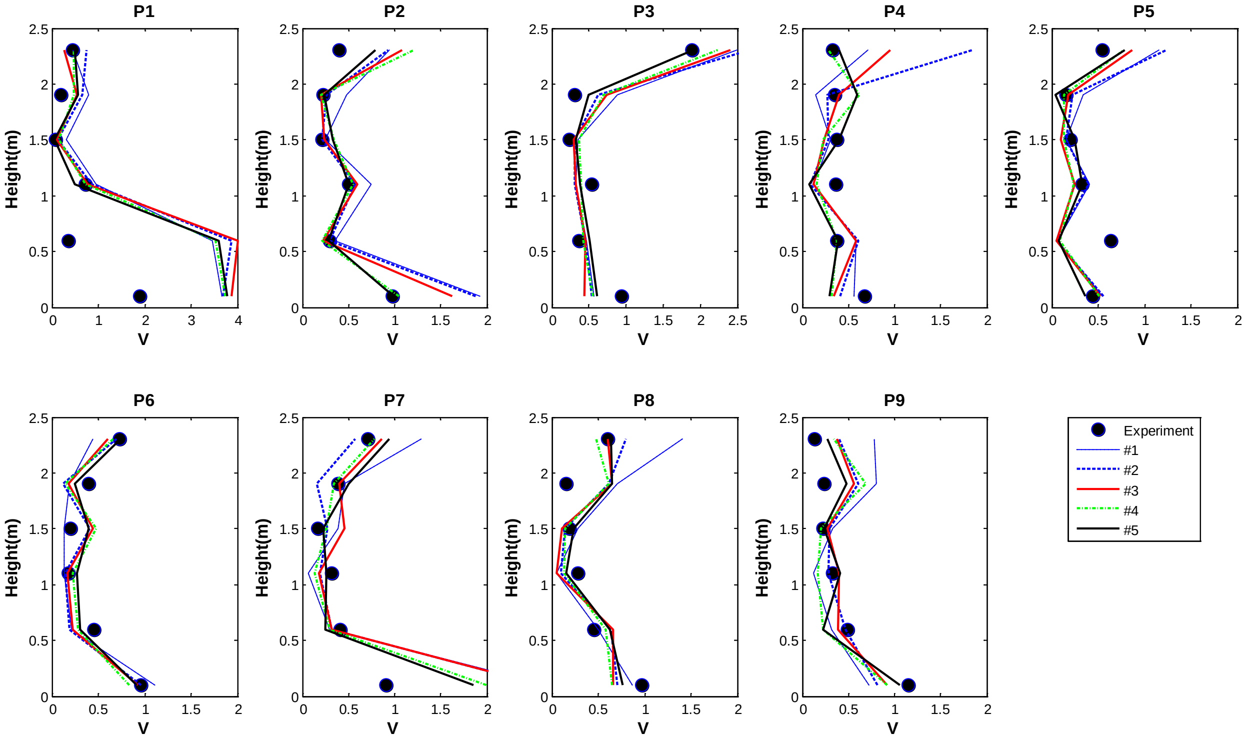


Figure 17. Dimensionless velocity profile predictions upon different grid resolutions for the 3D office case

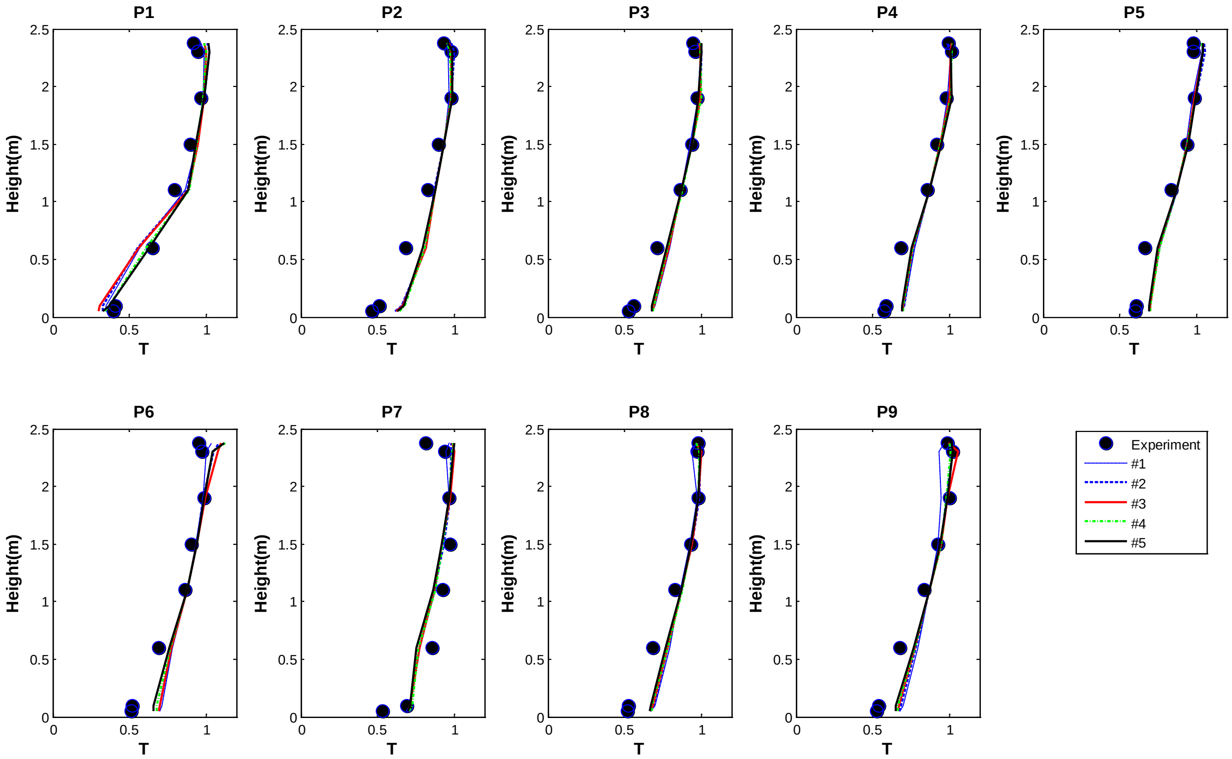


Figure 18. Dimensionless temperature profile predictions upon different grid resolutions for the 3D office case

Figure 19 presents the comparison of the turbulent viscosity with the numerical viscosity of the QUICK scheme for the grid-independent (#4) result. The contour is for the intersection at mid-height of the room. The top contour is the turbulent viscosity, and the bottom contours show the numerical viscosity on the U (horizontal velocity) equation and the V (vertical velocity) equation. The result verifies that the numerical viscosity is much smaller than the turbulent viscosity in grid #4, indicating the grid-induced error is mainly eliminated. This result agrees with the conclusion from the RMSE and profile plot analysis.

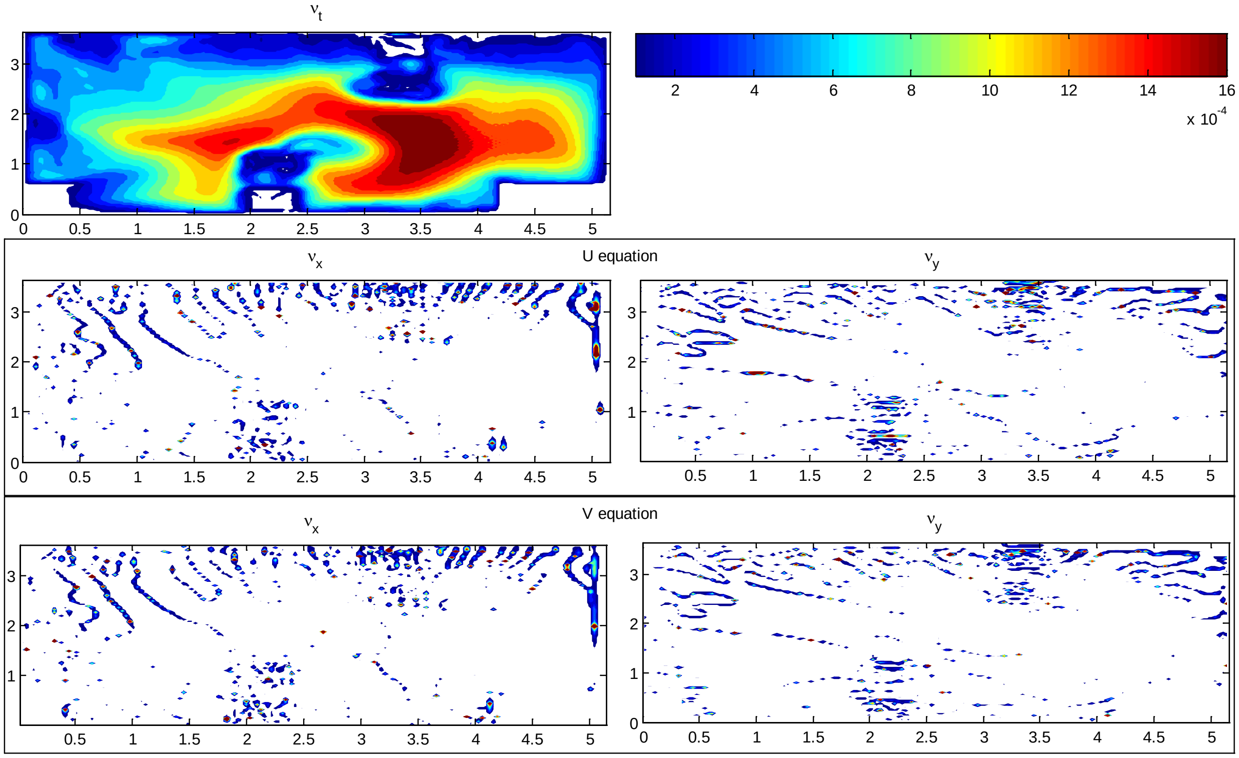


Figure 19. Comparison of turbulence viscosity with numerical viscosity for the 3D office case (grid independent #4). Top – turbulent viscosity; Bottom – numerical viscosity on U equation (top) and V equation (bottom)

4.2.5 Conclusive Summary

This section focuses only on the uniformly distributed grid to testify the proposed method for CFD grid independency study. For non-uniform grids, which are common in CFD practice, the RMSE and numerical viscosity analysis are still applicable. Commercial CFD software can incorporate the function of calculating numerical viscosity into the post-process of the tools, allowing users to evaluate the grid independency of simulation results.

It is noted that the turbulent viscosity in the FC case is about two magnitudes greater than it is in the NC and MC cases. This indicates that FC flows have greater tolerance to large grid size if using the same order of differencing scheme. In other words, for non-isothermal cases, the grid size to reach grid independency will be much smaller than that under iso-thermal conditions. This conclusion can be consolidated if further study verifies that non-isothermal conditions usually lead to smaller turbulent viscosity than isothermal conditions in built environments.

Equation (4.4) reveals that lower order schemes will cause greater coefficients for the normalized RMSE. Hence, if using the same 10% criterion, the lower order differencing scheme will require more grid numbers (or smaller grid size) to reach the same RMSE value. This agrees with the general CFD principle that a higher order differencing scheme leads to a smaller numerical viscosity and thus requires less grid to reach grid independency.

For differencing schemes of second order and higher, calculation of numerical viscosity requires a division by a second order derivative term (diffusion). Take CDS for example, the actual truncation error (TE) term is (as in Table 2.)

$$TE = -\frac{|u_i| \cdot \Delta x_i^2}{6} \frac{\partial^3 u_i}{\partial x_j^3} \tag{4.13}$$

and the numerical viscosity is TE divided by the second order derivative term

$$\nu_{num} = -\frac{TE}{\frac{\partial^2 u_i}{\partial x_j^2}} \tag{4.14}$$

With very small local diffusion terms in the domain as the denominator, the numerical viscosity at these local points may become very big (or even infinite). This can be observed in the contour plots of numerical viscosity for the cases with the CDS and QUICK schemes, where those small dark dots appear. However, this will not bring much concern in the actual simulation because the total error that numerical viscosity brings is the whole truncation error term.

A normalized RMSE value presents a great opportunity of using a single index to evaluate the overall closeness of CFD simulation results upon different grid resolutions. It becomes an effective and convenient criterion to judge the grid independency status for indoor environment CFD modeling with less demand on knowledge of numerical fundamentals. A typical comparison in CFD grid independency study usually requires doubling (or halving) the total grid numbers between two simulations. The prediction difference, if under 10% for the normalized RMSE, between a grid and its immediate finer grid may be considered negligible, indicating the reach of a grid-independent solution. In practice, because it is quite expensive thus not common to test a series of many grid resolutions, numerical viscosity analysis can be used to check the status of grid independency. Numerical viscosity study provides the fundamental insight into grid independency conditions.

In indoor environment modeling, to reach a solution's grid independency, numerical viscosity magnitude in the entire domain should be much smaller than corresponding local turbulence viscosity so that the effect of numerical viscosity can be neglected.

4.3 Numerical Experiment on Coarse Grid Possibilities

According to the numerical analysis, discretization induced numerical error ends up with artificial diffusion terms on each direction, which are determined by both numerical viscosity values and the magnitude of second order derivative terms. Such second order derivative terms can be different magnitudes, which provide an opportunity of manipulating the CFD grid size along different directions without significantly increasing grid-induced error. This hypothesis is tested in the following section with consideration of flow mechanism in typical IEQ study. Since the RNG k-ε model [43] has the overall best performance among different RANS models [87, 88], this turbulence model is adopted for all tests in this section.

4.3.1 Natural Convection Validation

The same two-dimensional (2D) natural convection experiment [104] conducted originally for CFD code validation purposes in section 4.2.1 is used to verify the hypothesis in the theoretical analysis part. Figure 6 shows the configuration of the experiment. Two vertical walls are heated and cooled respectively to keep uniform temperatures; the two horizontal walls each have temperature distribution that can be discretized to ten small pieces. Uniform temperature can be used as the boundary condition for each piece. A clockwise-circulated flow inside the cavity is formed due to buoyancy. Detailed measurement data of temperature and velocity are available along the central height and central width lines. Along mid-height line, vertical velocity component (v) and temperature data (t1) are available, while along mid-width line, temperature data (t2) is available for further validation. The LDA used for velocity measurement has an accuracy of 0.07% while the accuracy of the thermal couple probe for space temperature measurement is ±0.5°C.

To ensure the comparability of difference in temperature with that in velocity, the absolute temperature is normalized as

$$T = \frac{t - t_c}{t_h - t_c} \tag{4.15}$$

where t_h and t_c are hot and cold wall temperatures, and t is the predicted temperature on each measurement location. Velocity values can also be normalized by dividing a reference velocity, but it makes no difference to the RMSE value.

The investigation starts from testing a series of uniformly distributed grid resolutions, with total grid number doubled each time. Table 7 summarizes the grid resolution used.

Table 7. Grid resolutions tested against measurements for NC case

Grid index	#1	#2	#3	#4	#5	#6	#7	#8
Grid # (XY)	10x6	10x12	16x16	23x23	31x31	44x44	63x63	88x88
Grid index	#9	#10	#11	#12	#13	#14	#15	
Grid # (XY)	125x125	177x177	250x250	354x354	500x500	707x707	1000x1000	

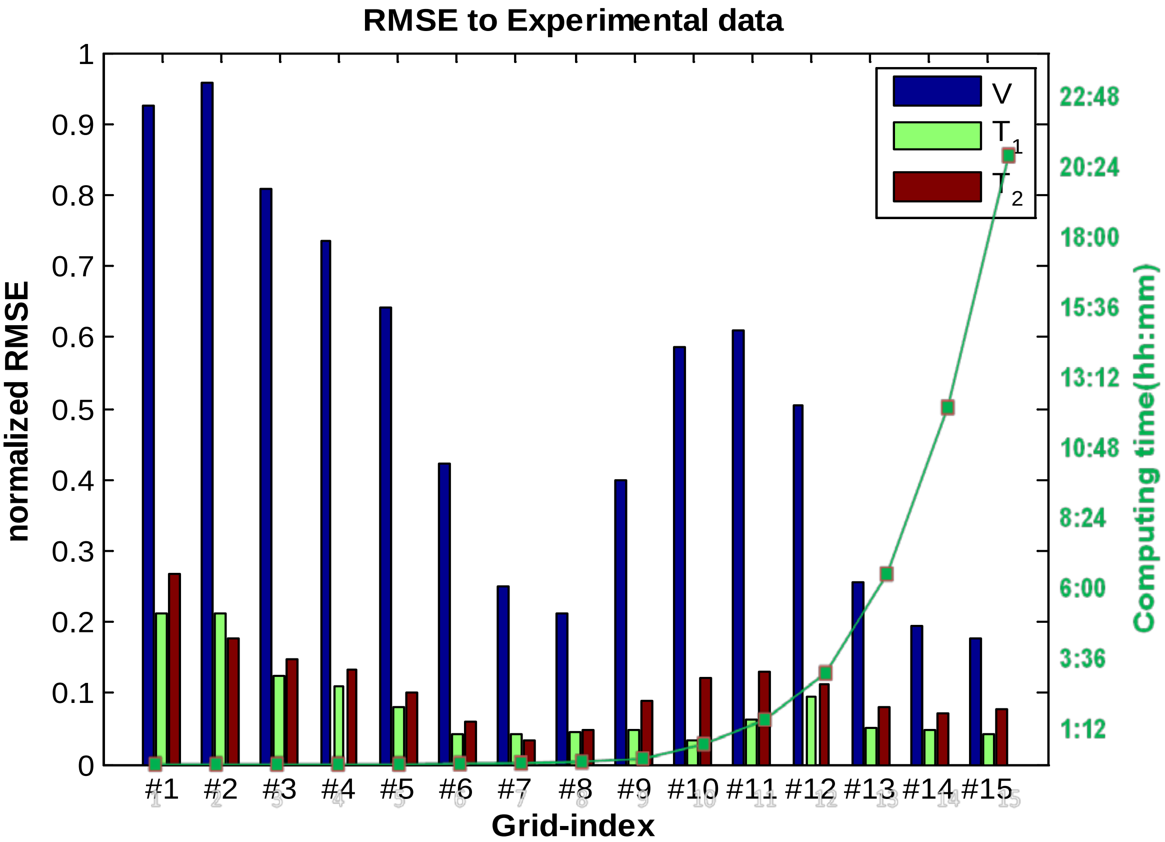
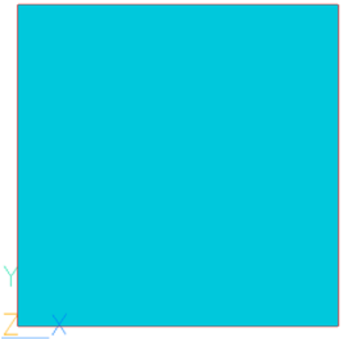
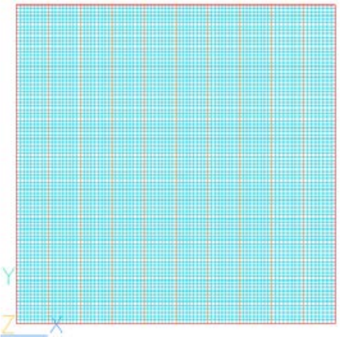
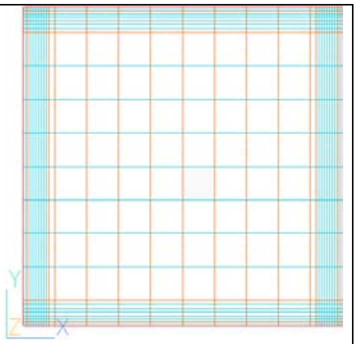


Figure 20. Performance of different uniform grid resolutions for NC Case

The result of each grid resolution is compared with experimental data through a normalized RMSE index (Eq.4.5) to evaluate the performance of each grid resolution as in Figure 20. The result shows that finer grid resolution generally leads to better performance but consumes more computing time. However, due to the inherit deficiency of CFD model, coarse grid can have better performance than its finer counterparts. Coarse grid such as grid #7 and #8 consumes much less computing resource than fine grid but has better performance than some finer grid resolutions.

Starting from grid #7 or #8, which is called “promising uniform” coarse grid, local grid refinement and reduction are conducted. The goal of such investigation is finding out the optimized trade-off between simulation accuracy and computing cost. Therefore, the performance evaluated by the normalized RMSE index as well as the computing cost is compared with each other.

Table 8. Comparison of different grid resolutions for NC case

Grid distribution			
Computing cost	t=100%	t=1.91%	t=0.44%
Grid resolution	707×707 (grid independent)	88×88 (promising uniform)	30×24 (optimized coarse)
RMSE V	0.1952	0.2120	0.2210
RMSE T1	0.0484	0.0321	0.0444
RMSE T2	0.0716	0.0330	0.0362

Finally an optimized grid resolution, which balances between the computing cost and simulation accuracy, is found out. The grid distributions of grid independent (uniform), promising uniform coarse grids, as well as the optimized coarse grid, are summarized in Table 8. Near the vertical walls, local refinement on grid size is adopted; the grid size in the center area of the domain is around 1/10 of computational domain size.

Under each figure, “t” is the relative computing time under the same computer platform, assuming the grid independent solution consumes one unit of time. The optimized coarse grid consumes less than one percent of the computing time of the grid independent solution.

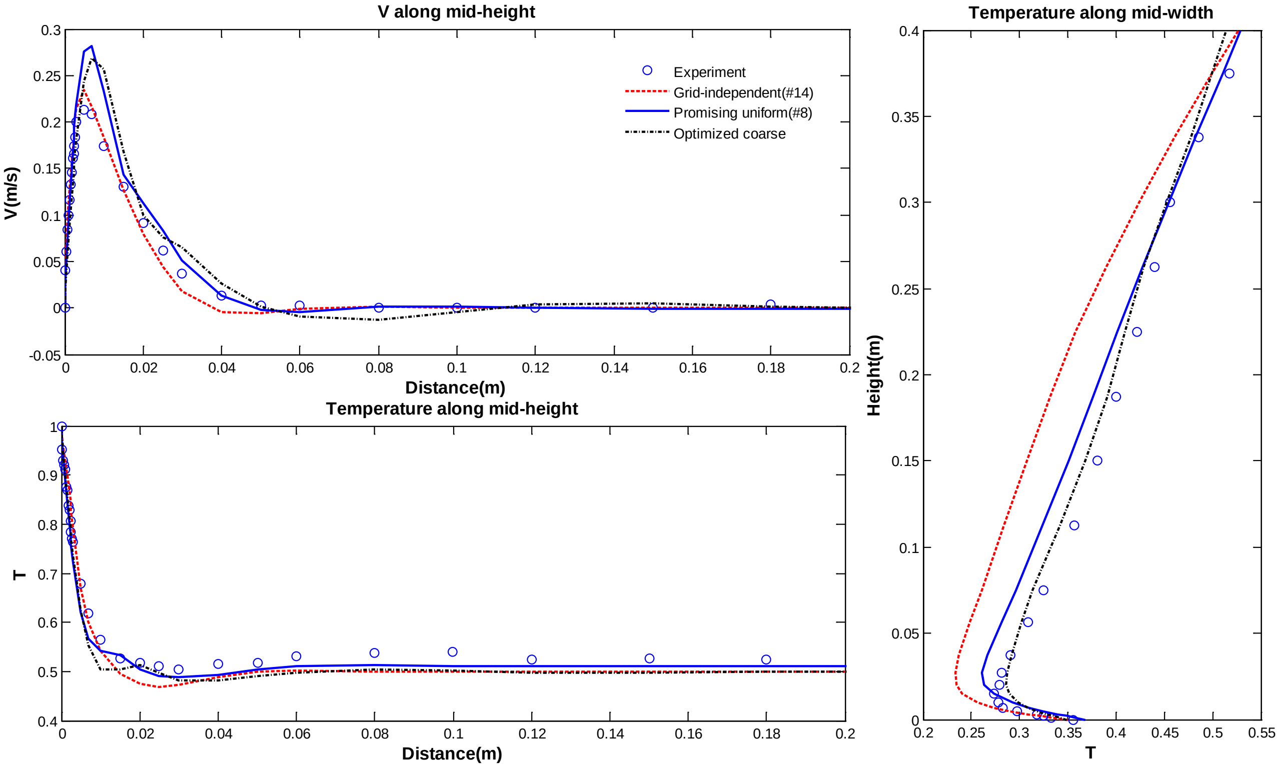


Figure 21. Profile comparison of solutions of different grids against experimental data for NC Case

Figure 21 shows the performance of the three grid resolutions in comparison to the normalized RMSE value and profile. They indicate that the optimized coarse grid, which consumes less than one percent of the computing time than the grid independent one, has a comparable performance. The velocity prediction is slightly farer away from measured data, and the temperature is even slightly better than grid independent one. Coarse grids can bring numerical error, which is a main source of the difference between numerical solutions and exact solution. However, since factors such as turbulence model and boundary conditions also have uncertainties built in, the numerical error introduced by coarse grid has the possibility of accumulation on the total error or to cancel out part of the error introduced by the other factors.

In Figure 22, the numerical viscosity distribution is visualized. Comparing the contour of numerical viscosity from grid independent and optimized coarse grids, obviously coarse grid introduces much more numerical viscosity than the grid independent one. Include the turbulence viscosity into this comparison, it is clear that grid independent solutions only introduce a negligible numerical viscosity, while the optimized coarse grids introduce numerical viscosity that has values of similar or even higher order of magnitude, which indicates a relatively large numerical error.

As for the predicted result part, the coarse grid turns out to have a comparable performance as the grid independent one.

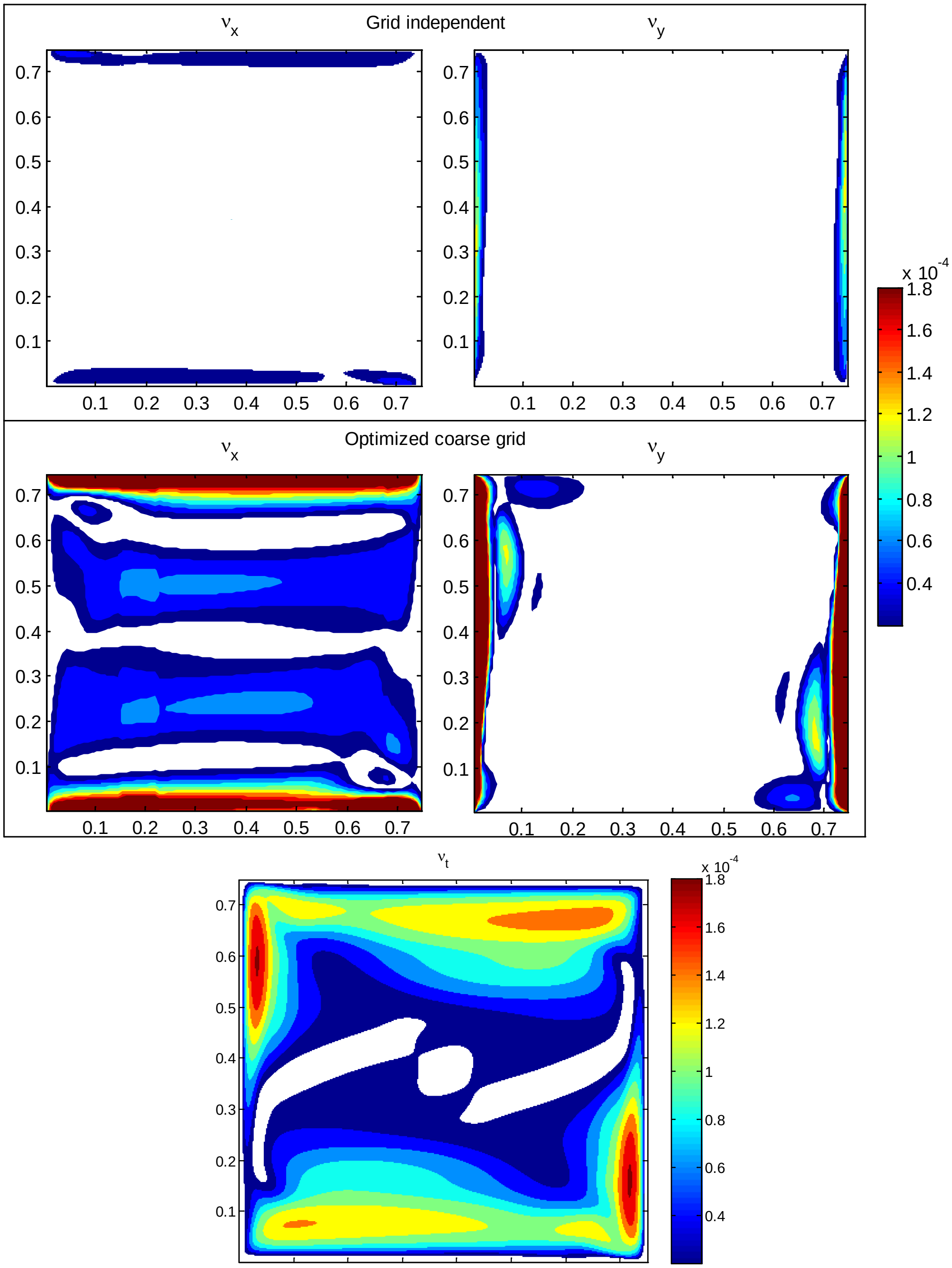


Figure 22. Comparison of numerical viscosity of fine and coarse grid and turbulent viscosity in the computational domain for NC case

To investigate the fundamental reason for such an outcome, the second order derivative terms (diffusion) are visualized in Figure 23, using the optimized coarse grid result. Note that since the simulation result of this grid is very close to the grid independent one, the diffusion distribution in the domain for the grid independent result is almost the same.

The result shows that along the top and bottom walls of the cavity, diffusion terms have an order of magnitude difference $(\partial^2 u)/(\partial x^2) \ll (\partial^2 u)/(\partial y^2)$, so that even a large value of ν_x, in other words, a large grid size for X direction, will not cause much error; similarly, along the two vertical walls, $(\partial^2 v)/(\partial x^2) \gg (\partial^2 v)/(\partial y^2)$, indicates a large tolerance on grid size for Y direction within this region. The optimized coarse grid distribution occurs to have such character as in Figure 22, explaining the outcome that coarse grid solution is not significantly affected by the numerical error introduced by grid size.

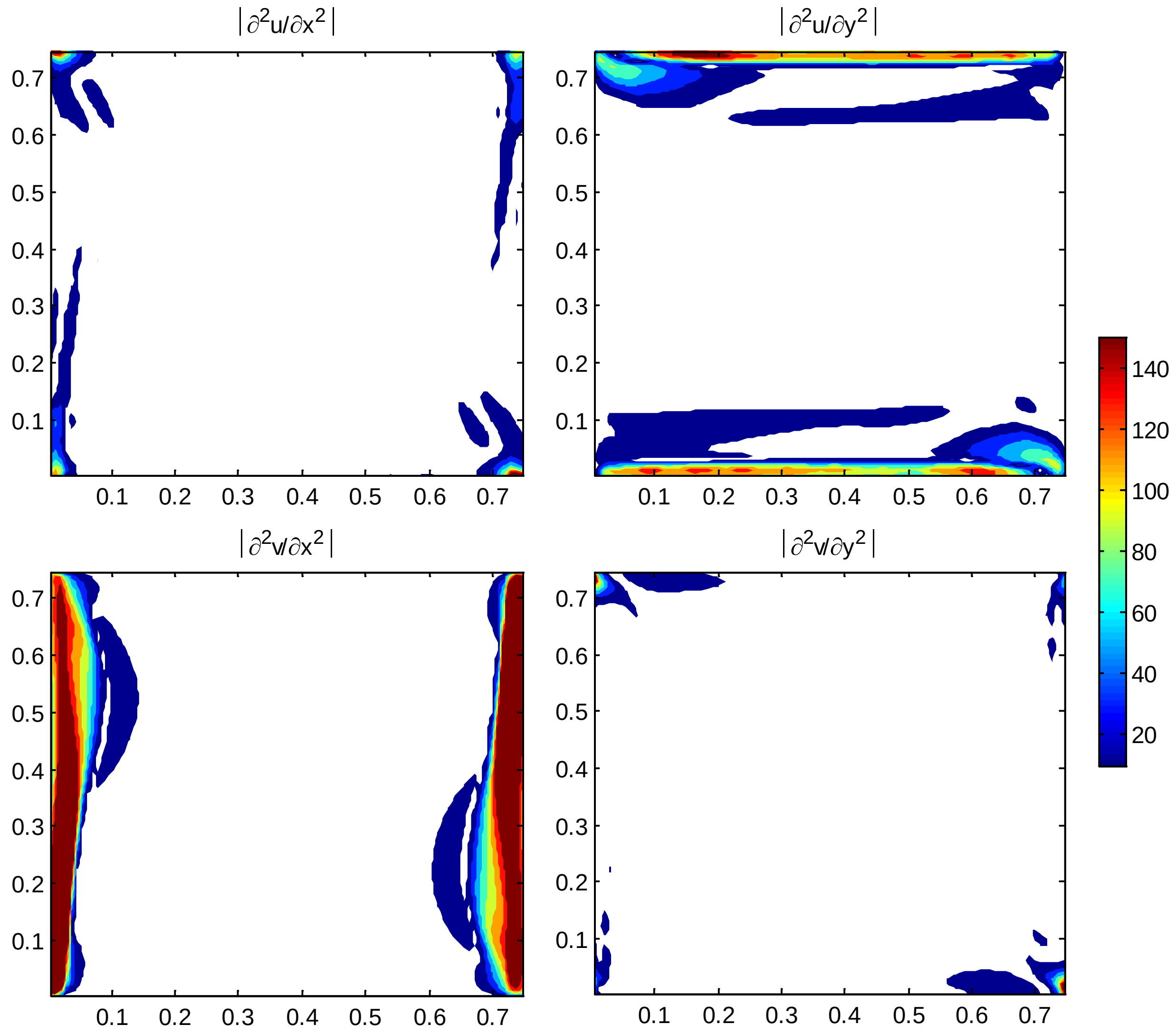


Figure 23. Second order derivative terms of momentum equation for NC case

To further confirm the analysis, Figure 24 shows a total grid-induced error term for the grid independent solution and optimized coarse grid. Although coarse grid introduces a relatively large numerical viscosity, the total error introduced by space discretization is almost the same for the coarse and fine grids, except for the region near the corner of the cavity. The investigation on this NC case confirms the hypothesis in the methodology part.

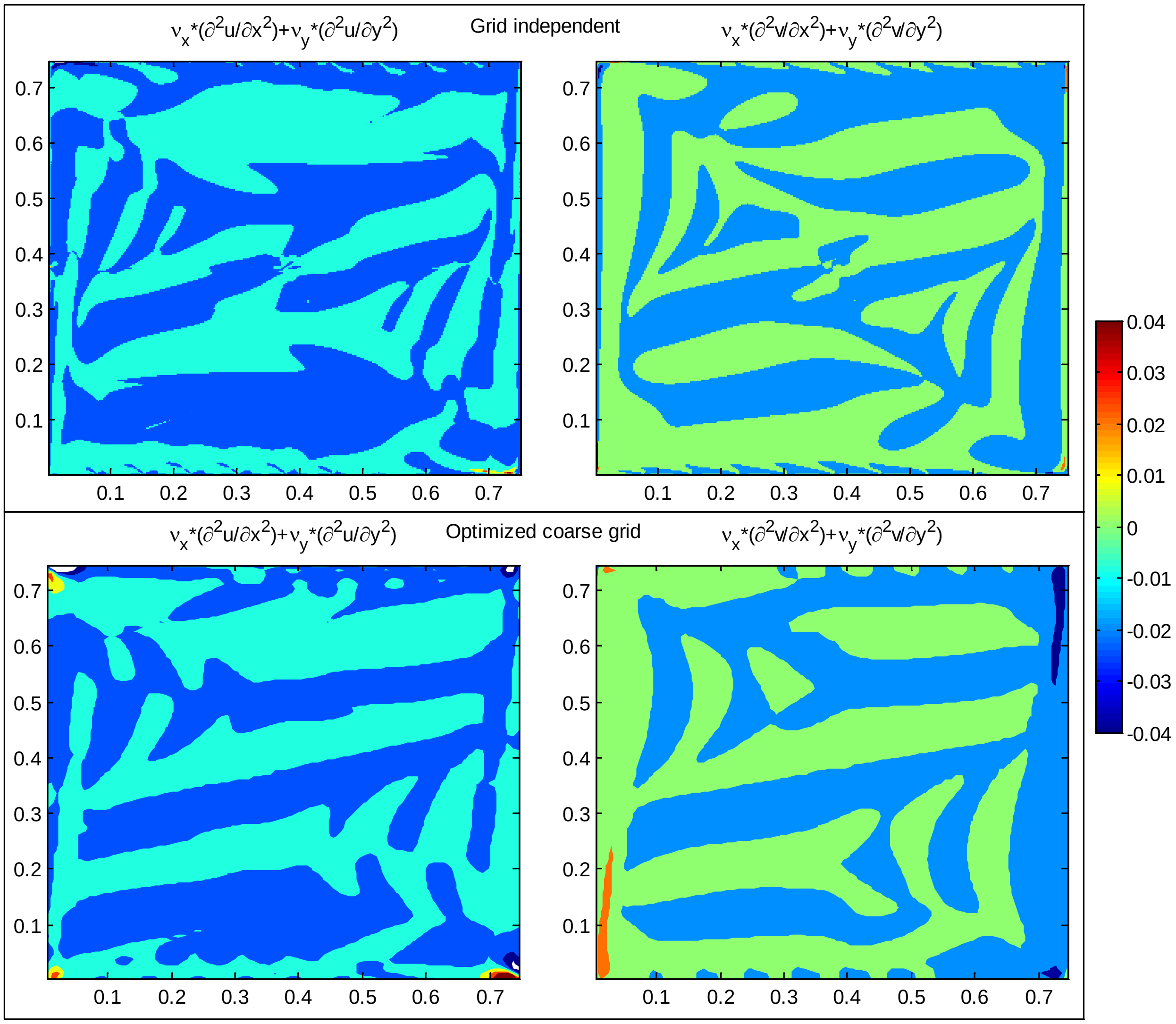


Figure 24. Comparison of total grid-induced error for NC case

4.3.2 Forced Convection Validation

The same FC case as in 4.2.2 has been used in this validation. To compare the difference between the CFD prediction and measurement, the velocity values are dimensionless through dividing by the inlet velocity as

$$U = \frac{u}{u_{inlet}} = \frac{u}{0.455} \tag{4.16}$$

The hybrid scheme was used in this FC case. The grid resolution, similarly, is doubled each time to test a series of uniformly distributed grid resolutions to identify a promising uniform coarse grid as a starting point to optimize grid distribution. Table 9 presents the uniform grid resolution used in this test; Figure 25 illustrates the performance of each grid resolution by comparison to experimental data using normalized RMSE, as well as the computing time of each grid resolution consumes.

Table 9. Grid resolutions tested against measurements for FC Case

Grid index	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**
Grid # (XY)	9x3	13x5	19x6	26x9	37x13	53x18	75x25
Grid index	**#8**	**#9**	**#10**	**#11**			
Grid # (XY)	106x36	150x50	212x71	300x100			

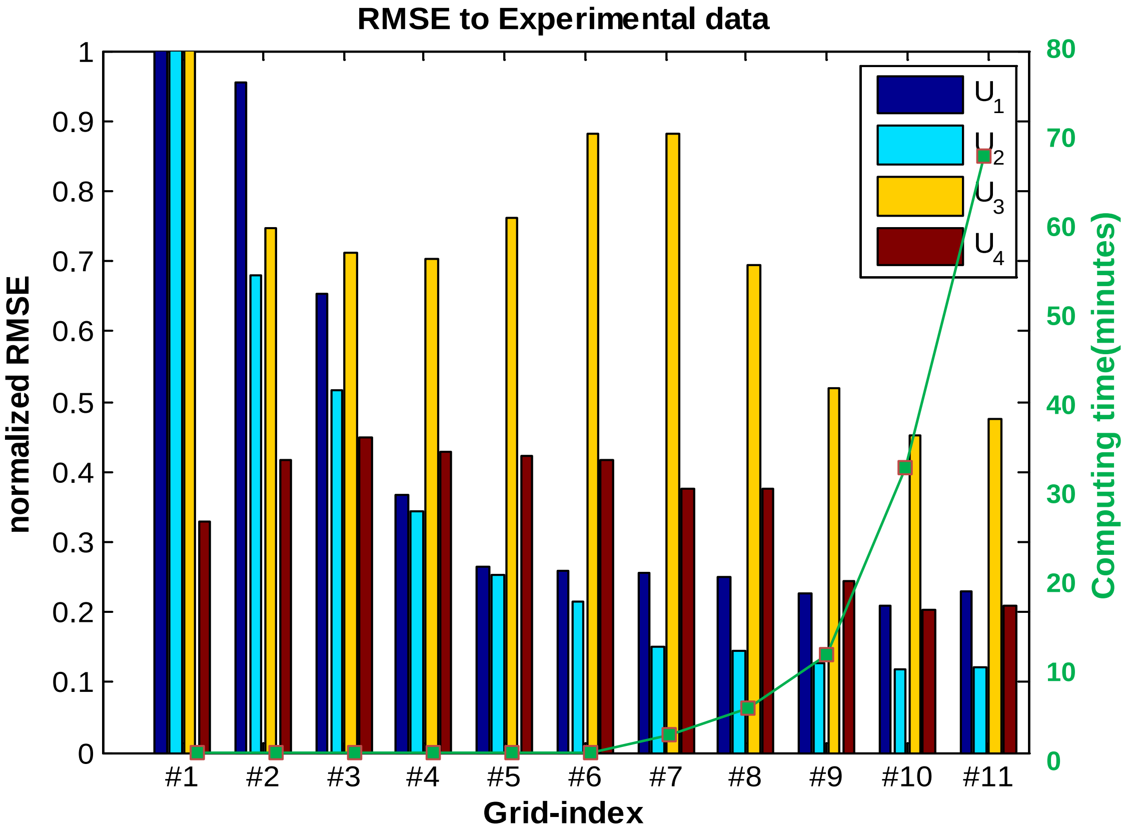


Figure 25. Performance of different uniform grid resolutions for FC Case

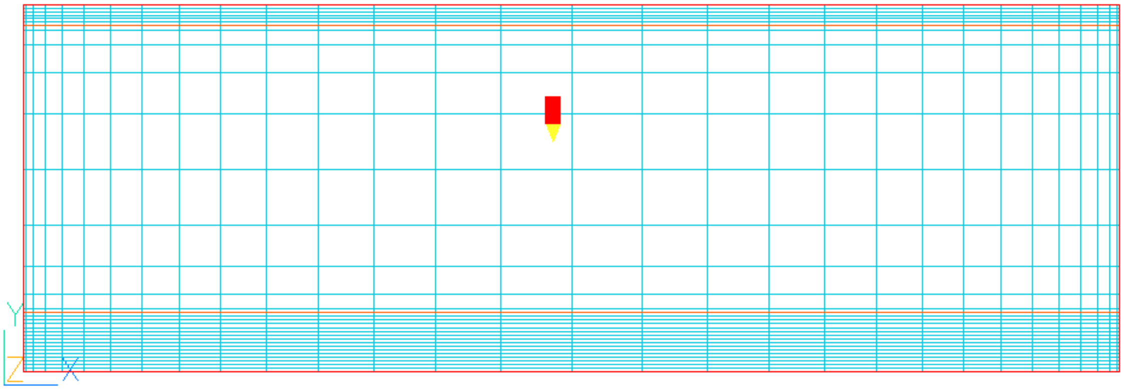


Figure 26. Distribution of optimized coarse grid for FC case

Figure 26 shows the grid distribution of the optimized coarse grid. Local refinement is specified in the normal direction of inlet and outlet flow as well as boundaries. Other than that, the grid size is around 1/10 of the length scale of geometry, which is the height of the cavity.

Table 10. Grid resolutions and normalized RMSE results for FC Case

Grid index	Grid number(X*Y)	Computing cost	Normalized RMSE compare to experimental data			
			U1	U2	U3	U4
Grid independent	300*100	t = 100%	0.2096	0.1181	0.4510	0.2033
Optimized coarse	30*32	t = 5.8%	0.2088	0.1149	0.3890	0.1583

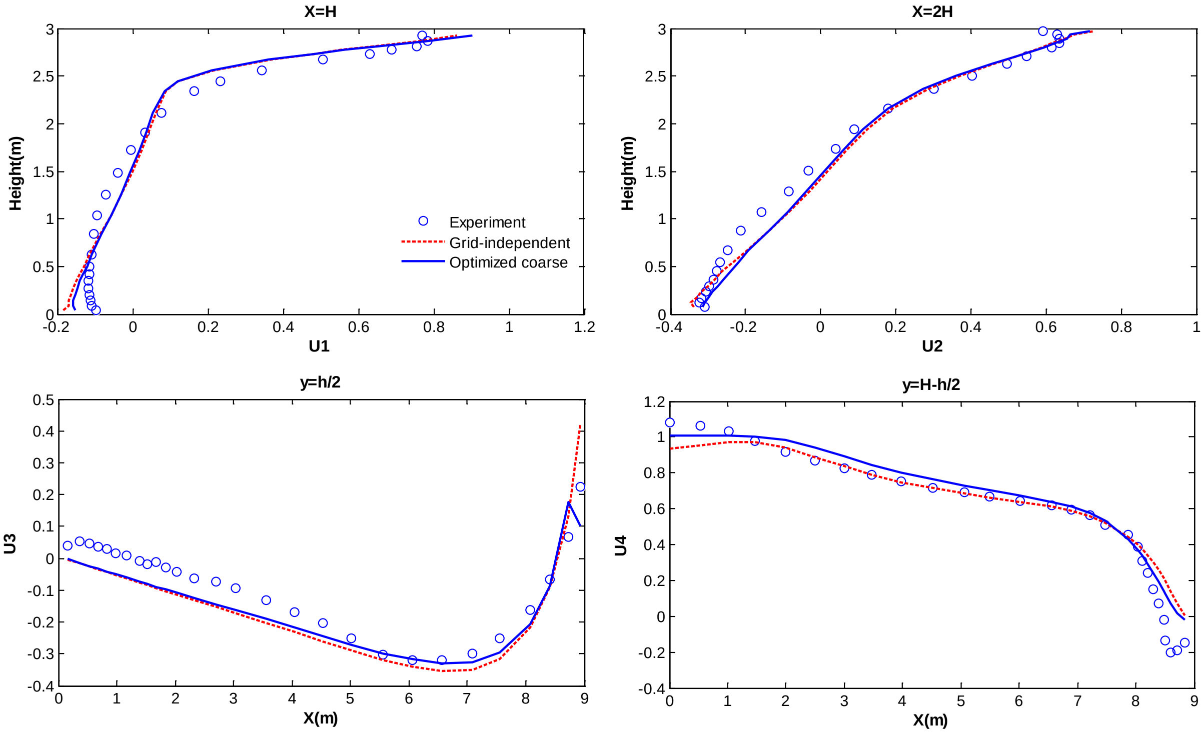


Figure 27. Profile comparison of solutions of fine and coarse grid against experimental data for FC Case

Table 10 shows the result of the grid independent and optimized coarse grid comparison in terms of a normalized RMSE index. With only 5.8% of the grid independent computing cost, the optimized coarse grid gives an almost identical prediction as the grid independent one. This can be further confirmed in Figure 27, which presents the the profile plot of CFD solutions against experimental data.

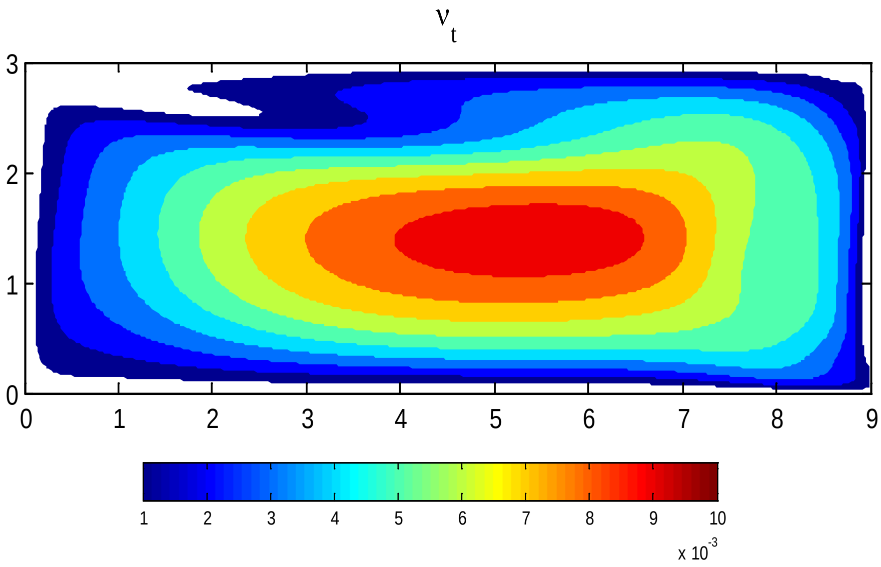


Figure 28. Turbulent viscosity of grid-independent solution of FC Case

The turbulence viscosity of grid independent solution of the FC case is as Figure 28. The maximum value is at the magnitude of 0.01. To evaluate the effect of grid size on the prediction accuracy, Figure 29 shows the contour of numerical viscosity for both fine and coarse grids in the computational domain using the same color range. It shows that the grid independent solution has a negligible value that "totally" eliminates the numerical viscosity, but the coarse grid has a numerical viscosity comparable and even much greater than turbulent viscosity. It needs to be pointed out that even the grid independent one has ν_x slightly greater than turbulence viscosity near the ceiling adjacent to the inlet. Ideally, the grid needs to be further refined to achieve grid independency. But for most of the computational region, the effect of numerical viscosity has been completely eliminated.

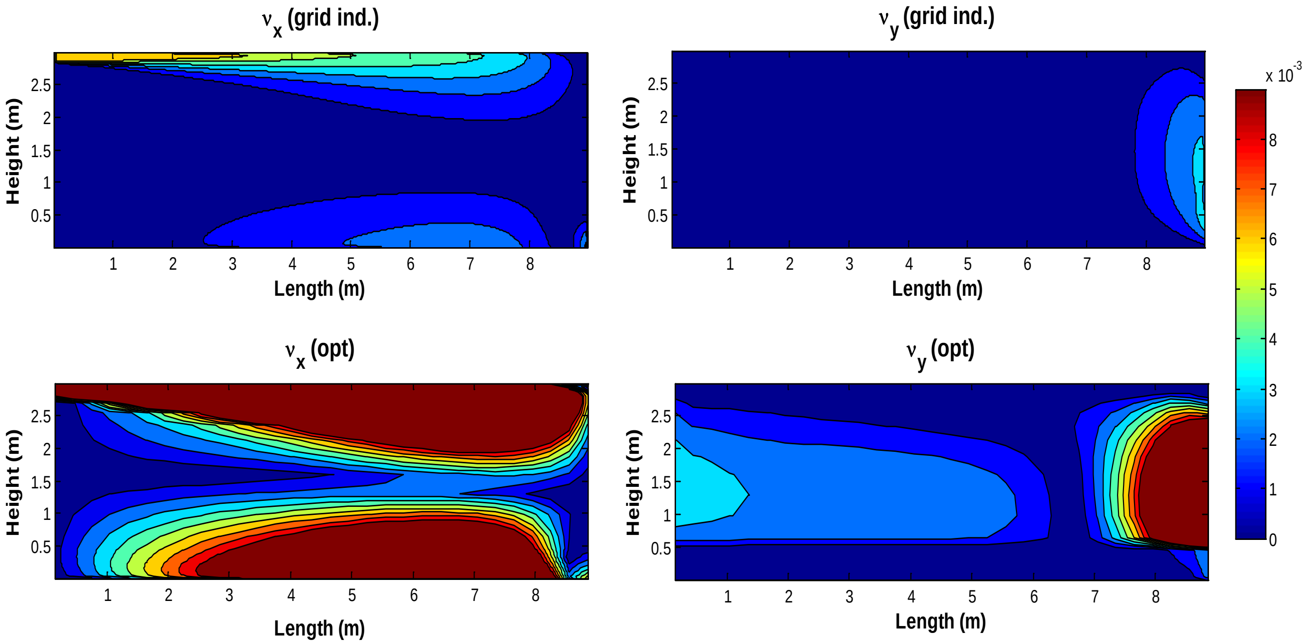


Figure 29. Numerical viscosity of grid independent (top) and optimized coarse (bottom) grid of FC Case

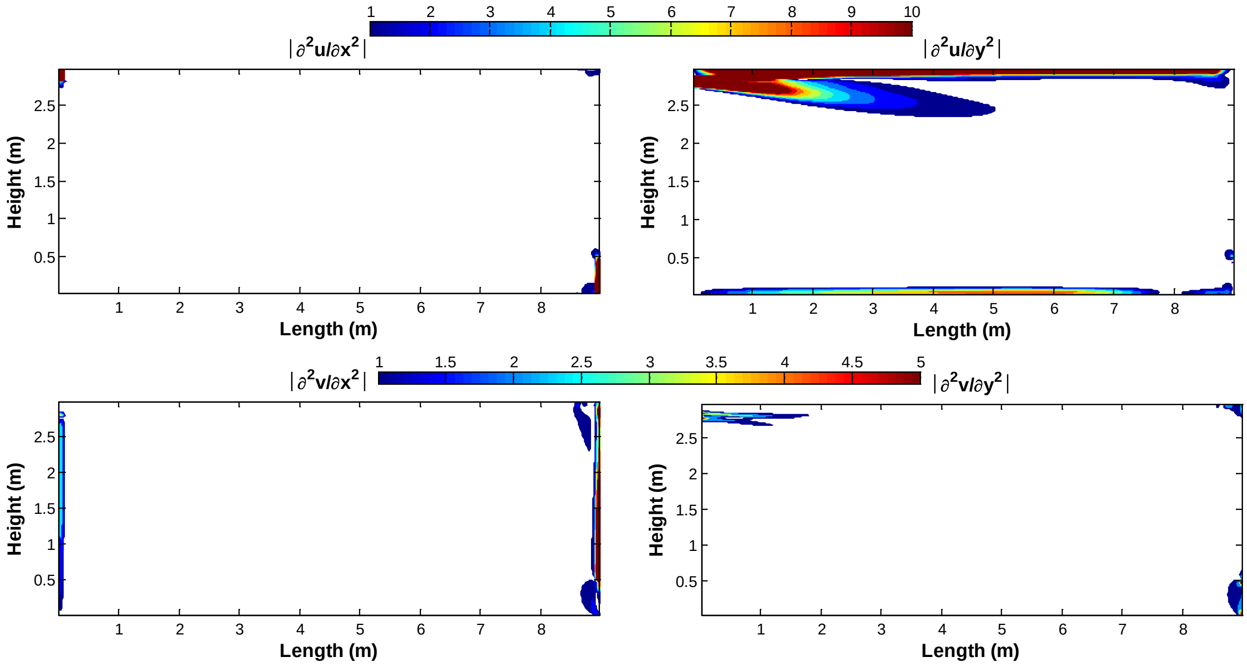


Figure 30. Calculated second order derivative terms for the FC case (U-top, V-bottom)

For the coarse grid, still ν_x near horizontal walls and ν_y near vertical walls are much greater than turbulent viscosity.

For the second order derivative term, as the contour plot in Figure 30:

Near top and bottom walls:

$$\frac{\partial^2 u}{\partial x^2} \ll \frac{\partial^2 u}{\partial y^2} \tag{4.17}$$

This gives a large flexibility to $\mathrm{v_x}$, or in other words, the size of X direction grid size. So the coarse grid can be applied for these regions, but on Y direction, the grid needs to be fine enough to eliminate the error.

And near two vertical walls:

$$\frac{\partial^2 u}{\partial x^2} \gg \frac{\partial^2 u}{\partial y^2} \tag{4.18}$$

This gives a large flexibility to ν_y, or in other words, the size of Y direction grid size. So the coarse grid can be applied for these regions, but on X direction, the grid needs to be fine enough to eliminate the error.

This gives the whole truncation error the term of grid independent and coarse grid solution close to each other, as in Figure 31.

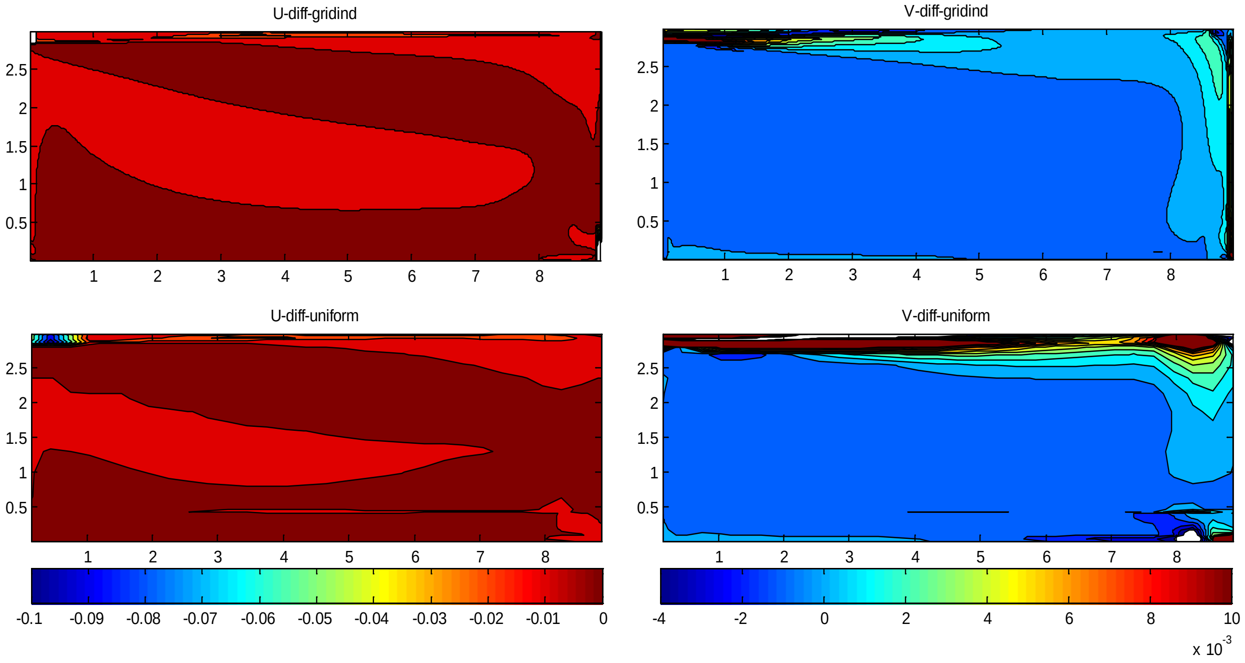


Figure 31. Whole diffusion value of grid independent (top) and optimized coarse (bottom) grid of FC Case

4.3.3 Mixed Convection Validation

The same mixed convection case has been used in this validation. The QUICK scheme is used for this case. Eq. (4.19) and (4.20) are used for normalization of measured variables (temperature and velocity).

$$T = \frac{t - t_w}{t_f - t_w} \tag{4.19}$$

$$U = \frac{u}{u_{inlet}};\ V = \frac{v}{u_{inlet}} \tag{4.20}$$

where t_f and t_w are the floor and wall temperatures and t is the predicted temperature on each point in the domain. Velocity can also be normalized by the inlet velocity $u = 0.57 m/s$, but this will not influence the RMSE result.

Table 11. Grid resolutions tested against measurements for MC Case

Grid index	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**
Grid # (XY)	4x5	6x7	9x9	13x13	18x18	26x26	36x36	52x52
Grid index	**#9**	**#10**	**#11**	**#12**				
Grid # (XY)	74x74	104x104	147x147	208x208				

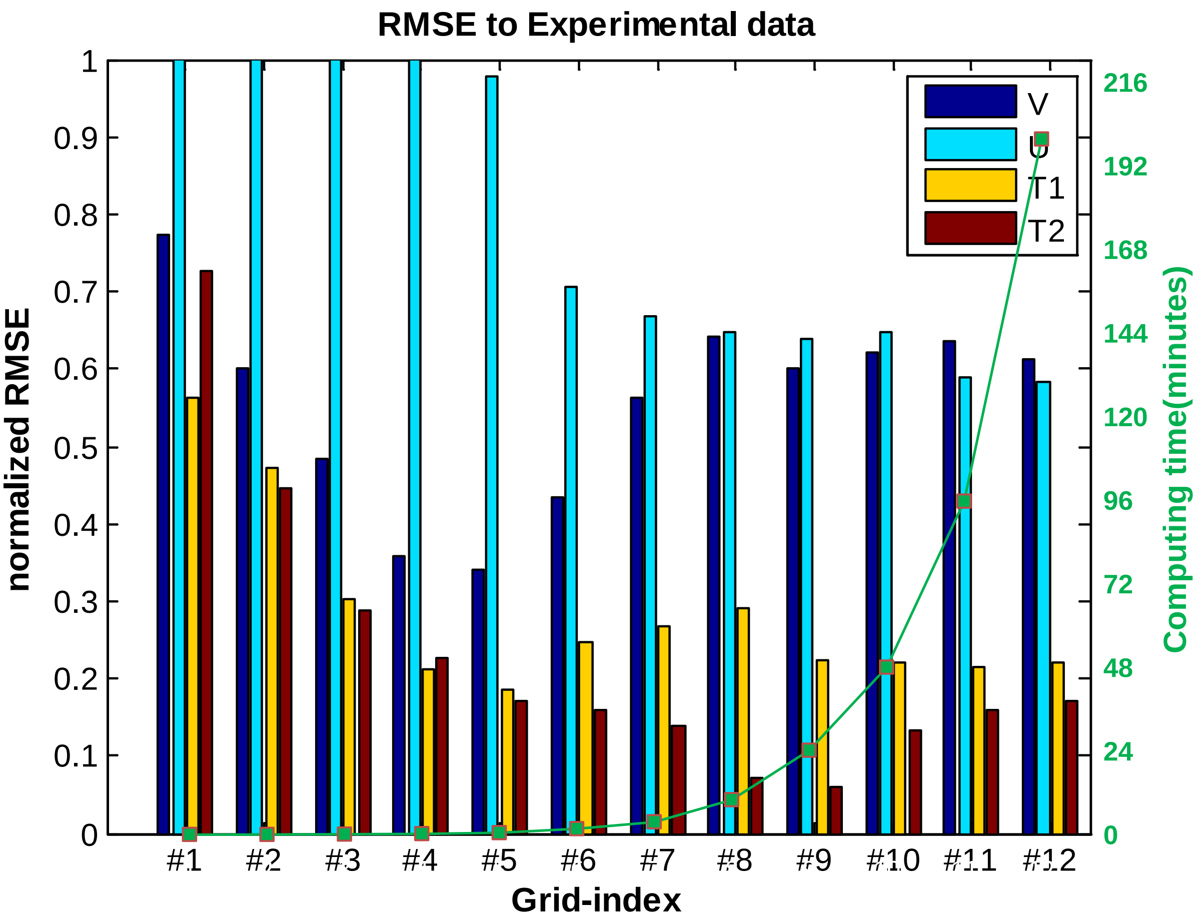


Figure 32. Performance of different uniform grid resolutions for MC Case

Table 11 and Figure 32 show the uniformly distributed grid resolutions and their performance by using the normalized RMSE value to compare experimental data. The computing time of each grid resolution is also included in Figure 32.

For this case, envelop serves as a heat source and sink due to the temperature difference, as observed in previous NC and FC cases, local refinemt with grid size around 1% of the geometry size was applied. For the surrounding inlet and outlet region, local refinement was also applied. Other than that, a coarse grid was used and transitional grid size was used in between to stabilize the computation. One can specify such transitional grid size or not, and the results will not be significantly affected. The optimized coarse grid was identified as in Figure 33.

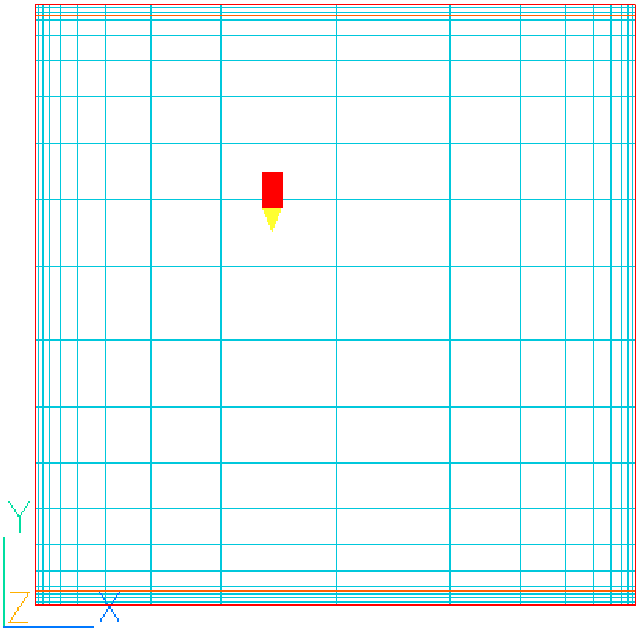


Figure 33. Grid distribution of grid independent and optimized coarse grid for MC case

Table 12 summarizes the performance of both the grid independent and optimized coarse grid. With only 0.7% of the grid independent computing cost, the optimized coarse grid gives a slightly worse prediction than the grid independent one, except T1, in which the optimized coarse grid performs better.

Table 12. Grid resolutions and normalized RMSE results for MC Case

Grid index	Grid number(X*Y)	Computing cost	Normalized RMSE compare to experimental data			
			V	U	T1	T2

Grid independent	147×147	t = 100%	0.1540	0.1712	0.1909	0.1526
Optimized coarse	15×22	t = 0.7%	0.1968	0.2267	0.0693	0.2192

The plot in Figure 34 shows the profile comparison against experimental data. It is noted that along mid-height, near the left wall, the grid independent solution over-predicts the temperature, whereas the optimized grid has a better prediction.

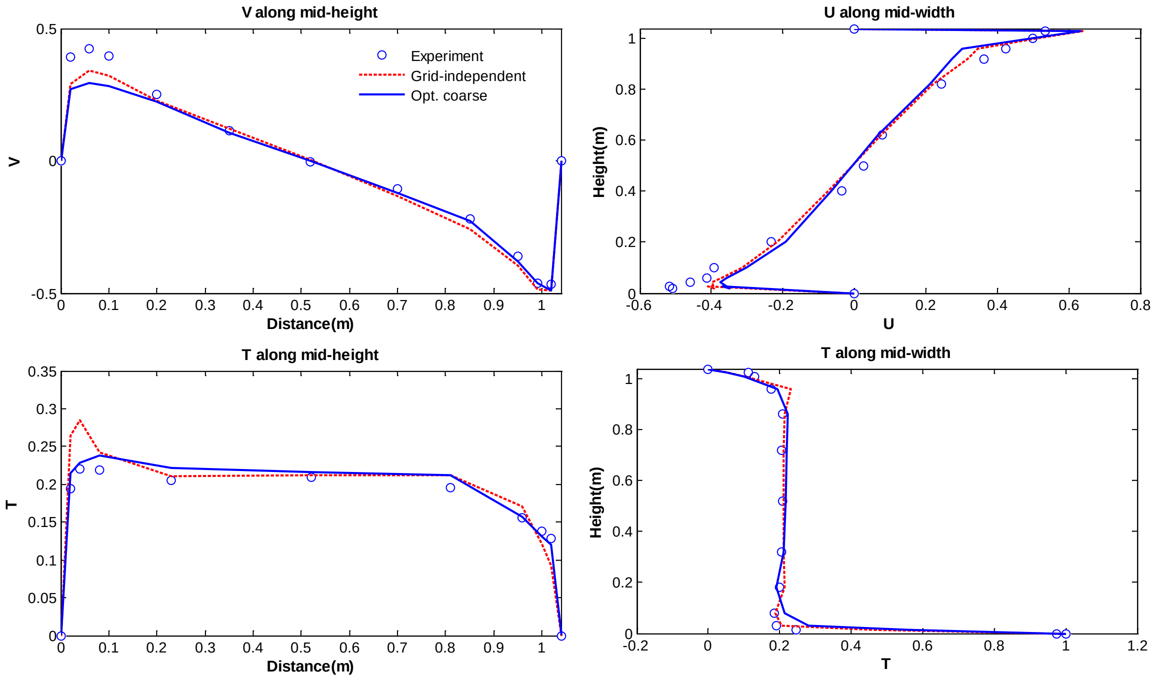


Figure 34. Profile comparison of different grids against experimental data for MC Case

Figure 35 gives the turbulence viscosity from the grid independent result as a standard for comparison.

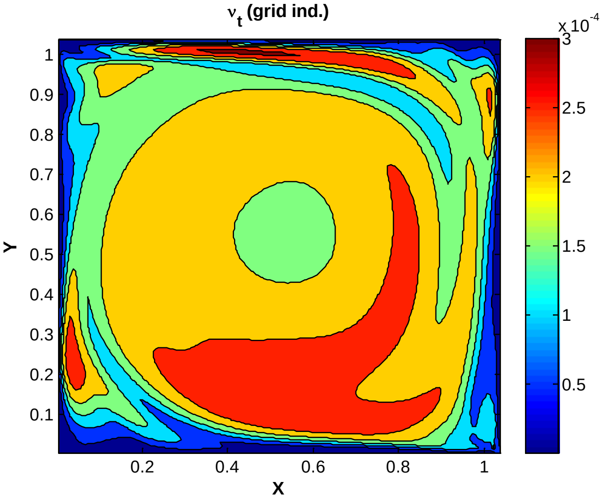


Figure 35. Turbulent viscosity of MC Case

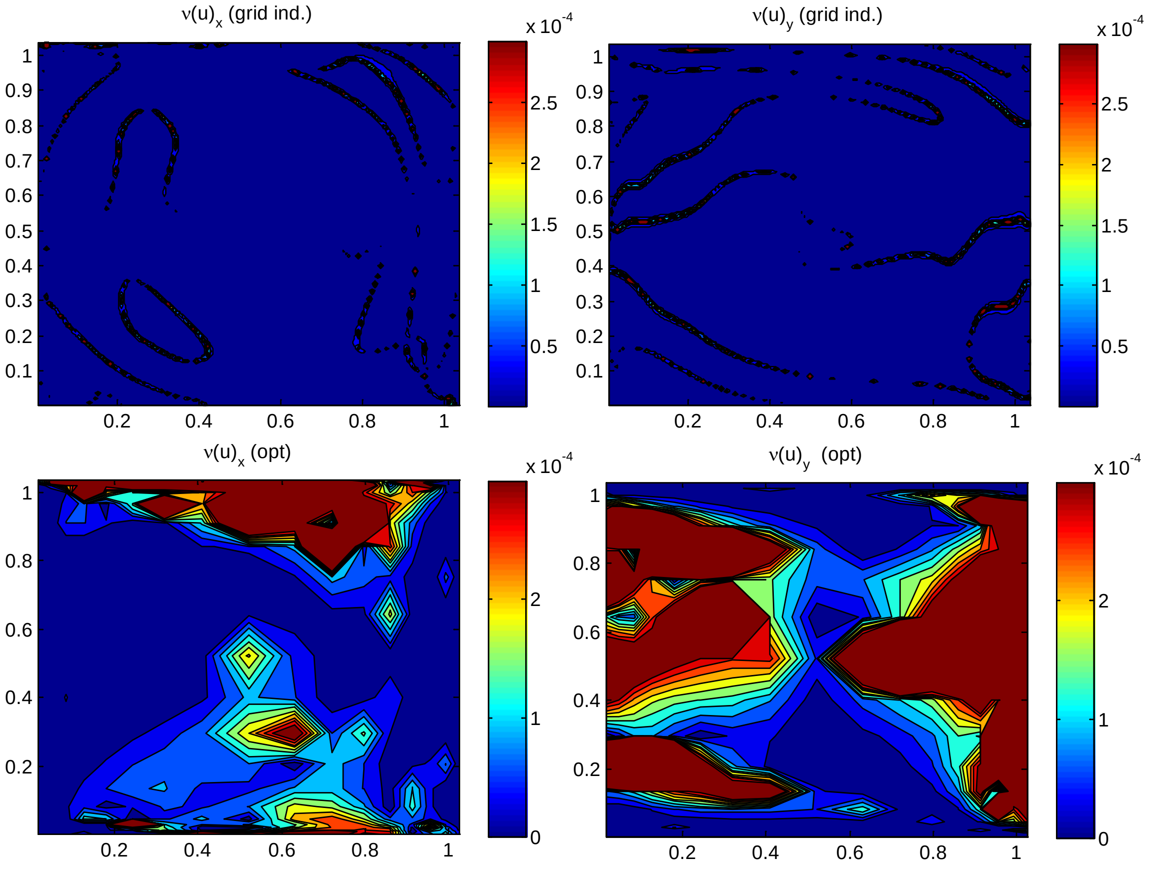


Figure 36. Numerical viscosity on U equation of grid independent (top) and optimized coarse (bottom) grid of MC Case

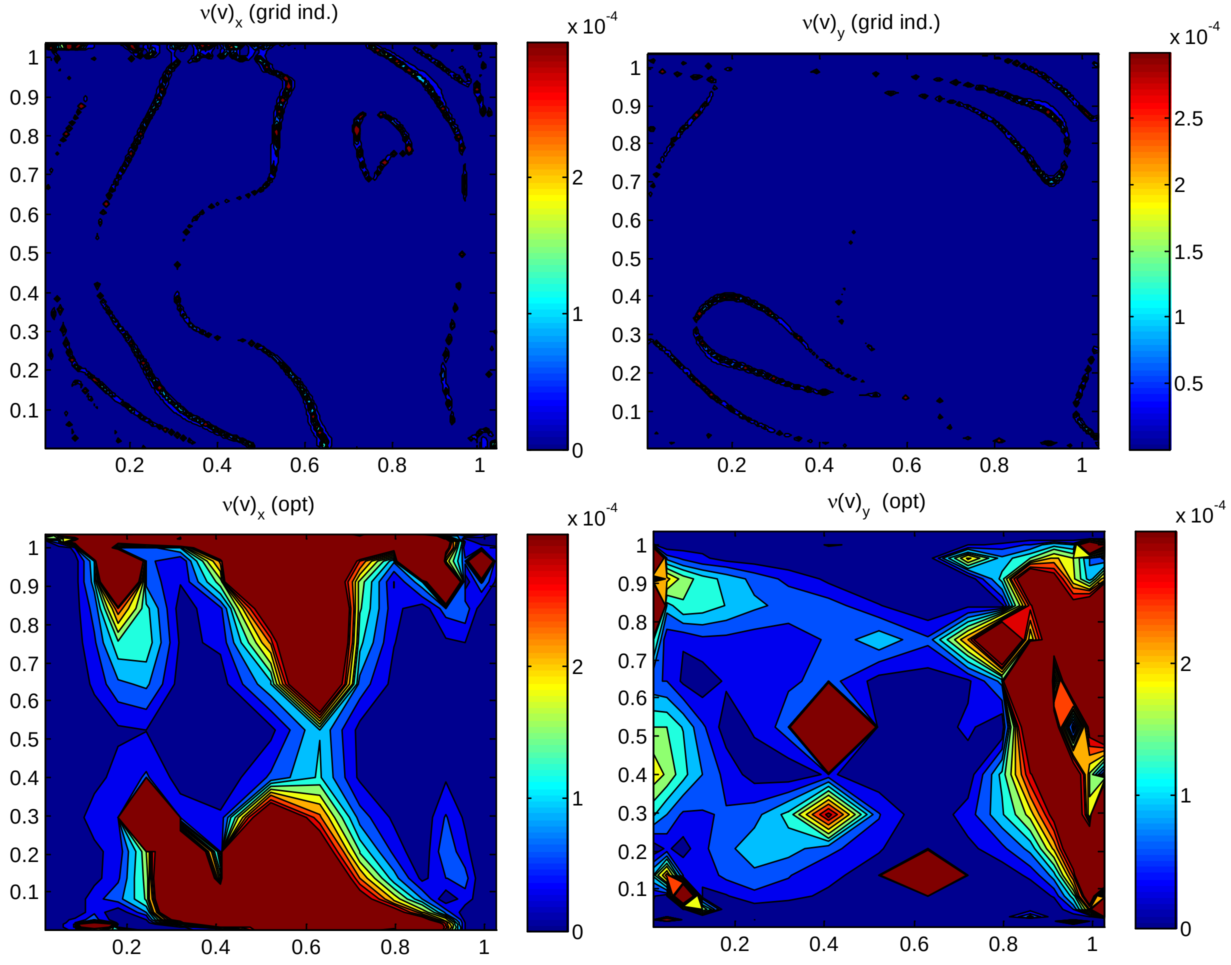


Figure 37. Numerical viscosity on V equation of grid independent (top) and optimized coarse (bottom) grid of MC Case

The numerical viscosity for QUICK scheme is different for U equation and V equation. Figure 36 and Figure 37 are the numerical viscosity on U and V equations respectively for both the grid independent and optimized coarse grids. The top is for the grid independent solution and the bottom is for the coarse grid solution. Note that they are put in the same range as turbulent viscosity contours. This verifies that the grid independent has much smaller numerical viscosity; the coarse grid has comparable or even much greater value than turbulent viscosity, which has the potential of causing great error.

Similarly, Figure 38 is the second order derivative term plot showing the regions for potential coarse grid application.

Near top and bottom walls:

$$\frac{\partial^2 u}{\partial x^2} \ll \frac{\partial^2 u}{\partial y^2} \tag{4.21}$$

This gives a large flexibility to ν_x, or in other words, the size of X direction grid size. So the coarse grid can be applied for these regions, but on Y direction, the grid needs to be fine enough to eliminate the error.

And near the right vertical wall:

$$\frac{\partial^2 u}{\partial x^2} \gg \frac{\partial^2 u}{\partial y^2} \tag{4.22}$$

This gives a large flexibility to ν_y, or in other words, the size of Y direction grid size. So the coarse grid can be applied for these regions, but on X direction, the grid needs to be fine enough to eliminate the error.

This does not exactly correspond the numerical viscosity of the coarse grid, but there is an approximate match.

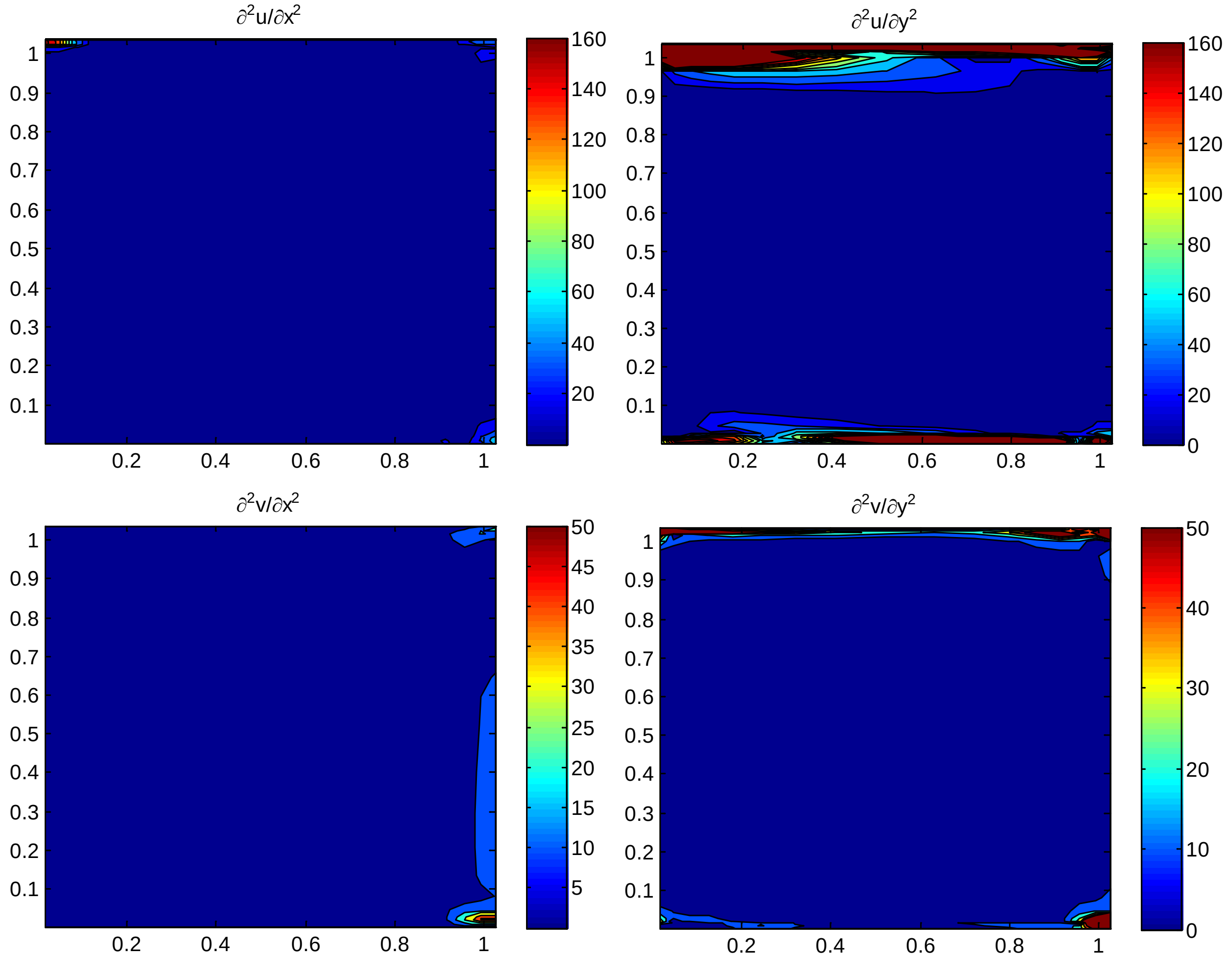


Figure 38. Second order derivative term of MC Case (U-top, V-bottom)

For the Quick scheme, numerical viscosity is used for analysis about grid independency; and since the truncation error is on the third derivative term, the discretization-induced error is also investigated on the coefficient of this third derivative term.

The Quick scheme is different because the truncation error is on the third derivative term as highlighted in the following momentum equation:

$$u\frac{\partial u_{i,j}}{\partial x}+v\frac{\partial u_{i,j}}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+\nu\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]-u\frac{\Delta x^2}{24}\frac{\partial^3 u_{i,j}}{\partial x^3}-v\frac{\Delta y^2}{24}\frac{\partial^3 u_{i,j}}{\partial y^3}+S_x+O(\Delta x^3)+O(\Delta y^3) \tag{4.23}$$

Figure 39 is the coefficient of the third order derivative term. For coarse grid, this coefficient is much greater, especially near horizontal walls along the X direction, and near vertical walls along the Y direction.

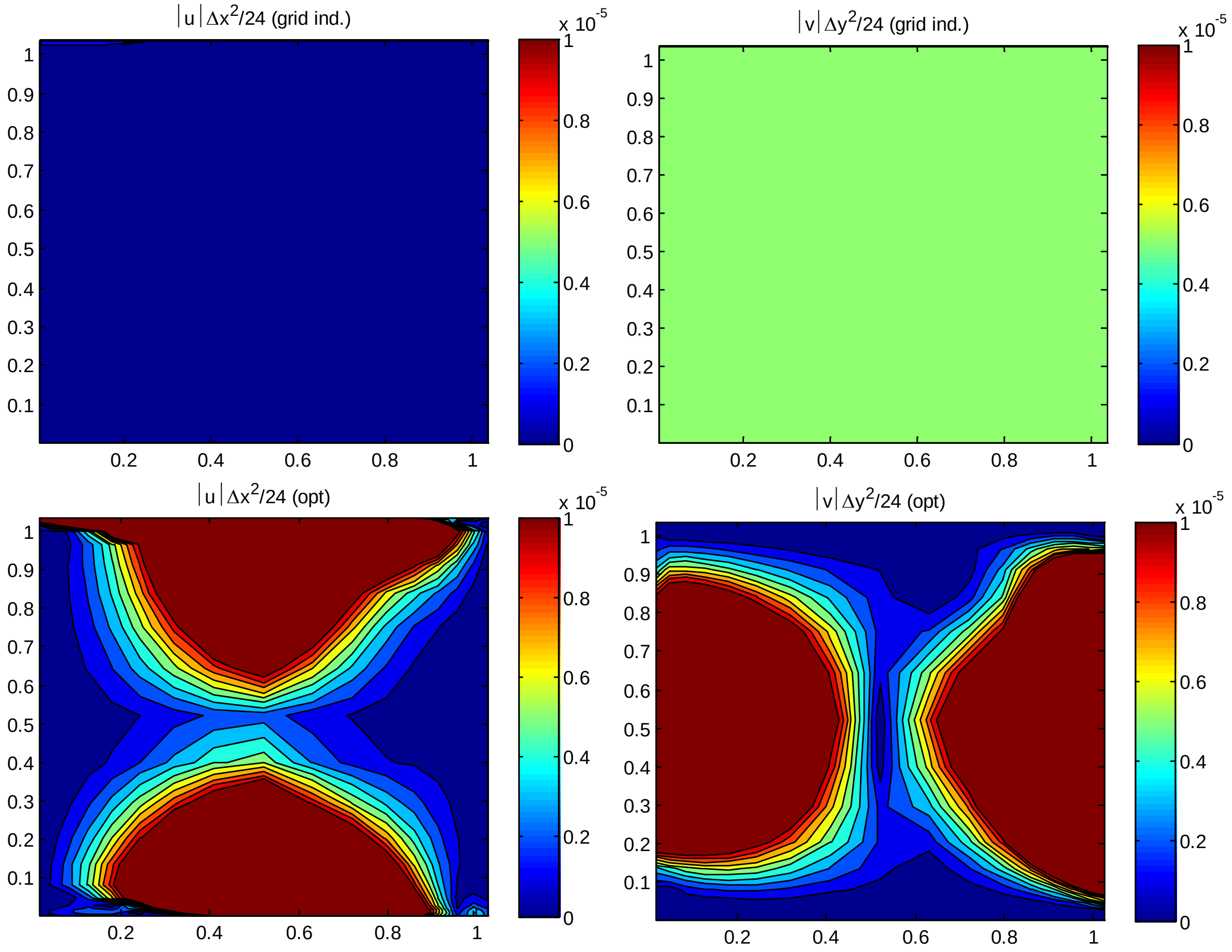


Figure 39. Coefficient from discretization on third order derivative term of MC Case (Grid independent-top, Optimized coarse grid-bottom)

As for the third order derivative term in Figure 40, the gradient over x is greater near vertical walls, and the gradient over y is greater over y. This explains why near vertical walls fine grid on the x direction is required near vertical walls (a coarse grid can be applied on y direction); near horizontal walls, a fine grid on y direction is required near vertical walls (a coarse grid can be applied on x direction).

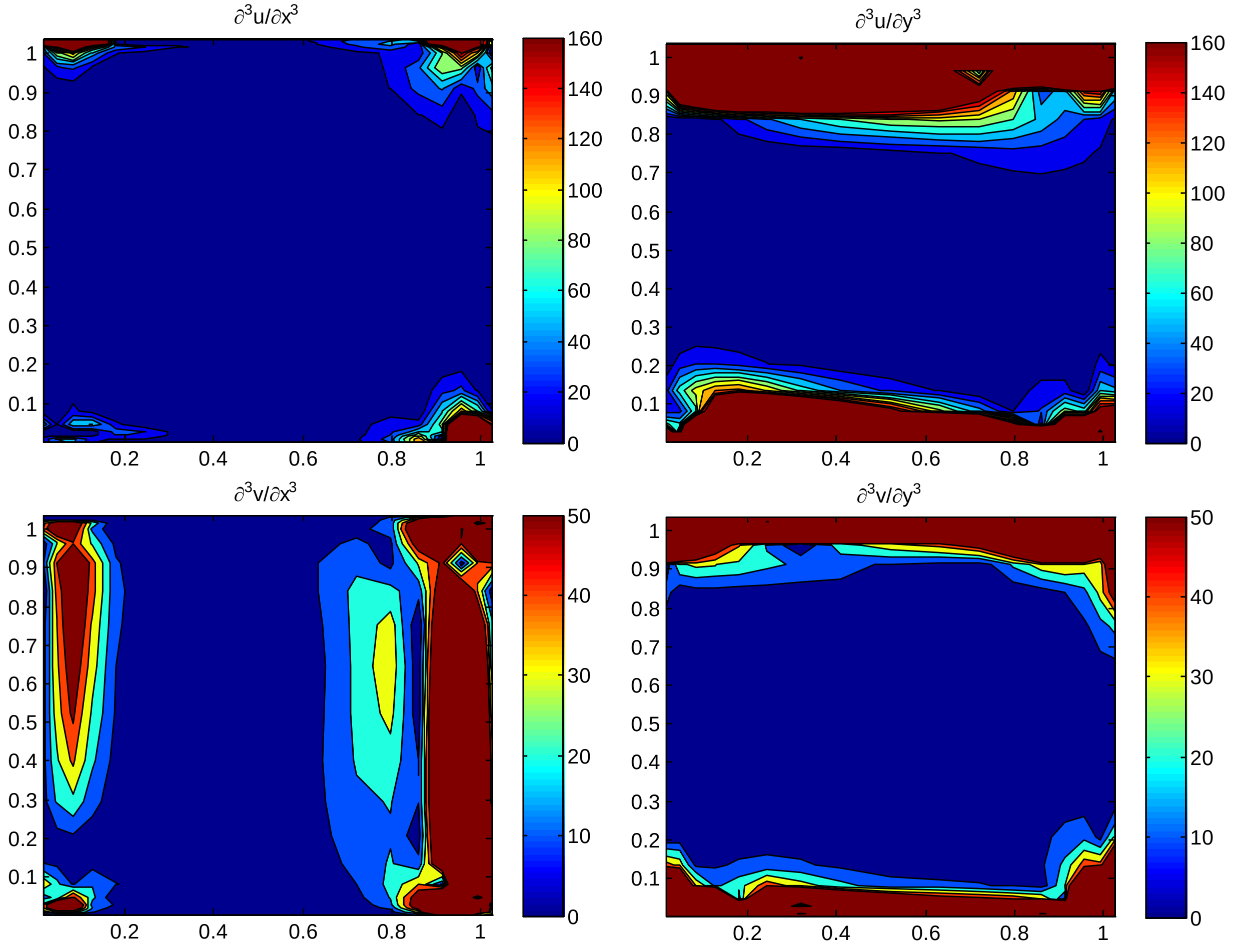


Figure 40. Third order derivative term of MC Case (U-top, V-bottom)

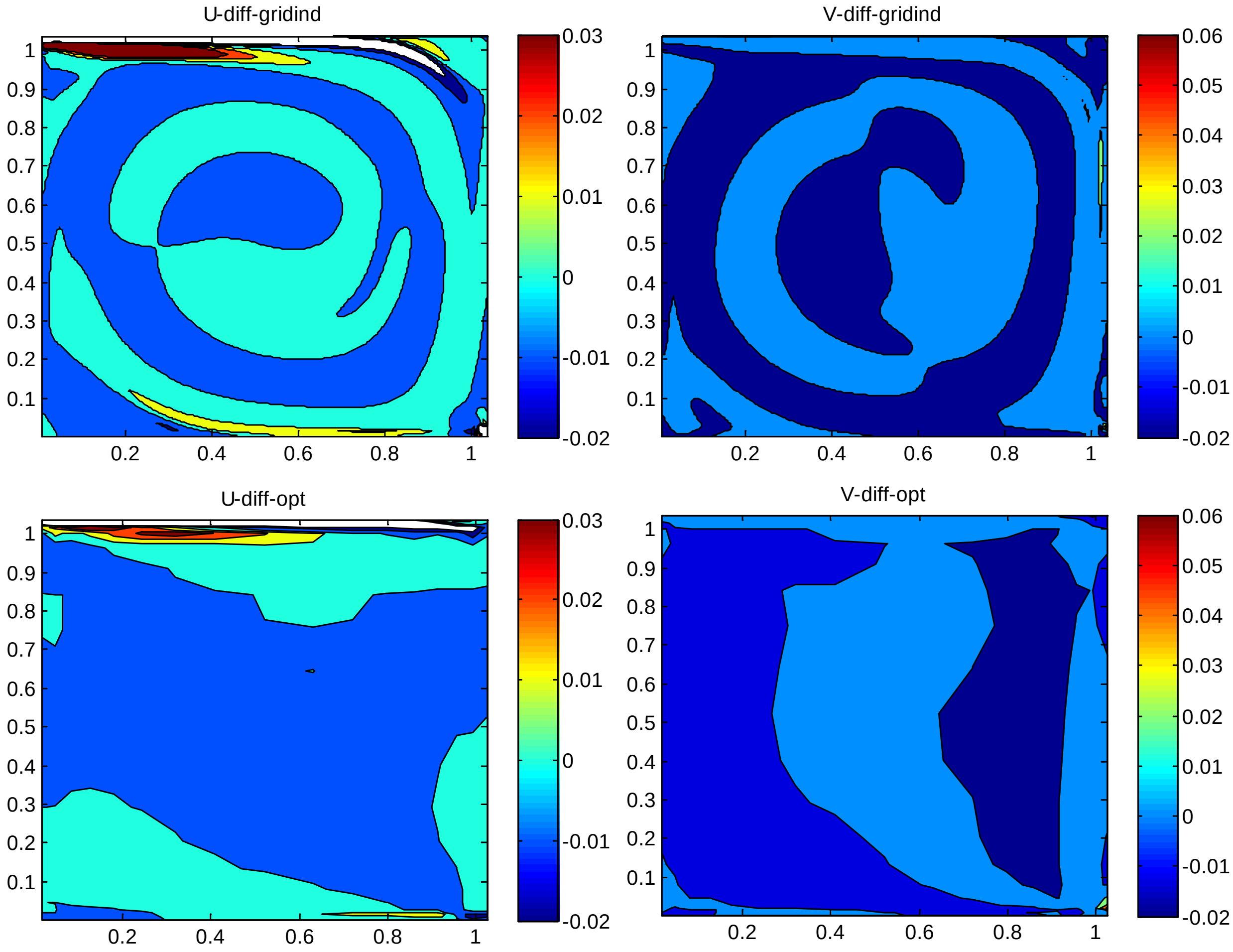


Figure 41. Whole diffusion value of grid independent (top) and optimized coarse (bottom) grid of MC Case

The total effect of this can be seen in Figure 41, which shows the whole truncation error term plot for both grid independent and optimized coarse grids. They are in good agreement with each other.

The analysis from both aspects is for the grid induced error term. It can be seen that the whole error term can be minimized by using a well-designed grid properly distributed in the domain. For different orders of numerical schemes, the trend for grid refinement is almost the same. The effect of the numerical scheme on such analysis is not quite important; the first order scheme such as hybrid has similar viscosity and gradient term in a domain.

Through a series of 2D test cases, it is found that local refinement is usually necessary near regions where large gradient occurs, such as normal direction of heat source and sink, and normal direction of flow inlet and outlet. The local refined grid size is around 1% of the geometry size of the computational domain, whereas the coarse grid size is around 10% of such size. Next three dimensional case will be used to test the conclusion of such observations.

4.3.4 Three Dimensional Case Validation

4.3.4.1 Natural Ventilation

A buoyancy driven natural ventilation (NV) room [108] was identified for demonstration. The detailed configuration of this experiment is in Figure 42. The test chamber adjacent to the environmental chamber has a heater inside. The chamber system is put inside a larger room. The chambers are well insulated and the wall is supposed to be adiabatic. Air velocity and temperature were measured along five vertical poles marked as P1-P5. In the CFD model in this research, only the test chamber is modeled, so the data for comparison in this research is on P2 to P5.

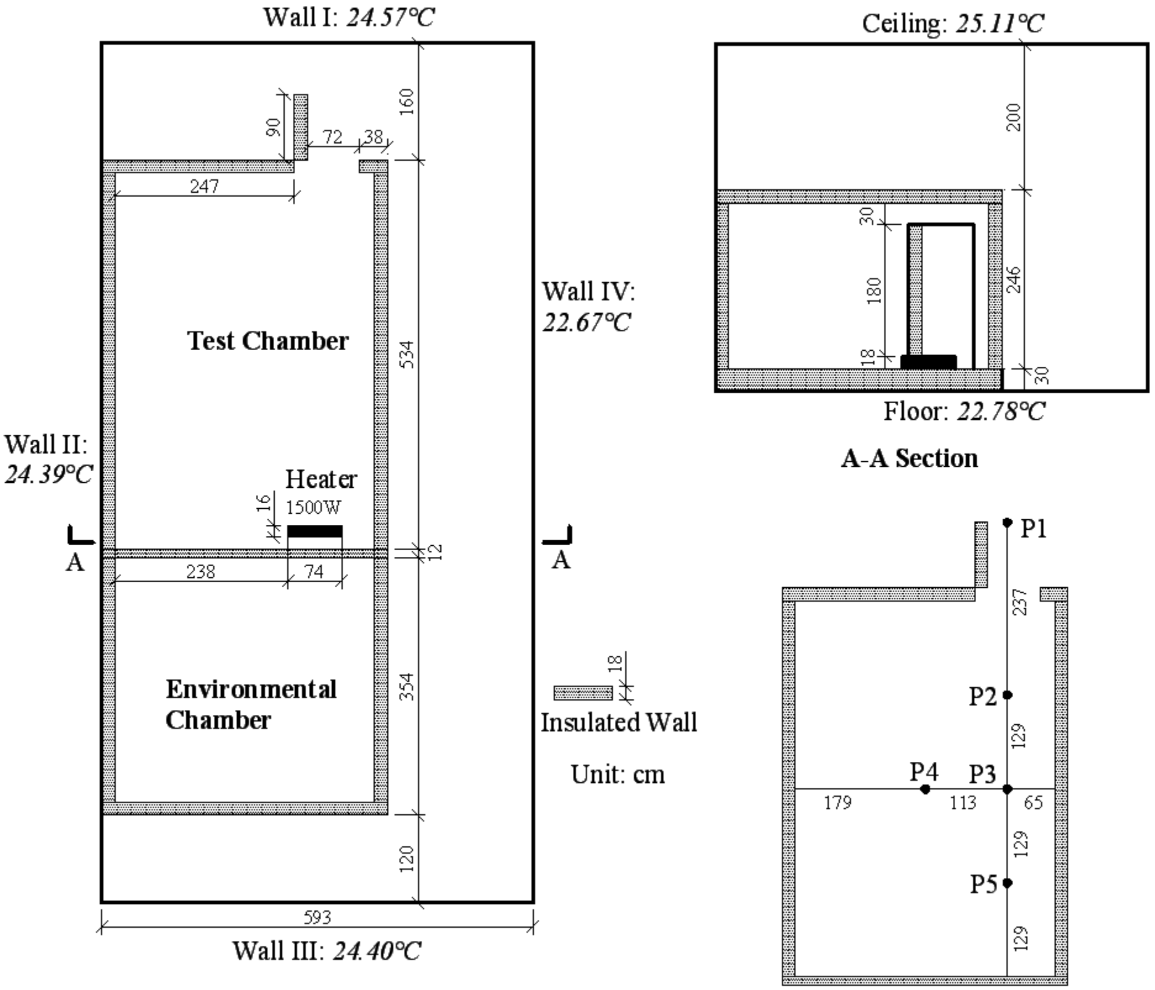


Figure 42. Configuration of buoyancy-driven natural ventilation case

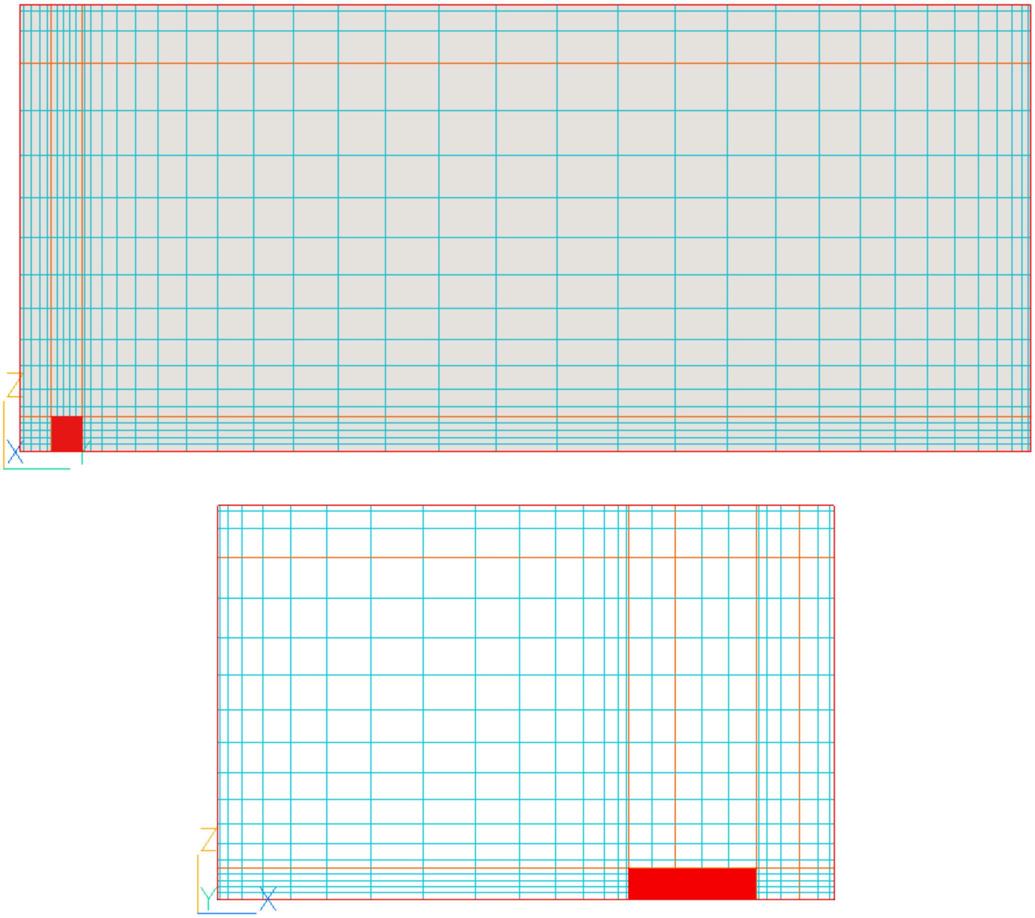


Figure 43. Grid distribution of optimized coarse grid for buoyancy-driven natural ventilation case

According to the general rule of coarse grid specification, local refinement applies to heat source/sink for this buoyancy-driven natural ventilation. Figure 43 shows the grid distribution of optimized coarse grid for this case from the view of two vertical intersections. Because of radiation, all inner surfaces act as heat sources, and local refinement normal to each surface thus applies.

The measured and predicted temperatures are normalized by exhaust air temperature and surrounding (environment) air temperature as

$$T = \frac{t - 25}{33 - 25} \tag{4.24}$$

Table 13. Grid resolutions and normalized RMSE results compared to experimental data for natural ventilation Case

Grid index	Grid number(X*Y*Z)	Computing cost	Normalized RMSE to experimental data	
			V	T
Grid independent	80*78*50	t = 100%	0.3989	0.1400
Optimized coarse	33*40*19	t = 5.0%	0.4033	0.2043

Table 13 summarizes the grid resolution and performance of two grid resolutions in terms of computing cost and normalized RMSE value comparing to experimental data. With about 5% of the original computing cost, the coarse grid gives a comparable prediction to the grid independent one. Figure 44 further confirms the closeness of the prediction of fine and coarse grids.

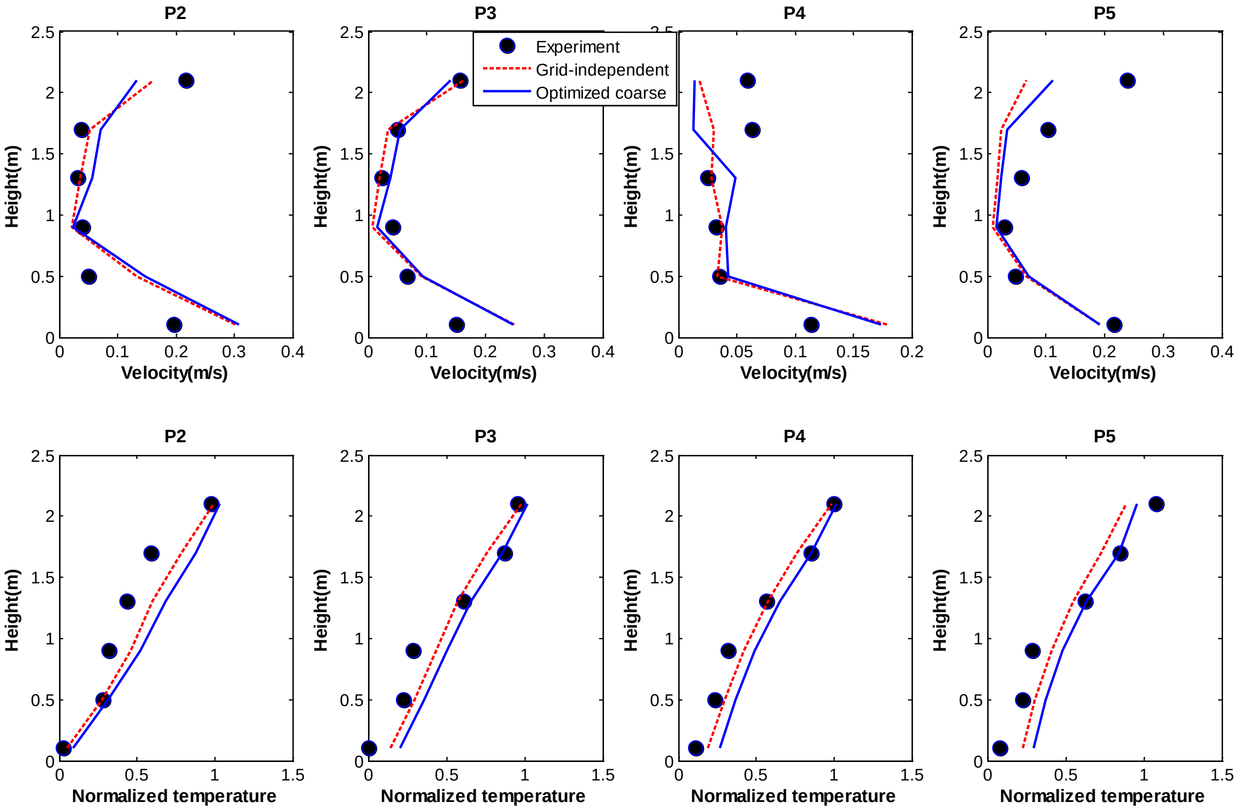
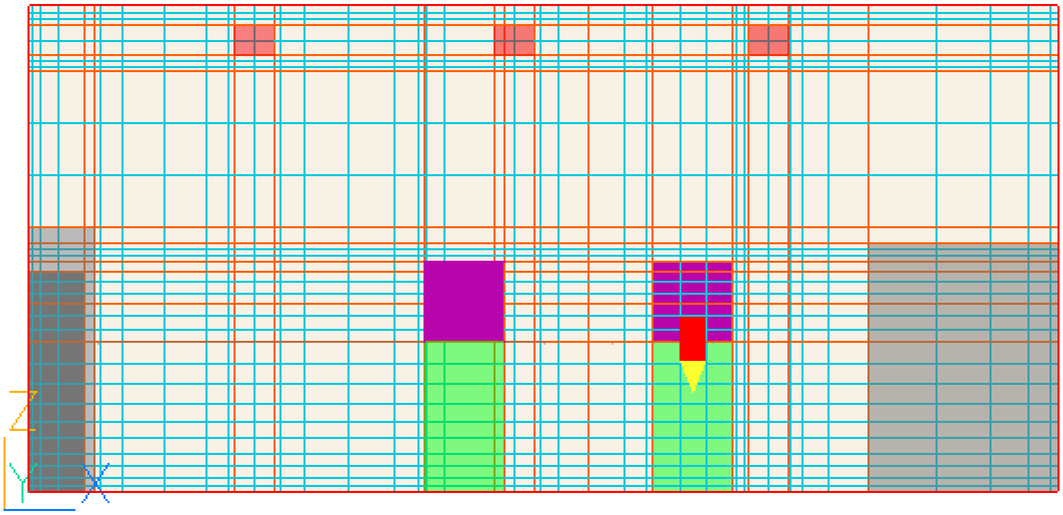


Figure 44. Profile comparison of solutions of different grids against experimental data for NV case

4.3.4.2 Displacement Ventilation

A side-wall supply displacement ventilation (DV) case [107] as in Figure 15 is also used for verification and demonstration. The experimental chamber serves as a displacement ventilation room. The diffuser inlet on the wall provided a ventilation rate of 4ACH for the chamber, indicating the supply air flow rate Q=183.1m^3/h. The supply air temperature is t=17.0℃. The effective area ratio is 10%, so the supply air velocity of the diffuser is equal to v=0.864m/s. With such information, a CFD model can be built. Temperature and velocity were measured along nine vertical poles uniformly distributed within streamwise central intersection (P1-P5) and cross section central intersection (P6-P9) as in Figure 16.

The coarse grid is complex for this case due to the number and positions of objects. For the main heat source surfaces such as lamp, computer and person, local refinement with about 1% the height of the room is applied. The same local refinement applies to the region surrounding the inlet diffuser and exhaust. The detailed distribution of the coarse grid is as Figure 45.



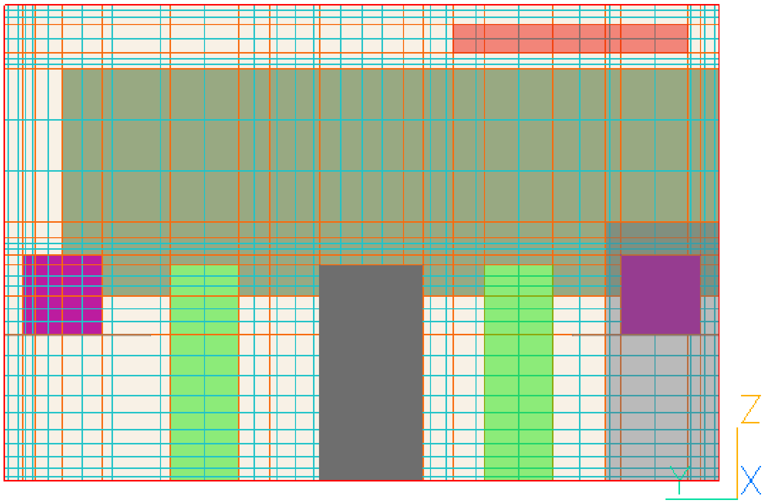


Figure 45. Grid distribution of optimized coarse grid for DV case

The measured and predicted data are normalized for further comparison. Temperature is normalized by inlet and outlet temperature as Eq. (4.24); while velocity is normalized by equivalent inlet velocity as Eq. (4.25).

$$T = \frac{t - 26.7}{26.7 - 17.0} \tag{4.25}$$

$$V = \frac{vel.}{u_{inlet}} = \frac{vel.}{0.086} \tag{4.26}$$

Table 14. Grid resolutions and normalized RMSE results for DV Case

Grid index	Grid number (X*Y*Z)	Computing cost	Normalized RMSE compare to experimental data			
			V-streamwise	V-cross section	T-streamwise	T-cross section
Grid independent	123×86×54	t = 100%	0.7320	0.4301	0.0259	0.0288
Optimized coarse	48×44×32	t = 5.9%	0.7777	0.5140	0.0207	0.0291

Table 14 summarized the performance of the two grid resolutions in terms of computing cost and normalized RMSE. The optimized coarse grid consumes 5.9% of the uniform grid independent computing cost based on the same computer platform. The prediction results are close to each other. The profile comparison against experimental data is as Figure 46, which only shows the results along vertical poles 1-5 for the space limitation of this paper. The temperature predictions are almost identical except some certain points.

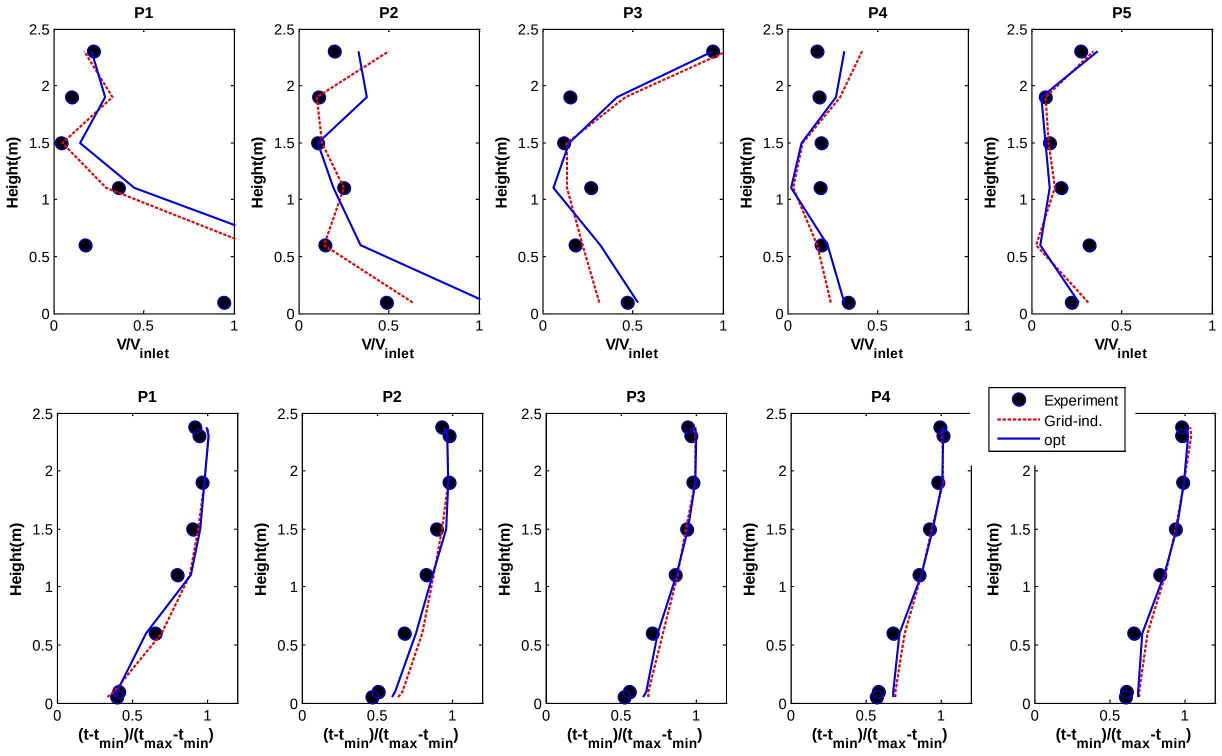


Figure 46. Profile comparison of solution of different grids against experimental data for DV case

4.3.5 Conclusive Summary

The results presented in this section confirm the hypothesis that grid-induced error, in other words artificial diffusion, is not only determined by the magnitude of numerical viscosity but also the corresponding derivatives that multiply the numerical viscosity. This provides the opportunity to use the coarse grid with optimized distribution in CFD. Practically, CFD users can specify a coarse grid to get the general flow pattern of a model, comparing the orders of magnitude of diffusion and determining the fine and coarse grid areas in the domain. Usually a large gradient exists in the normal direction of flow in/out and heat source/sink. As recommended as a rule of thumb[109], assume the characteristic length of a geometry under investigation is L; a grid size of 1%L is recommended for these areas, and for other areas and directions, a 10%L is recommended. The goal of using a coarse grid is mainly to reduce the computing cost. The geometry height is usually a good representative of characteristic length as indicated by different test cases. With at most around 5% of the original computing cost, the optimized coarse grid according to these guidelines can have comparable numerical results as grid independent solutions.

4.4 Conclusions

The normalized RMSE value provides an alternative for using a sing index to show the overall closeness between CFD solutions as well as of CFD results to experimental data. Normalized RMSE provides a metric to compare the difference between CFD predictions for grid independency study purposes. Together with numerical viscosity analysis, an objective alternative to analyze CFD grid independency is developed. Normalized RMSE can be also used to compare the difference between CFD solutions and experimental measurements to evaluate the performance of a CFD model.

Magnitude analysis on numerical viscosity and artificial diffusion show great opportunity for using a coarse grid CFD to obtain comparable results as grid independent results in typical indoor environment modeling. By optimizing the distribution of the coarse grid, the total truncation error introduced by space discretization can be minimized. This is verified practically through applying the hypothesis to different indoor airflow mechanisms with different complexities of configuration.

CHAPTER 5
TURBULENCE MODEL SIMPLIFICATION

As discussed in the literature review, turbulence modeling is another important factor that determines the computing intensity of a CFD model. Especially in indoor environment study, turbulence parameters are not critical for the environmental quality. It is usually computed in CFD models to determine the turbulence viscosity value used as a coefficient in the momentum equation. Turbulence modeling can be simplified to exclude the equations of turbulence models from the equation group, providing that we know or have an easier way to determine the turbulence viscosity value. This chapter will discuss the simplification of turbulence models by taking advantage of grid induced numerical viscosity.

5.1 Numerical Viscosity Analysis

For turbulence modeling solving the Reynolds-Averaged Navier-Stokes (RANS) Equations [38], according to the Boussinesq approximation [39], the turbulence transfer mechanism can be modeled as an eddy viscosity or turbulent viscosity, which involved extra equations in the CFD model. This simply indicates that turbulent flow behaves as a more viscous fluid compared to laminar flow, and the viscosity depends on the turbulent property. Take turbulent viscosity into consideration, as a recall of Eq. (4.1), the 2D steady-state momentum equation solved in CFD is

$$\bar{U}\frac{\partial \bar{U}}{\partial x}+\bar{V}\frac{\partial \bar{U}}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+\left[\nu_{physical}+\nu_t+\nu_x\right]\frac{\partial^2 \bar{U}}{\partial x^2}+\left[\nu_{physical}+\nu_t+\nu_y\right]\frac{\partial^2 \bar{U}}{\partial y^2}+S \tag{5.1}$$

where $\nu_{physical}$ is the physical viscosity written as ν previousely; $\bar{U}$ and $\bar{V}$ denotes the Reynolds averaged velocity component along X and Y directions. For typical indoor environment airflow, physical viscosity is usually one to two orders of magnitude smaller than turbulent viscosity ν_t, so it is therefore neglected in the further analysis.

As numerical viscosity is proportional to grid size, it is possible to specify a certain grid size such that $\nu_x \cong \nu_t$ and $\nu_y \cong \nu_t$; therefore, the effect of numerical viscosity (i.e., grid size) may substitute that of turbulent viscosity from turbulence models. This specification usually ends up with a coarse grid in the real CFD practice for typical indoor airflow simulations; therefore, there is great potential of reducing computational cost due to the utilization of coarse grids and the removal of PDE-based turbulence models.

5.2 Constant Viscosity Model

5.2.1 General Rule Test

According to the theoretical analysis, coarse grid induced numerical viscosity may be of benefit in indoor airflow modeling, provided that the numerical viscosity provides similar contributions to fluid flows as the turbulence viscosity. To verify this hypothesis, numerical experiments on a benchmark case with detailed experimental data are conducted. The Nielsen's experiment [105] used in previous chapters, which is a two-dimensional (2D) case, is employed in this study. As a recall, Figure 47 shows the configuration of the experiment. It is an isothermal case. The horizontal velocity component data along four measurement lines is available and used in the result analysis. The relatively simple configuration of this case excludes uncertainties in boundary conditions and heat transfer from the model and makes it easier to focus on the parameters under investigation.

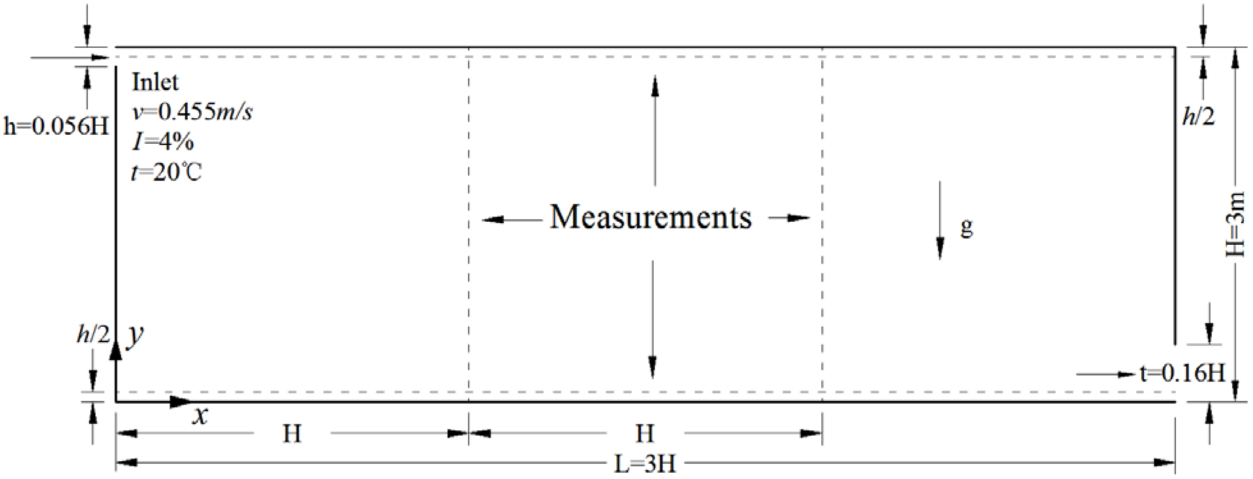


Figure 47. Configuration, boundary condition and measurements of the 2D forced convection case

The purpose of this validation is to use the coarse grid induced numerical viscosity to replace turbulent viscosity. However, such viscosities are not a real physical property of airflow. To determine the general pattern of turbulent viscosity in this case, the Renormalized Group (RNG) k-ε model [43] has been used to simulate this case. The distribution of predicted turbulent viscosity using a fine grid is demonstrated as Figure 48. The predicted turbulence viscosity combined with the simulated velocity values can be used to design and calculate the necessary grid size using Eq.(5.2). the designed grid size can produce the same numerical viscosity as the turbulence viscosity.

$$\begin{aligned} \nu_t &\cong \nu_x = \left|\bar{U}\frac{\Delta x}{2}\right| \\ \nu_t &\cong \nu_y = \left|\bar{U}\frac{\Delta y}{2}\right| \end{aligned} \tag{5.2}$$

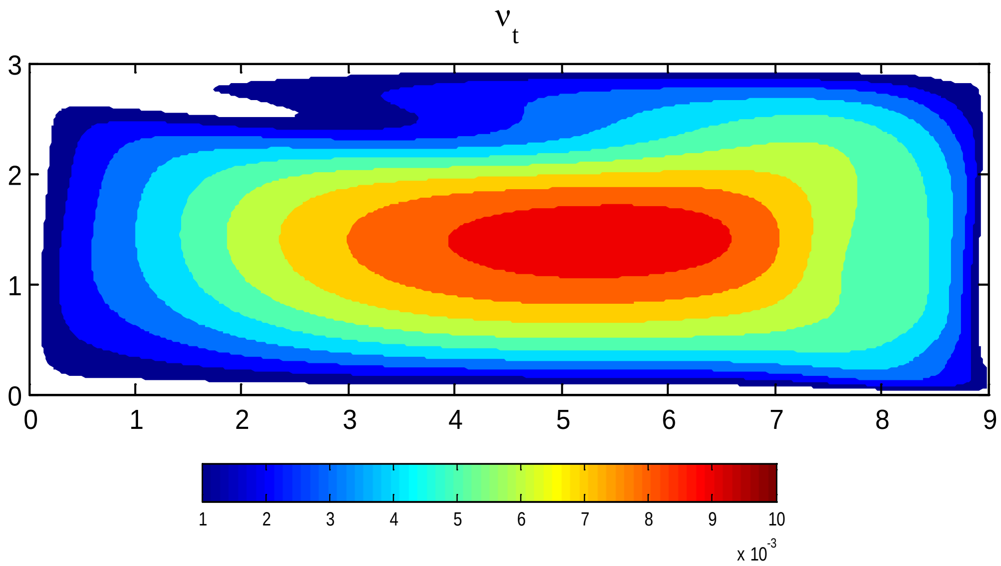


Figure 48. Turbulent viscosity of FC case obtained from a fine grid CFD with the RNG k-ε model

Assume a structured grid is used; the desired grid size is calculated and plotted in Figure 49. However, in a real simulation, it is difficult to generate a CFD grid as shown in this figure that generates exactly the same numerical viscosity as the turbulent one, especially with a structured grid. More importantly, the grid size distribution should also consider the need to provide sufficient resolution for large gradient areas: typically the grid size shall not be greater than 1/10 of the length of geometry in order to capture the flow trend details.

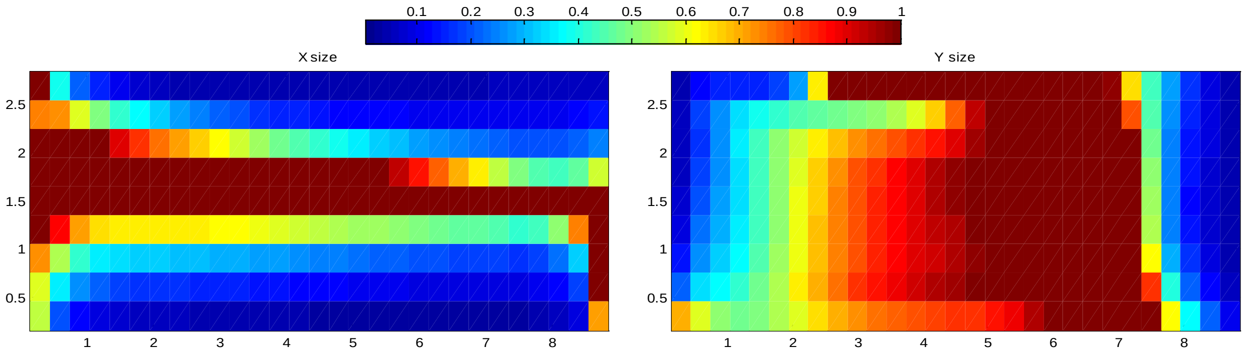


Figure 49. Grid size (m) requirement if using numerical viscosity to replace turbulent viscosity

On the other hand, the magnitude difference of the second order derivatives in Eq. (5.1) can be utilized to compromise the grid size. Figure 4 plots the second order derivatives of the velocity components. Comparing the top two contours, $(\partial^2 u)/(\partial x^2) \ll (\partial^2 u)/(\partial y^2)$ near the floor and ceiling of the domain $((\partial^2 u)/(\partial x^2\ ) < 1$ and $(\partial^2 u)/(\partial y^2) > 10$*),* the corresponding viscosity on each term, ν_x and ν_y, thus have different weighting impacts on the whole diffusion term. Near these regions, ν_x , or in other words, the grid size along the horizontal direction, can be compromised and will not significantly affect the solution of the equations. Generally, the "X size" in Figure 49 near the floor and ceiling is not necessarily kept as very small. Similarly in Figure 50, near the two vertical boundaries, $(\partial^2 v)/(\partial x^2) \gg (\partial^2 v)/(\partial y^2)$, hence "Y size" can be compromised with less influence to the simulation.

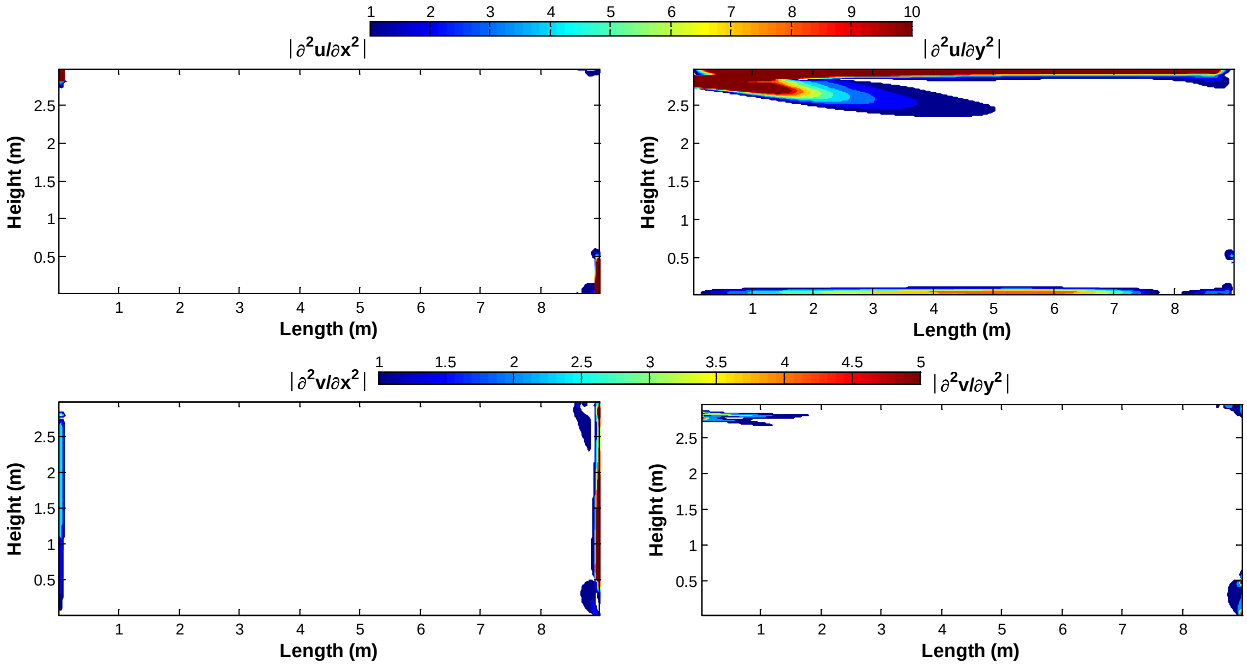


Figure 50. Calculated second order derivative terms for the 2D forced convection case

Taking all these analyses into consideration, a grid shown in Figure 51, similar to the uniform structured one (17×6), is generated and tested for the prediction performance. Since this grid resolution has already been compromised based on the previous analysis, the numerical viscosity induced is not large enough to substitute for the entire turbulence viscosity, especially at the central part of the domain. Some background turbulence viscosity values need to be added artificially. A constant turbulence viscosity model may serve this purpose well with good prediction accuracy [40]. The constant viscosity model will simply change the property of the air without adding additional PDE or algebraic equations.

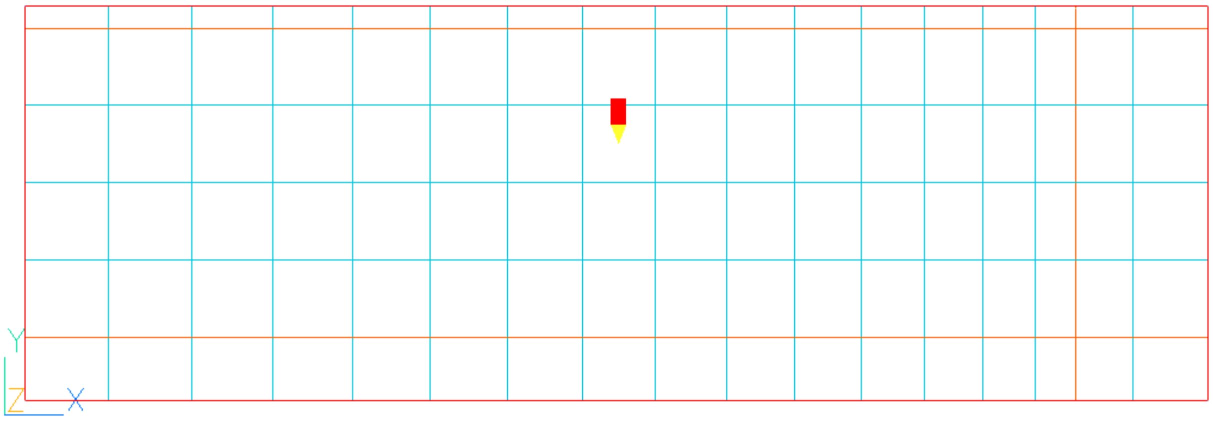


Figure 51. Designed coarse grid distribution (17×6) for the 2D forced convection case

Assume the physical kinematic viscosity of air is ν. The results of adding 10ν, 20ν, 50ν, and 100ν are demonstrated in Figure 52, simulated by the commercial CFD software PHOENICS [15]. The experimental data and the grid independent (300×100) results using the RNG k-ε model are also included for the comparison. Figure 53 applies a normalized RMSE to estimate the overall performance of each CFD model through the comparison against available experimental data. A smaller RMSE value indicates a closer simulation solution to the experimental data. It appears that the 20ν result has the best overall performance among those tested with the specific coarse grid. Disparities exist between the predicted and experimental results, even for the model with the computationally expensive turbulence model and the fine grid. The coarse grid predictions, even though not as accurate as the grid independent result, simulate the airflow characteristics in the domain with an acceptable error range. The computational cost of the coarse grid

CFD is only 30s using a PC with 2.5GHz CPU and 4G RAM, which is about 1% of 52 minutes for the grid independent model.

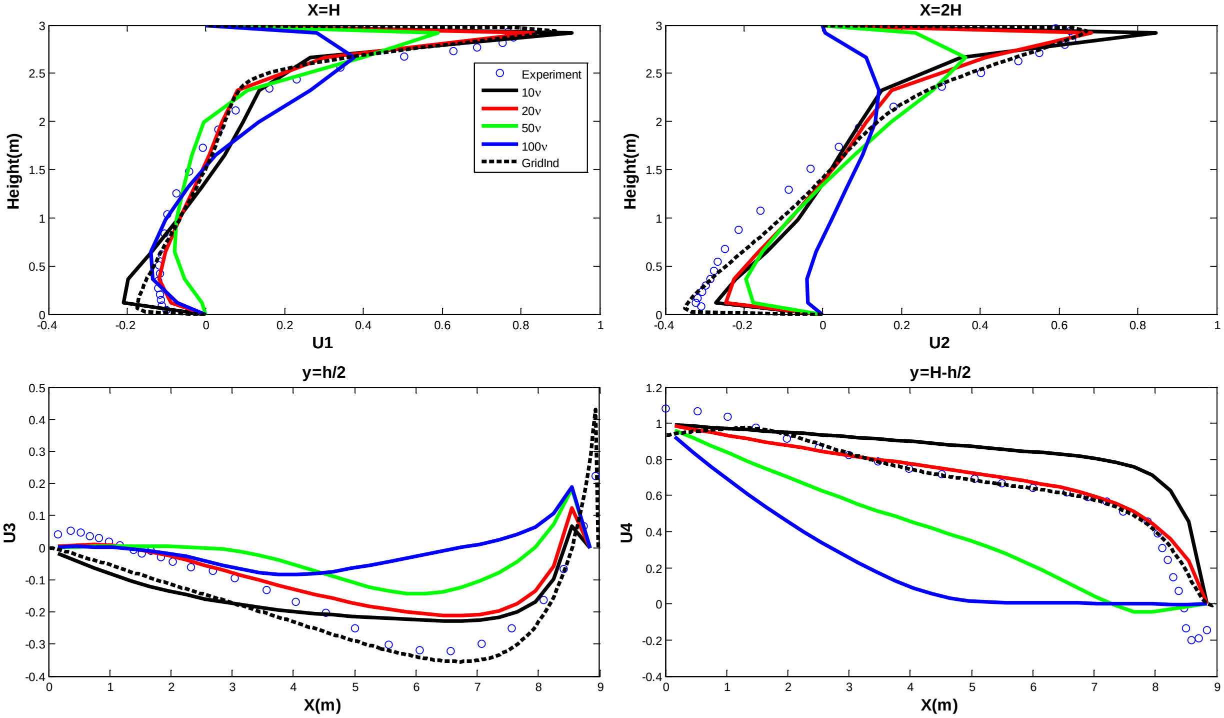


Figure 52. Comparison of predicted results with different effective viscosities with a grid of 17×6

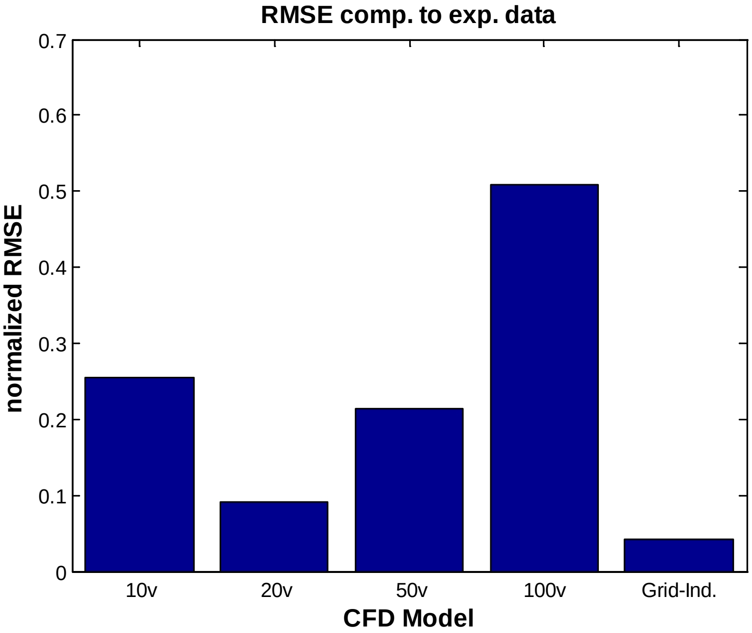


Figure 53. Normalized RMSE comparison of the overall performance of tested CFD models with grid resolution of 17×6

The 2D case tested provides motivation and valuable insight into utilizing the coarse grid to substitute complex turbulence models. It reveals that a uniformly distributed structured grid coupled with a constant turbulence viscosity model may be used to produce a reasonable CFD model for indoor environment modeling. According to the theoretical analysis, finer mesh resolution will generate smaller numerical viscosity; the constant artificial turbulence viscosity to be added to the model shall be greater in order to reach the same viscosity effect.

Figure 54 uses an intermediate fine grid 100×35 (comparing to a fine grid of 300×100 and the initially tested coarse grid 17×6) with different constant viscosities to validate this conclusion. Figure 55 shows the RMSE value to estimate the overall performance of each CFD model through the comparison against available experimental data. These two figures indicate that a constant viscosity at about 200ν provides the overall best prediction. The benefit of using this relatively finer grid is that the flow pattern is better presented as compared to the coarse grid in the previous simulation.

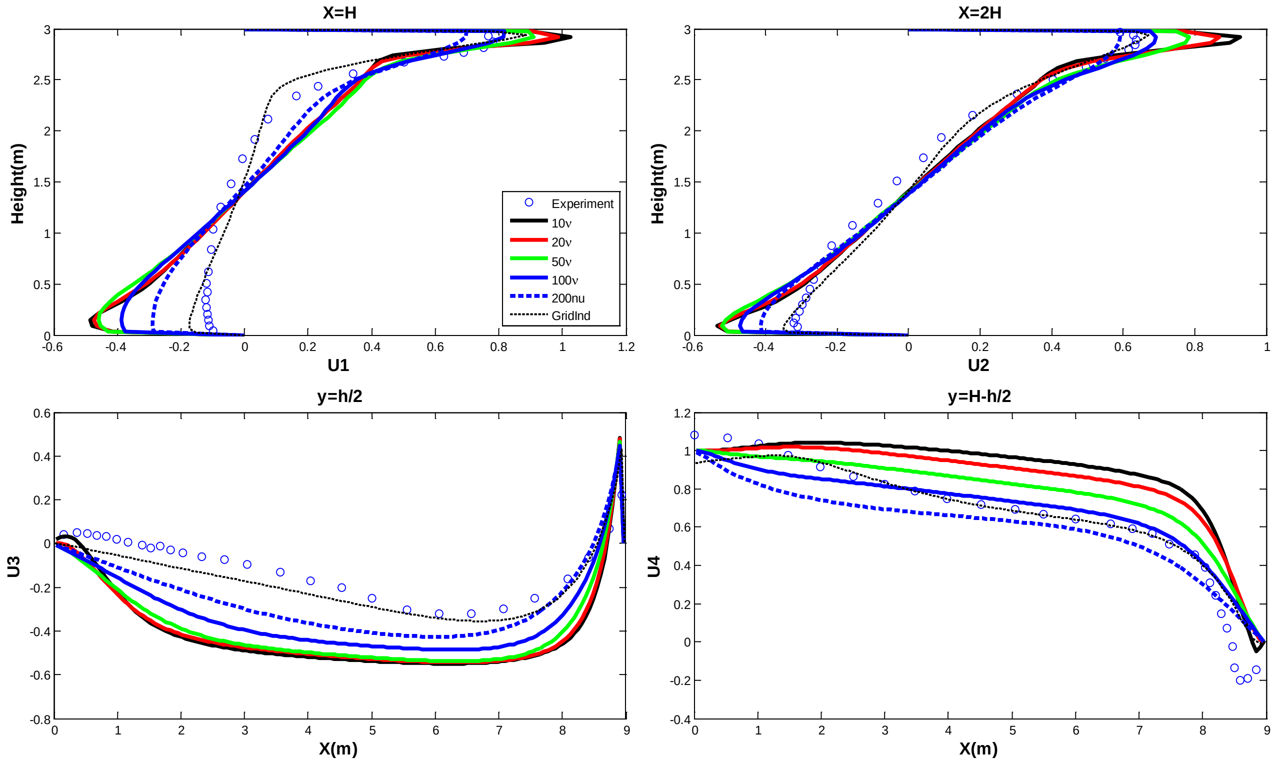


Figure 54. Comparison of predicted results with different effective viscosities with a grid resolution of 100×35

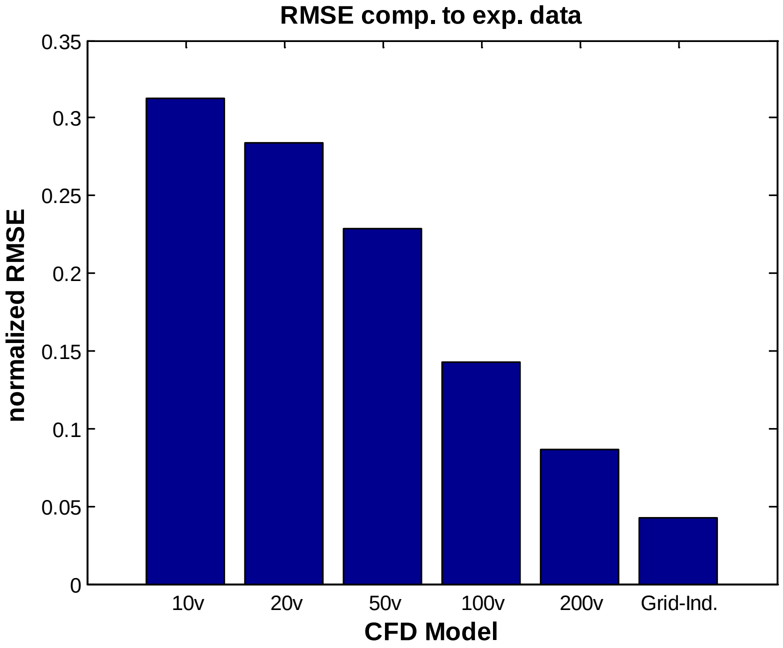


Figure 55. Normalized RMSE comparison of the overall performance of tested CFD models with grid resolution of 100×35

As one of the representative test using different coarse grid resolutions working with different amount of constant viscosity values, Figure 56 and Figure 57 present the profile and normalized RMSE value validation of grid resolution of 60×20 with constant viscosity model results against both experimental data and grid independent solution using sophisticated turbulence model. With slightly coarser grid than the previous test (100×35), according to the hypothesis, grid induced viscosity value will be more dominant, the necessary constant viscosity value to obtain the most accurate CFD solution should thus be smaller than the previous test (200ν). In this case, 100ν provides the best overall prediction results and proves the validity of the hypothesis.

While different density of grid resolution has its own sweet point of constant viscosity, and the computing speed is directly related to grid resolution, it is critical to identify the least grid resolution of typical indoor environment and find out the proper constant viscosity works with it. For these 3 groups of tests, 40×12 generally is the least grid resolution that provides enough grid points to capture the flow pattern and 100ν works best with it.

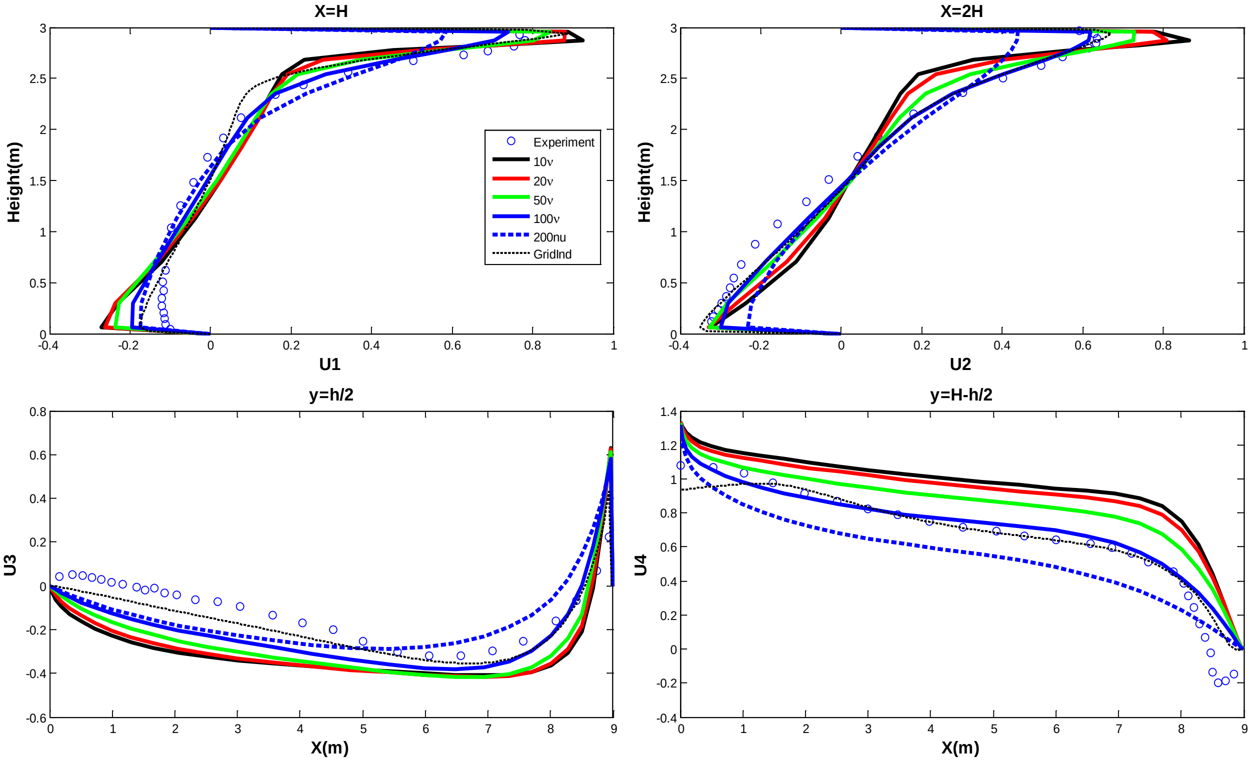


Figure 56. Comparison of predicted results with different effective viscosities with a grid resolution of 40×12

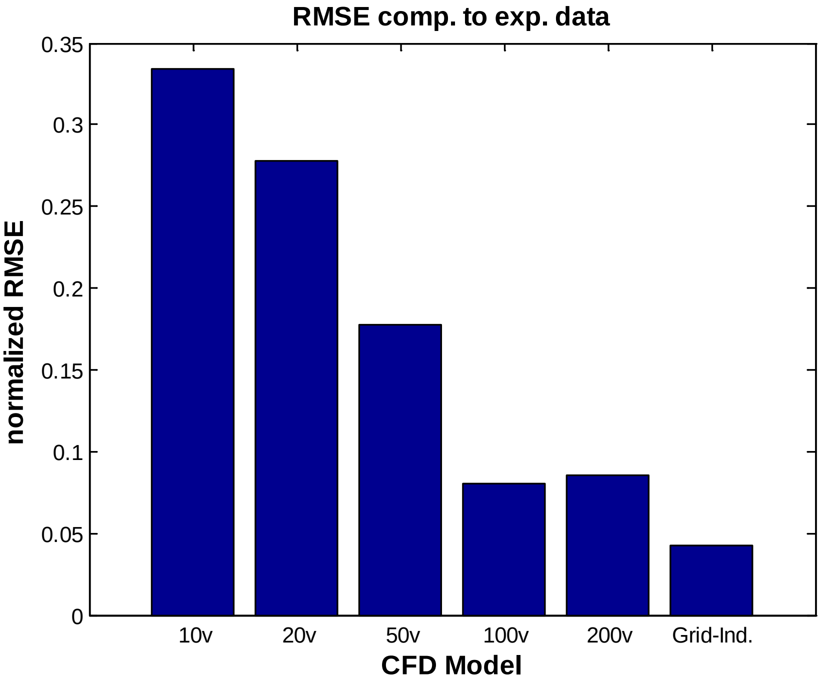


Figure 57. Normalized RMSE comparison of the overall performance of tested CFD models with grid resolution of 40×12

Nevertheless, the grid size in this demonstration case is pre-calculated by knowing the turbulent viscosity magnitude and distribution. Practically, turbulent viscosity is unknown a priori. However, proper grid and constant viscosity specifications may be pre-determined by analyzing the common flow characteristics in typical indoor environments and via extensive numerical experiments. Most indoor airflows are turbulent with a relatively low Reynolds number and heat transfer induced convection is usually weak. Adequate grid is needed to capture the flow features and gradients of interest, which in the tests is 40×12.

A general rule of thumb, based on numerous simulations of the tested cases, suggests that a proper grid size for indoor environment modeling is at the magnitude of 1/20 to 1/10 of the characteristic length of indoor space and the constant viscosity is at the magnitude of 100ν.

5.2.2 Three Dimensional Case Validation

In order to verify and demonstrate this general rule, the 3-dimensional office space [107] with typical configurations used in previous chapters, is modeled and the simulation results are compared with the experimental results. As a recall, this side-wall supply displacement ventilated office is as shown in Figure 58. The space has a dimension of 5.16m×3.65m×2.43m. The diffuser provides a ventilation rate of 4ACH for the space, indicating the supply air flow rate Q=183.1m^3/h. The supply air temperature is T=17.0℃. The effective inlet area ratio is 10%, making the supply air velocity of the diffuser equal to 0.864m/s. Heat fluxes of different heat sources are obtained from the original paper. Experimental data including velocity and temperature along vertical poles in the positions shown in Figure 59 are acquired for model validation.

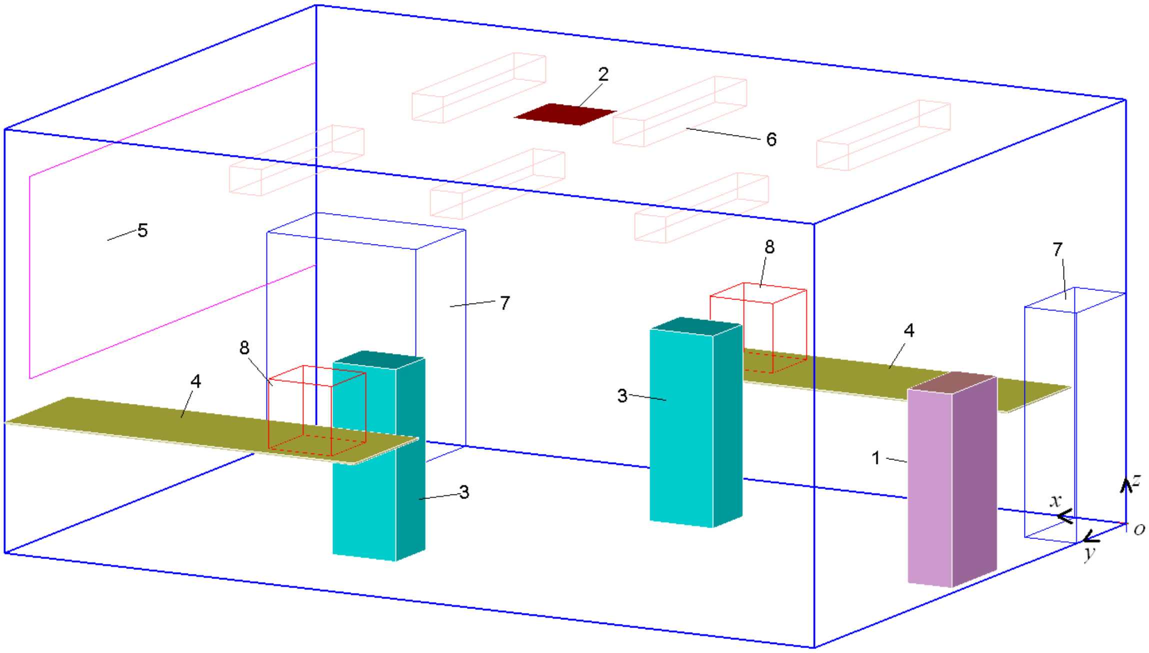


Figure 58. Configurations of the side wall air-supply displacement ventilation case (1. Inlet 2. Outlet 3. Person 4. Table 5. Window 6. Lamp 7. Cabinet 8. Computer)

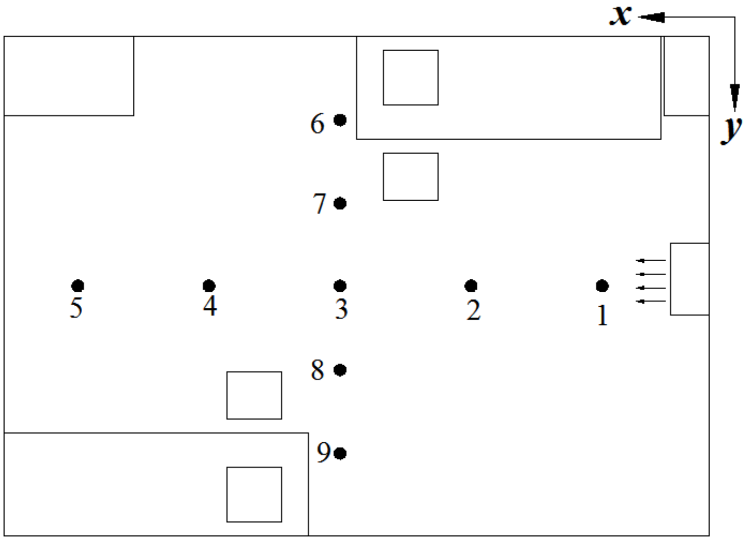


Figure 59. Positions of variable profile comparison with experimental data available along vertical poles of P1 through P9

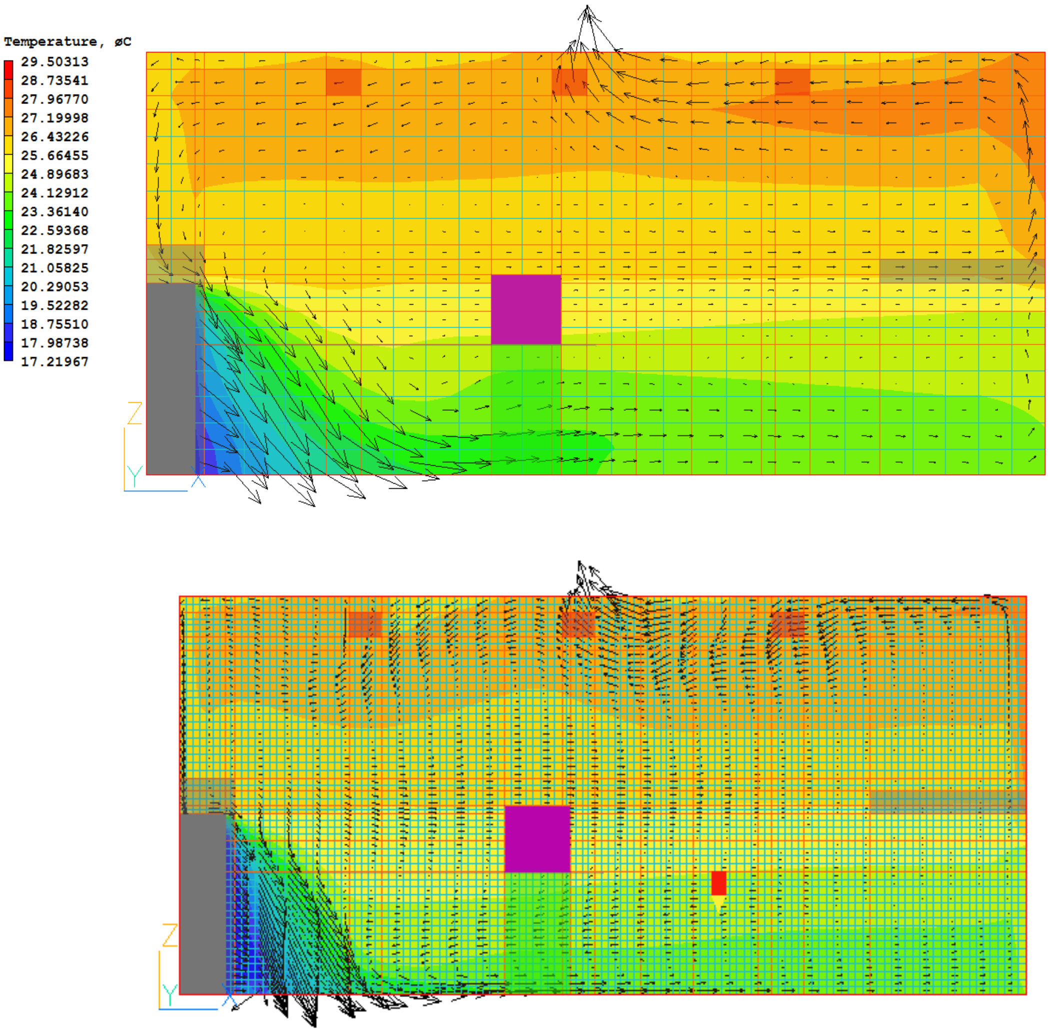


Figure 60. Grid specification and simulation results with coarse and fine grid

The grid resolutions shown in Figure 60 are used for simulations. A laminar flow model combined with a 100ν constant viscosity is used for the coarse grid case and the RNG k-ε model is used for the fine grid case. Airflow (vectors) and temperature (contour) distributions are presented in Figure 60. The grid size for coarse grid is around 1/20 to 1/10 of the height of the room. The results show that the coarse grid coupled with the constant viscosity model can provide similar results as the fine grid. Figure 61 presents the comparison of velocity and temperature profiles on P2, P4, P6, and P8, between the simulated and the measured results. The comparison reveals that both the coarse and fine grids predict the flow and temperature profiles close to the actual measurement.

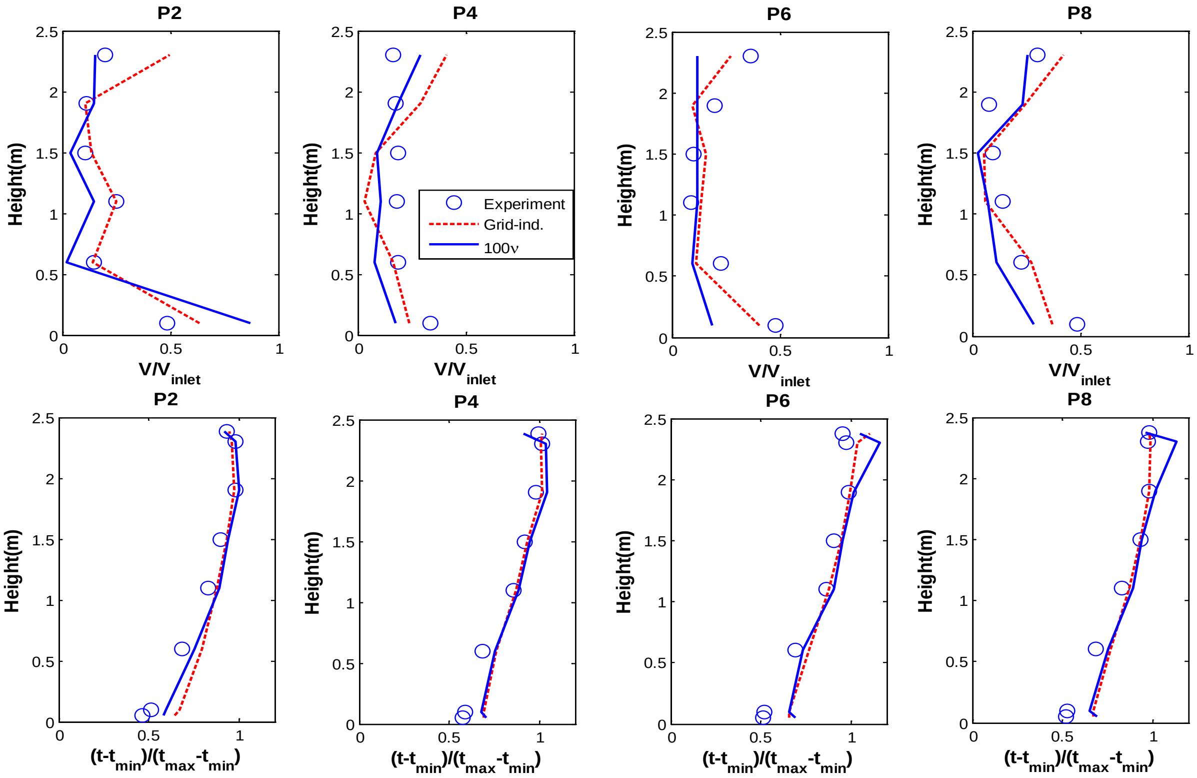


Figure 61. Profile comparison of the coarse and fine grid simulations with experiments

Figure 62 shows the calculated normalized RMSE indices between simulation and experiment for both velocity and temperature, which indicates the overall performance of the two models. Streamwise in the figure represents the data along positions P1 through P5, whereas a cross-section represents positions P6 through P8. This chart delivers a quick overview of the overall performance of a sophisticated CFD model versus a simplified model, and verifies that simplified coarse grid model may still be able to simulate complex indoor flows accurately. The computing time of the coarse grid CFD is about 20 minutes on a PC of 2.5GHz CPU and 4G RAM while it requires 8 hours for the grid independent one. That is, the computing cost of the coarse grid CFD model is only 4% of that for a grid-independent simulation.

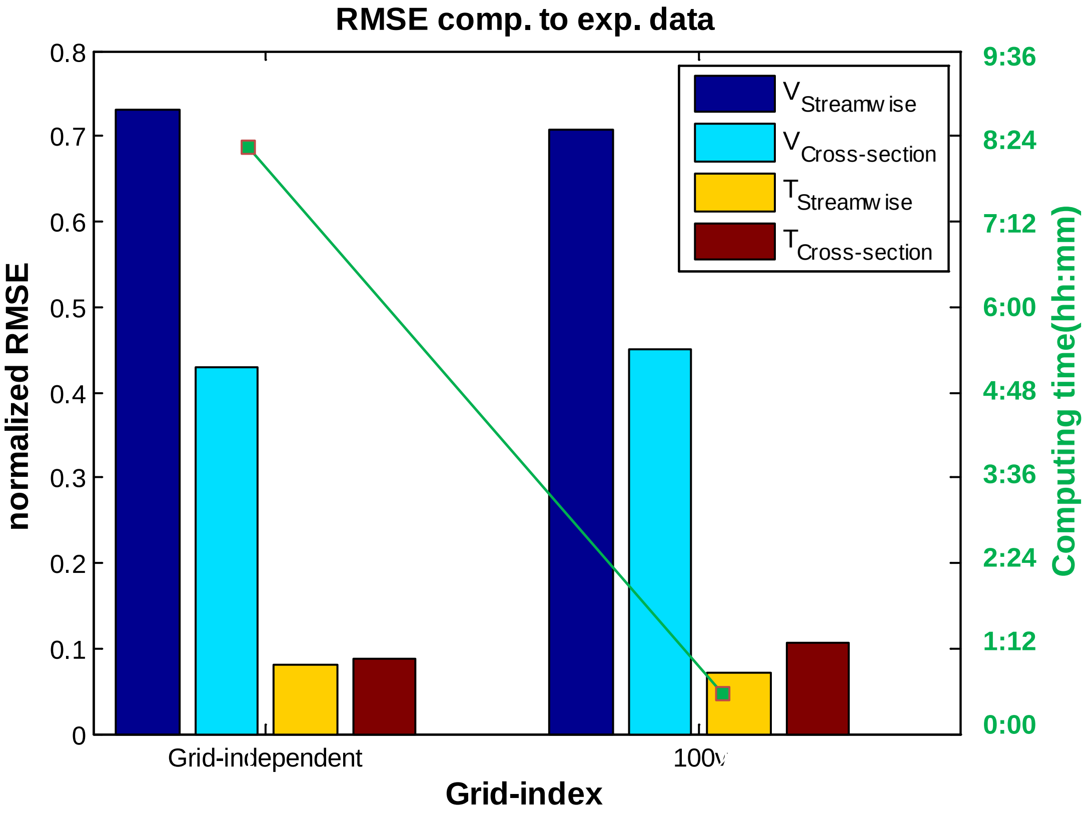


Figure 62. Normalized RMSE value and computing time of grid independent and constant viscosity coarse grid model against experimental data for the 3D office case

5.3 Conclusions

While a lot of previous work[110] focused on using higher order differencing scheme to eliminate the numerical error, this chapter explores the feasibility of the utilization of numerical viscosity induced by a coarse grid in the CFD simulation, combined with designated constant turbulence viscosity, to simulate indoor airflows, through both theoretical analysis and numerical simulation experiment. It is found that a coarse grid, coupled with a constant viscosity model, can greatly reduce the computing cost of the CFD while maintaining reasonable accuracy. A proper coarse grid size that provides satisfactory CFD predictions is roughly between 1/20 to 1/10 of the domain characteristic length, while the designated constant viscosity is about 100 times the air kinematic viscosity. This simplified CFD model will cost around 1-5% computing time of that required by a grid independent simulation with two-equation RANS turbulence models, while the accuracy of the coarse grid CFD is comparable to (or slightly worse than) that of a fine grid CFD with complex turbulence models, which is mostly acceptable for indoor environment study.

CHAPTER 6
TRANSIENT FLOW SIMULATION

Most engineering problems are unsteady or transient in nature; therefore, understanding the instability and flow evolution process is exceptionally important. Since most of the CFD validation practice, especially in the IEQ study area, focuses on steady-state simulation results, it is of great benefit to identify or develop algorithms that are computationally inexpensive. Reviews reveal that for transient flow simulation, iterative methods such as the SIMPLE algorithm are computationally expensive, due to the fact that iterations are required within each time step. Non-iterative algorithms such as PISO and FFD provide good alternatives with great perspective to reduce the computing cost for transient flow simulation.

6.1 Existing Algorithm Comparison

6.1.1 Steady-State Simulation Performance Evaluation

According to the literature review, transient airflow simulation draws much less attention than steady-state airflow simulation. To compare the performance of these discussed algorithms, it is important to evaluate the steady-state solution as a starting point.

Figure 63 is a lid-driven cavity under isothermal condition. Benchmark data for different Reynolds numbers are available from experimentation and numerical simulation. The laminar case of Re=100 is used for this validation in order to make the comparison solely on the physics of this case.

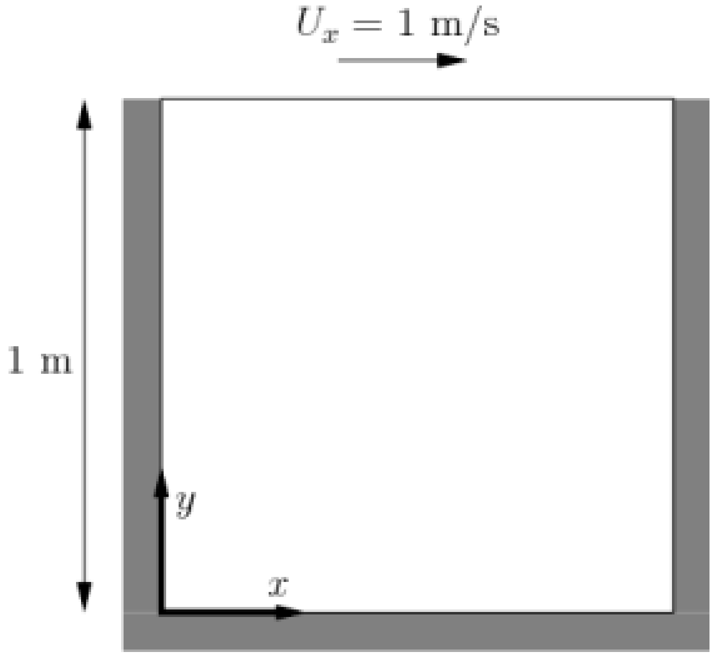


Figure 63. Configuration of lid-driven cavity

The benchmark CFD data from Ghia [111] is used for such validation. The vertical velocity component along mid-height of the cavity and the horizontal velocity component along the mid-width of the cavity are compared.

Due to the availability of CFD code using different algorithms, the SIMPLE and PISO algorithms are tested using openFoam [112], an open source CFD toolbox, under a Linux operating system. FFD used here is a stand-alone C code developed for specific CFD benchmark cases. The SIMPLEST algorithm used by PHOENICS is also included in the comparison. The same grid resolution of 129×129 is employed. The time step size and number of steps (iterations for SIMPLE and SIMPLEST algorithm) used are 0.01 and 2000, respectively. The solutions using these different algorithms are as in Figure 64. It proves that different algorithms give solutions close to each other and agree with benchmark data.

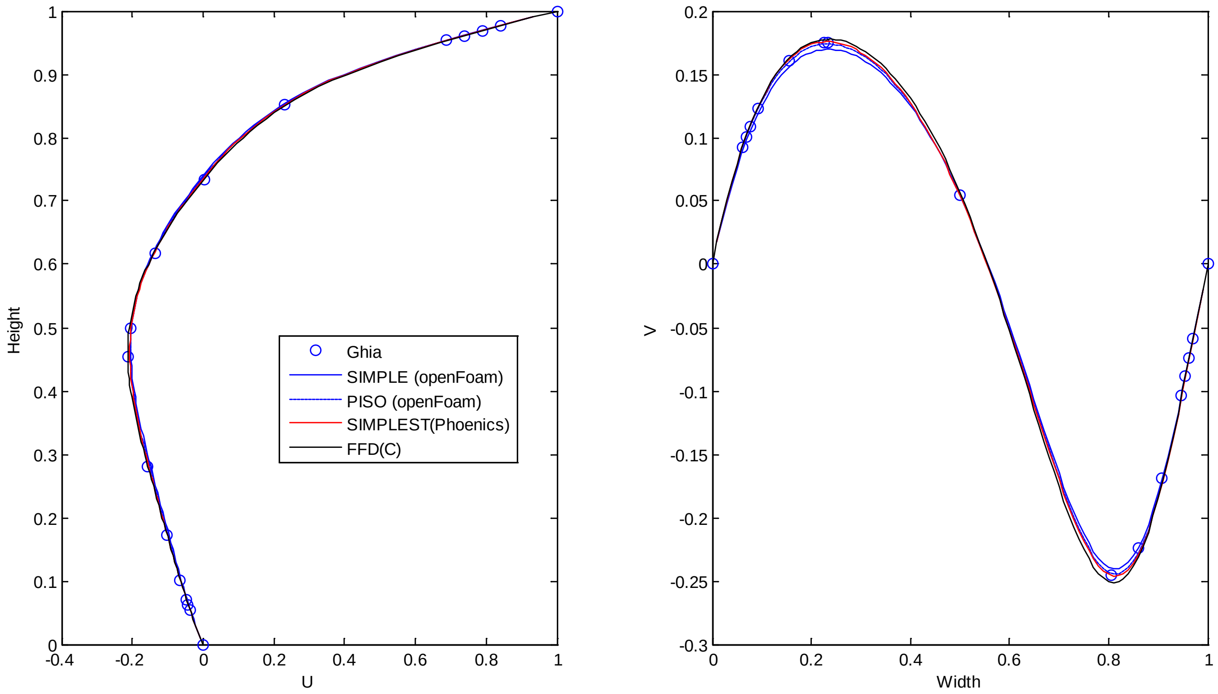


Figure 64. Solutions of lid-driven cavity case with different algorithms

The computing speed of each algorithm is difficult to compare under this circumstance because these simulations are not under the same CFD code and even the same operating system. But for the SIMPLE and PISO algorithms, it will be comparative since these are using the same CFD code under the same operating system.

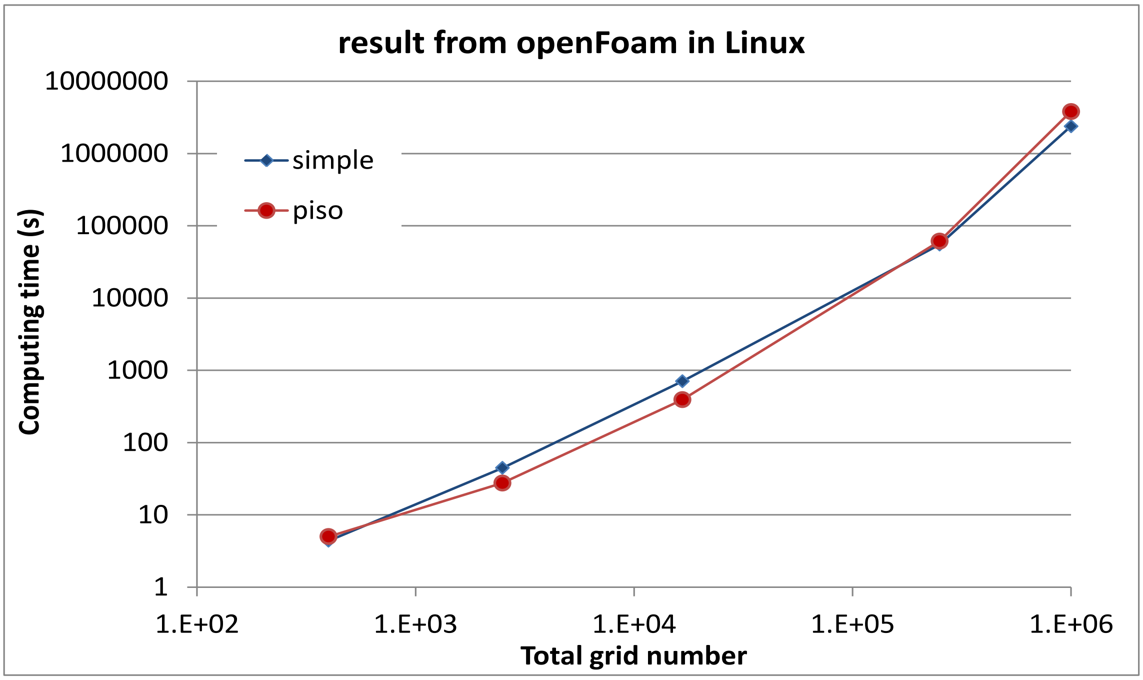


Figure 65. Computing speed comparison between SIMPLE and PISO algorithm

Figure 65 shows that the computing speed of these two algorithms is about the same when parameters of such grid numbers and iteration/step numbers are the same, despite that previous study [55] concludes that PISO can be much faster for calculating isothermal flow. However, the comparison in this figure does not provide any information on the minimum necessary time step number or iteration to reach a convergent solution, and thus it is not conclusive to estimate the computing speed of these algorithms on steady-state CFD simulation.

6.1.2 Transient Performance Evaluation

To evaluate the transient simulation performance of different algorithms, a transient flow simulation of the lid-driven cavity as in Figure 63 was conducted to evaluate the speed of algorithms such as SIMPLE, PISO and FFD. Time step size dt = 0.01s is used. For each time step, intermediate simulation result will be obtained based on the same convergence criterion. The speed of SIMPLE and PISO algorithm is compared using open source software openFoam, and the result is shown in Figure 66. PISO shows significant advantage on computing speed over SIMPLE algorithm. This is mainly because different numbers of iterations are usually required for SIMPLE algorithm to get convergent result; while for PISO algorithm, the convergence will be reached through the second pressure correction step.

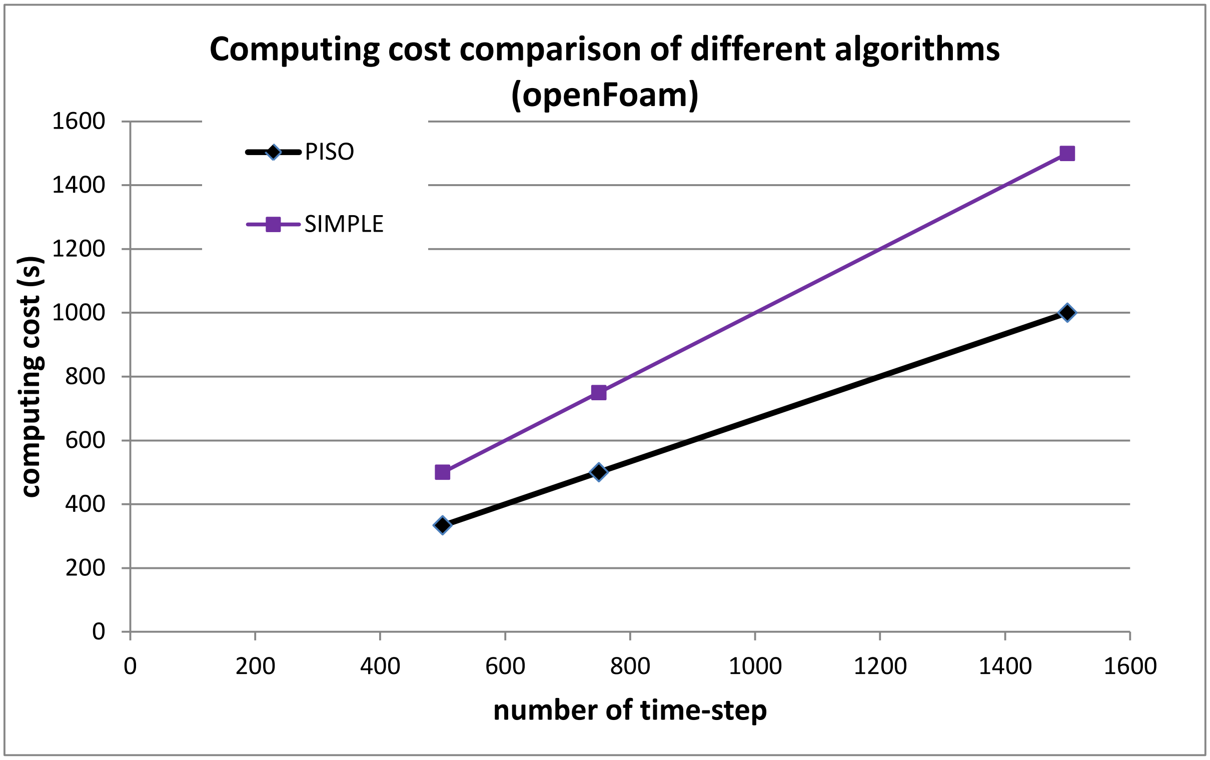


Figure 66. Computing speed comparison between PISO and SIMPLE algorithm on transient simulation

The PISO algorithm is implemented in C code as FFD. Figure 67 shows the speed comparison for different numbers of time steps used. PISO is much slower than FFD as illustrated in this comparison, mainly because for the advection term of the momentum equation, the coupling between velocity components in each direction requires inner iterations to solve. While for the FFD algorithm, since a semi-Lagrangian scheme is used for the advection term, no coupling or iteration is engaged.

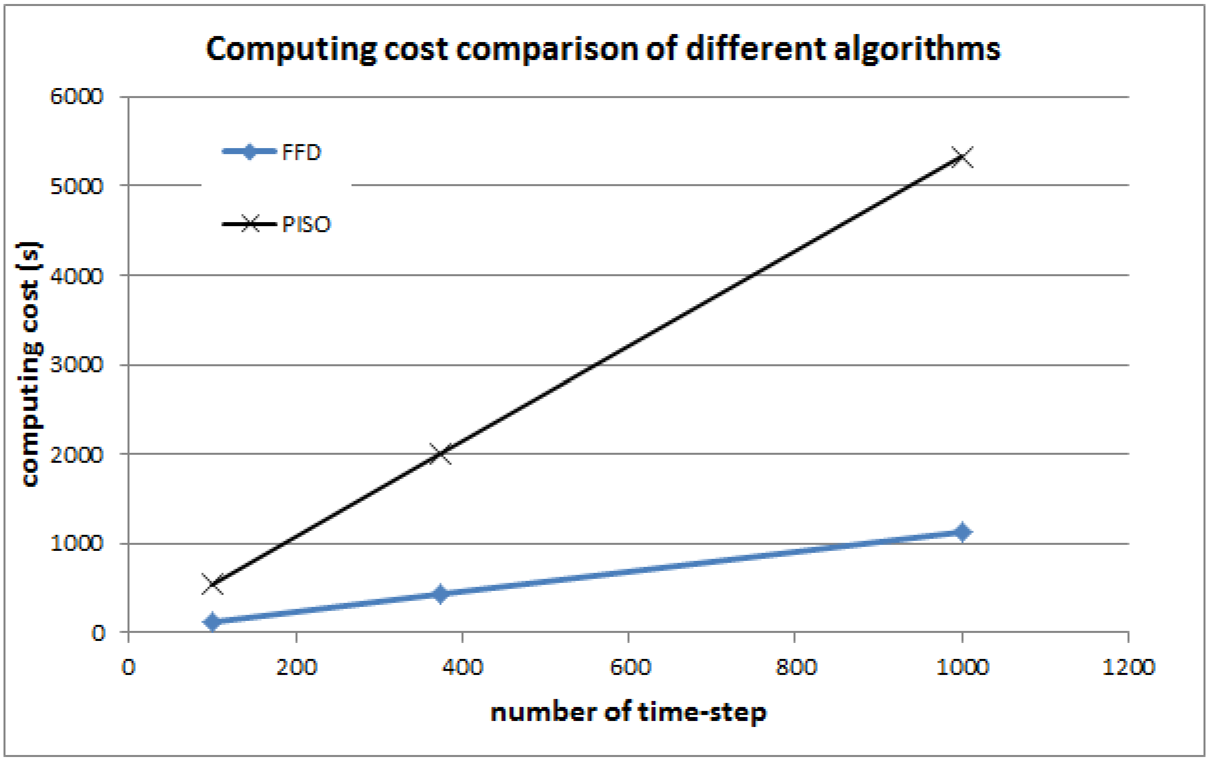


Figure 67. Computing speed comparison between PISO and FFD algorithm on transient simulation

It is the primary interest of this research to identify an appropriate algorithm that meets the transient airflow simulation purpose. An oscillatory lid-driven cavity with transient simulation benchmark data is used to validate the transient simulation performance. The configuration of this oscillatory lid-driven cavity is as Figure 68[113].

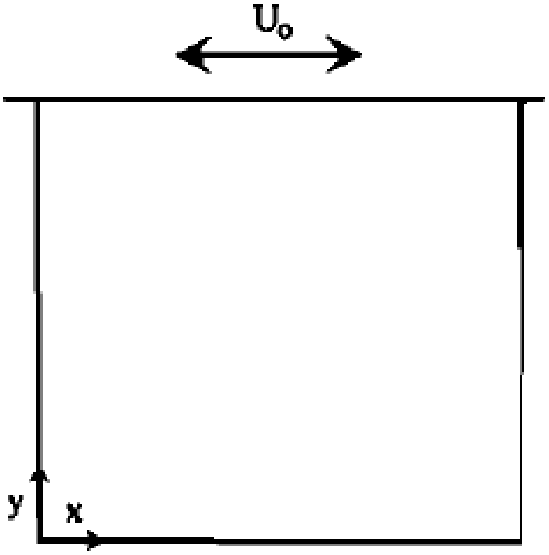


Figure 68. Configuration of the oscillatory lid-driven cavity

The benchmark data is obtained with a grid resolution of 121×121 and time step size of $2x10^{-4}$. The same grid resolution is employed in the validation. The moving wall on top of the cavity has oscillatory velocity following sinusoidal shape as

$$U_0 = \sin(\omega t)$$

In this particular case that is used as a benchmark, $\omega = \pi$ and $Re = 500$. The variation of the oscillatory lid boundary is as in Figure 69.

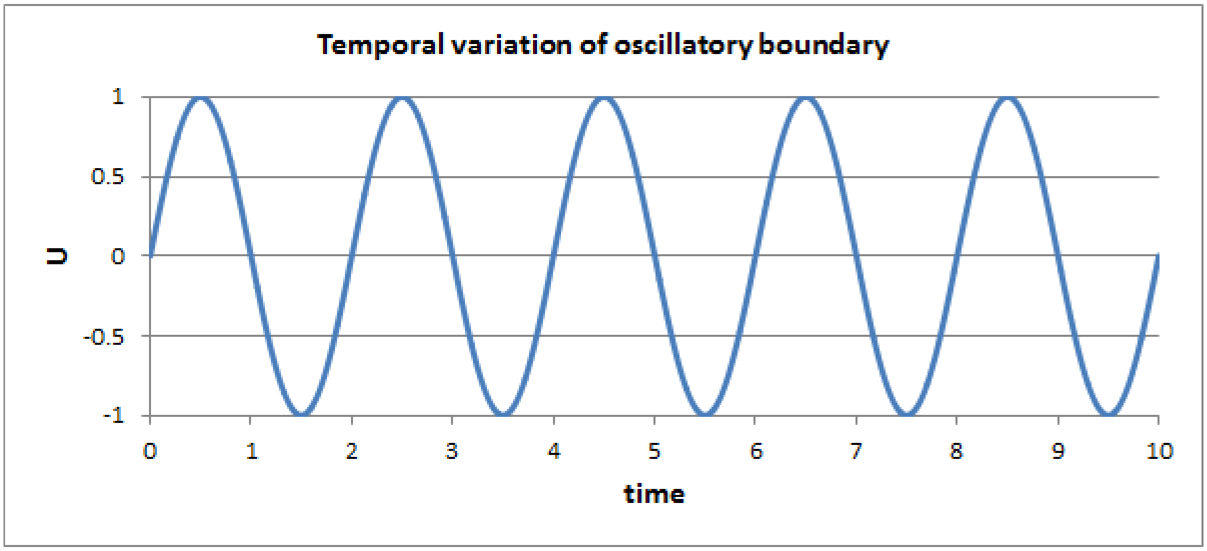


Figure 69. Oscillatory velocity of the lid boundary

At the mid-width of the cavity, the horizontal velocity component at different heights (0.25, 0.5 and 0.75) has temporal variation data. This data set is used in the validation that employs the two different algorithms, PISO and FFD. The time step size tested in such validation is the same as that used in the benchmark, which is $2\text{x}10^{-4}$. The predictions with these two algorithms are nearly identical with such a small time step. The time step size is thereafter increased gradually in order to test the performance of these two algorithms on a larger time step size.

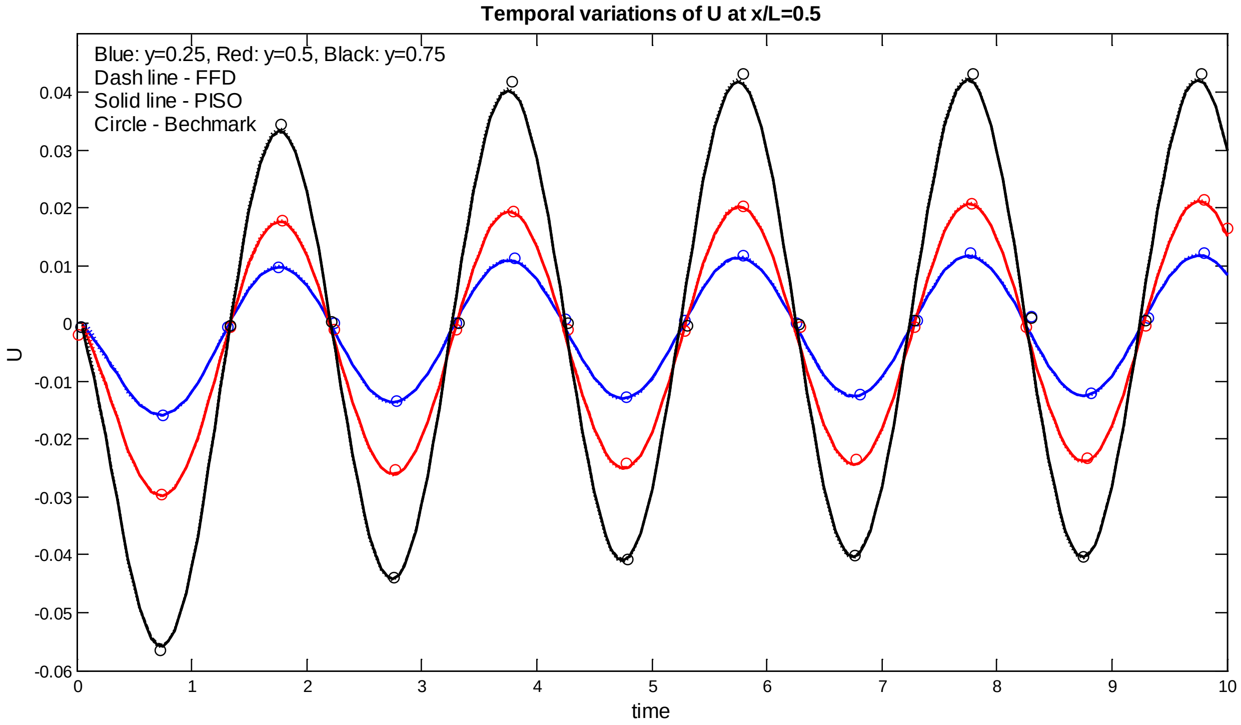


Figure 70. Comparison of PISO and FFD result on isothermal lid-driven at time step = 0.01

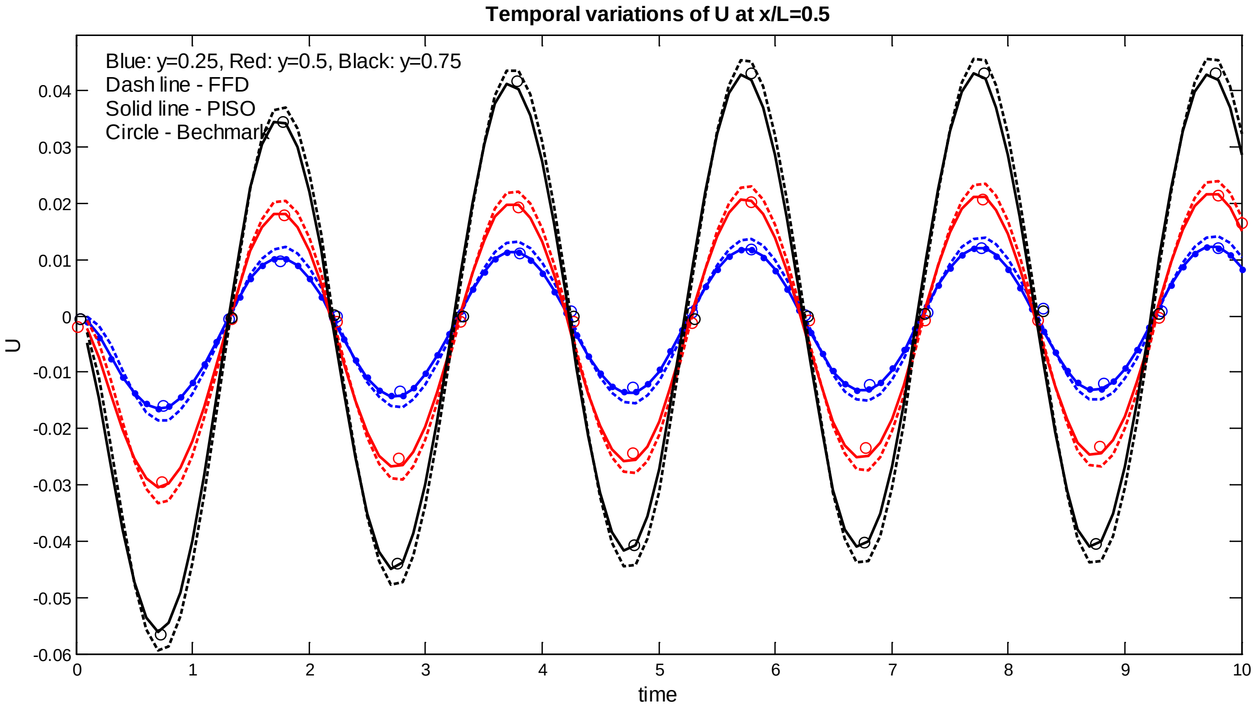


Figure 71. Comparison of PISO and FFD result on isothermal lid-driven at time step = 0.1

When the time step is small (0.01 as in Figure 70), the solutions of the two algorithms are not visible, and they both agree with the benchmark data; however, when the time step size becomes 0.1 as in Figure 71, the FFD prediction deviates from the benchmark, while PISO solutions are still close to the benchmark data.

Such an outcome that FFD has less tolerance on the time step size may be explained from the algorithm perspective. FFD utilizes splitting of operators, which treats each operator (advection, diffusion, source and pressure) as an external force to evolve the flow field, and all of which are added to the calculation of evolution one by one. The projection step of FFD calculates the pressure field solely from the velocity field, aiming to force mass conservation and ignoring the effect of historical pressure.

6.1.3 Conclusion

The candidates for fast transient flow simulation are tested and the performances are evaluated. Transient flow simulation algorithm FFD shows great advantage over the PISO algorithm in terms of computing speed. However, FFD is less accurate when the time step is large.

6.2 Semi-Lagrangian PISO Algorithm

6.2.1 Algorithm Derivation

A fundamental difference between FFD and PISO is on the projection. The FFD algorithm neglects the influence of pressure from the previous time step and assumes pressure solely determined by the velocity field under the continuity restriction. The accuracy of PISO, theoretically and practically, has advantage over FFD. For the FFD algorithm, under which the advection term is completely separated from the right hand side of the momentum equation, it makes it possible to utilize the semi-Lagrangian algorithm, which is faster and more stable compared to the conventional method of directly solving the advection equation. Under such inspiration, an algorithm integrating semi-Lagrangian advection to the PISO algorithm is proposed as follows.

Step 1

Use semi-Lagrangian advection ($P(x, -\mathit{\Delta} t)$) to get an intermediate velocity field.

$$\frac{u^1 - u^n}{\Delta t} = -(u \cdot \nabla)u \Rightarrow u^1 = u^n[P(x, -\mathit{\Delta} t)] \tag{6.1}$$

Step 2

With intermediate flow field u^1 and initial pressure field p^n, the intermediate field value u^2 can be obtained using the implicit scheme as Eq. (6.2) by adding other operators to velocity increment equation

$$\frac{u^2 - u^1}{\Delta t} = \nu \nabla^2 u^2 - \frac{1}{\rho} \nabla p^n + S \tag{6.2}$$

Since this is using p^n instead of p^{n+1}, u^2 will not satisfy the continuity equation.

Step 3

A second-step intermediate approximation of velocity is obtained from assuming an updated velocity field u^2 and pressure field p^1. Eq. (6.2) becomes

$$\frac{u^3 - u^1}{\Delta t} = \nu \nabla^2 u^2 - \frac{1}{\rho} \nabla p^1 + S \tag{6.3}$$

Eq. (6.3) subtract Eq. (6.2) yields

$$\frac{u^3 - u^2}{\Delta t} = -\frac{1}{\rho} (\nabla p^1 - \nabla p^n) \tag{6.4}$$

Take divergence for both sides of Eq. (6.4), and together with the continuity equation $\nabla u^3 = 0$, such velocity increment Eq. (6.4) yields the pressure increment equation to solve $p^1 - p^n$ field

$$\nabla^2 p^1 - \nabla^2 p^n = \frac{\rho}{\Delta t} \nabla \cdot u^2 \tag{6.5}$$

Step 4

The updated pressure field (or pressure increment field) can be substituted into Eq. (6.3) or Eq. (6.4) to update the velocity field and get the velocity field u^3.

Step 5

A replication of Step 2 is conducted using the updated result from Step 3 u^3 and after-advection initial value u^1, assuming a most updated pressure field p^{n+1}, yields an updated field

$$\frac{u^{n+1} - u^1}{\Delta t} = \nu \nabla^2 u^3 - \frac{1}{\rho} \nabla p^{n+1} + S \tag{6.6}$$

Eq. (6.6) subtract Eq. (6.3) yields

$$\frac{u^{n+1} - u^3}{\Delta t} = -\frac{1}{\rho} (\nabla p^{n+1} - \nabla p^1) + [\nu \nabla^2 u^3] - [\nu \nabla^2 u^2] \tag{6.7}$$

Take divergence for both sides of Eq. (6.7), together with the continuity equation $\nabla u^{n+1} = 0$ and $\nabla u^3 = 0$, such velocity increment Eq. (6.7) yields the pressure increment equation to solve $p^{n+1} - p^1$ field

$$\nabla^2 p^{n+1} - \nabla^2 p^1 = \nabla \cdot [\nu \nabla^2 u^3] - [\nu \nabla^2 u^2] \tag{6.8}$$

Step 6

The updated pressure field (or pressure increment field) can be plugged into Eq. (6.7) or Eq. (6.8) to update the velocity field and approximate the exact solutions u^{n+1} and p^{n+1}. This is proven to be sufficient for most practical purposes.

6.2.2 Sequence of Operation

The computational sequence of the semi-Lagrangian PISO algorithm can be summarized as Figure 72. The proposed semi-Lagrangian PISO algorithm, without the corrector steps (step 5 and step 6), is similar to FFD except that it takes into consideration the pressure field from the previous time step. Such an algorithm is expected to improve the accuracy of FFD without sacrificing much computing speed. Semi-Lagrangian advection algorithm can be expected to reduce the computing cost of direct advection term in PISO algorithm significantly.

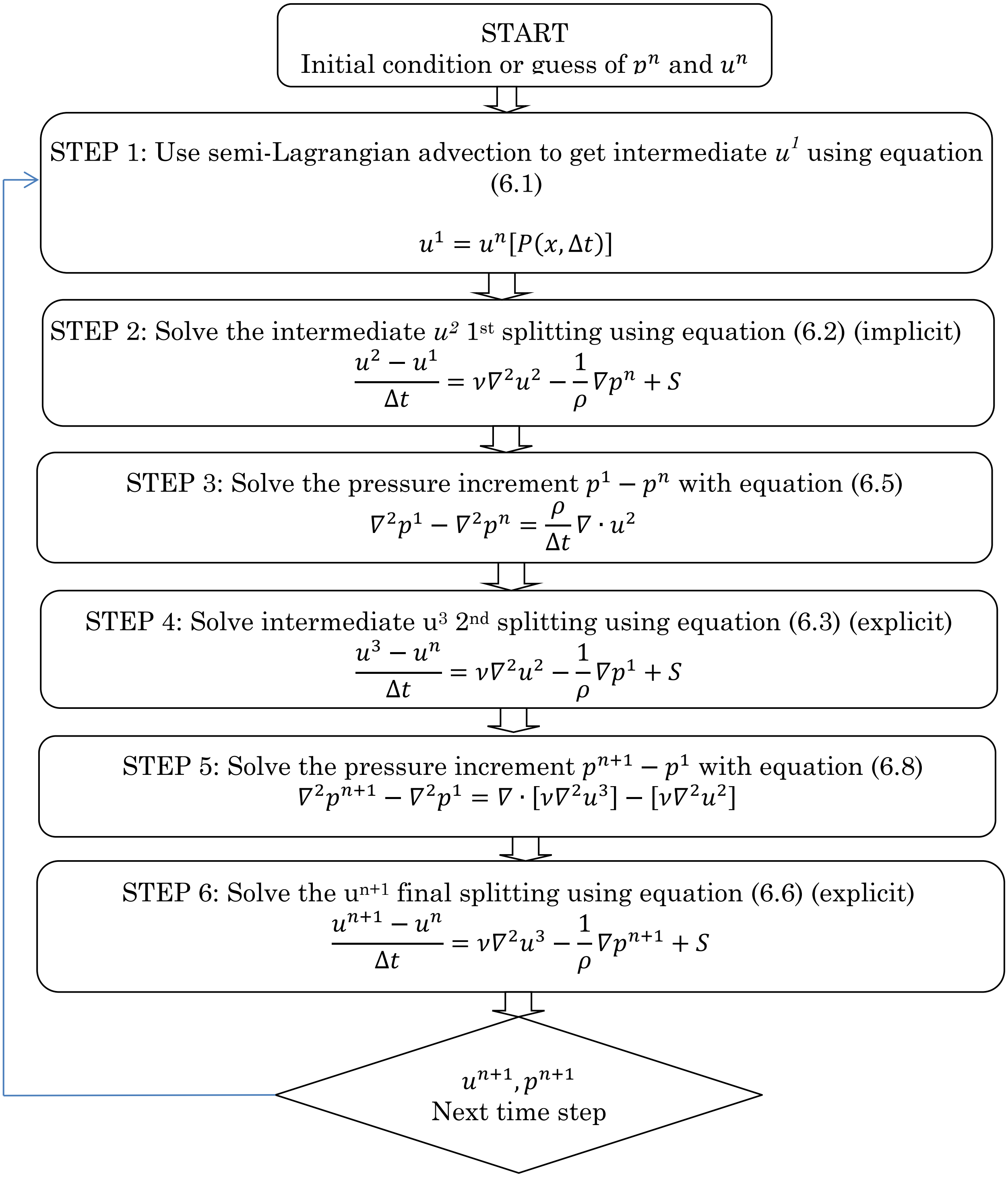

START
Initial condition or guess of p^n and u^n
STEP 1: Use semi-Lagrangian advection to get intermediate u^1 using equation (6.1)
$u^1 = u^n[P(x, \Delta t)]$
STEP 2: Solve the intermediate u^2 1st splitting using equation (6.2) (implicit)
$\frac{u^2 - u^1}{\Delta t} = \nu \nabla^2 u^2 - \frac{1}{\rho} \nabla p^n + S$
STEP 3: Solve the pressure increment $p^1 - p^n$ with equation (6.5)
$\nabla^2 p^1 - \nabla^2 p^n = \frac{\rho}{\Delta t} \nabla \cdot u^2$
STEP 4: Solve intermediate u^3 2nd splitting using equation (6.3) (explicit)
$\frac{u^3 - u^n}{\Delta t} = \nu \nabla^2 u^2 - \frac{1}{\rho} \nabla p^1 + S$
STEP 5: Solve the pressure increment $p^{n+1} - p^1$ with equation (6.8)
$\nabla^2 p^{n+1} - \nabla^2 p^1 = \nabla \cdot [\nu \nabla^2 u^3] - [\nu \nabla^2 u^2]$
STEP 6: Solve the u^{n+1} final splitting using equation (6.6) (explicit)
$\frac{u^{n+1} - u^n}{\Delta t} = \nu \nabla^2 u^3 - \frac{1}{\rho} \nabla p^{n+1} + S$
u^{n+1}, p^{n+1}
Next time step

Figure 72. Operation sequence of semi-Lagrangian PISO

6.3 Validation of SLPISO

The semi-Lagrangian PISO (SLPISO) algorithm brought out is supposed to inherit both speed and accuracy advantages of FFD and PISO. This algorithm is implemented in a C code for validation purposes. The validation will start from the same isothermal transient case to systematically compare the performance of these three different algorithms: FFD, PISO and SLPISO.

6.3.1 Laminar Isothermal Flow

The same oscillatory lid-driven cavity case as in Figure 68 is simulated with SLPISO. The same model configuration as in Chapter 6.1.2 is used for the SLPISO algorithm. With the time step increase from 0.01 in Figure 73 to 0.1 in Figure 75, the prediction of FFD deviates from the benchmark data and that of PISO is more accurate. SLPISO gives as good a prediction as PISO as the time step increases. As illustrated in Figure 74 and Figure 75, when the time step size is 0.04 and 0.1, as the FFD prediction begins to deviate from benchmark data, PISO and SLPISO can still provide good solutions similar to that when the time step is as small as 0.01. The time step size that is greater than 0.1 is not utilized in this validation. That is because it will not even resolve the sinusoidal boundary shape properly, and hence does not provide a meaningful CFD solution comparison for this validation purpose.

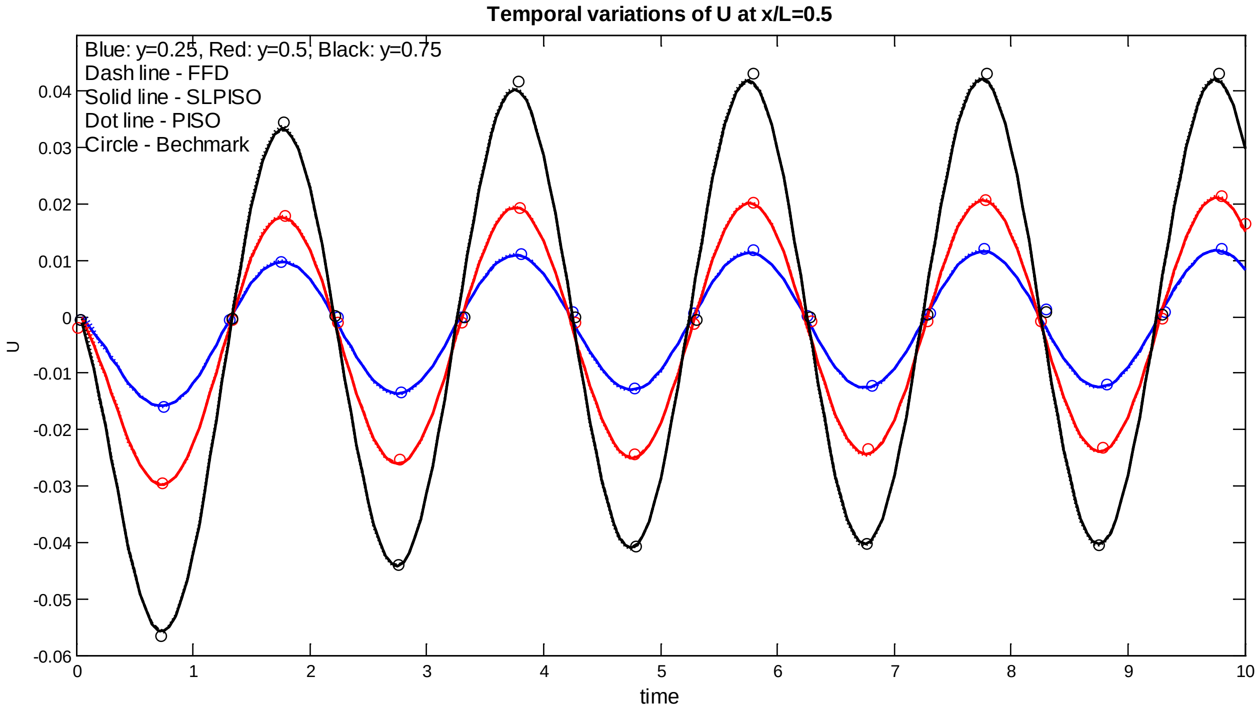


Figure 73. Comparison of FFD, SLPISO and PISO result on isothermal lid-driven at time step = 0.01

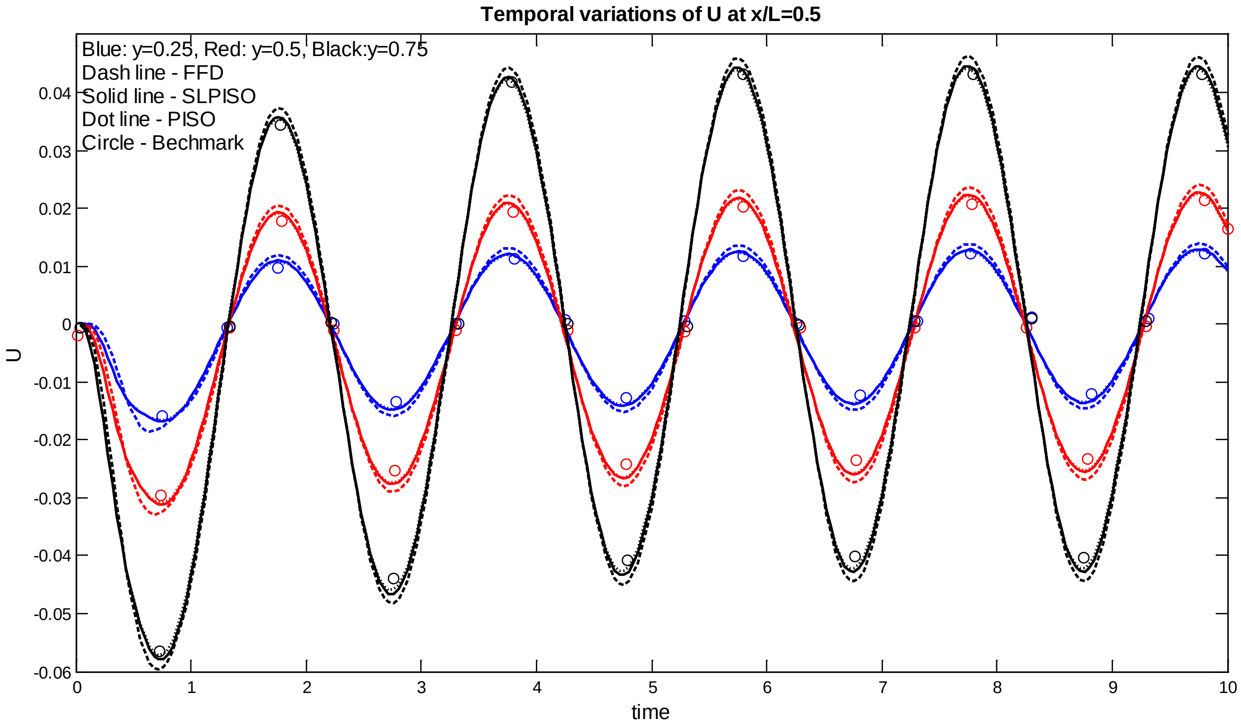


Figure 74. Comparison of FFD, SLPISO and PISO results on isothermal lid-driven at time step = 0.04

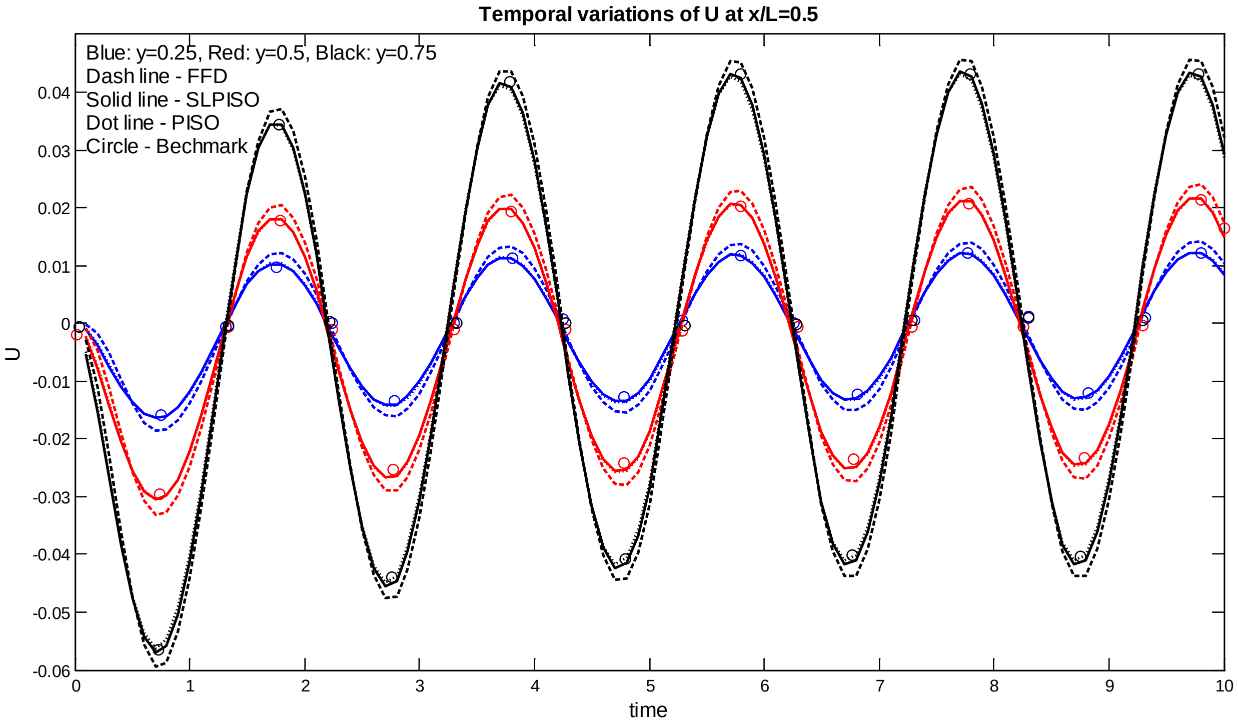


Figure 75. Comparison of FFD, SLPISO and PISO results on isothermal lid-driven at time step = 0.1

Computing speed is another critical parameter in this comparison. SLPISO, which adopts the semi-Lagrangian advection idea of FFD, will theoretically require much less computing resources than PISO. This is verified by the computing speed comparison in Figure 76. The SLPISO algorithm consumes about 1/3 more computing time than FFD, mainly because of the second pressure correction procedure employed by SLPISO.

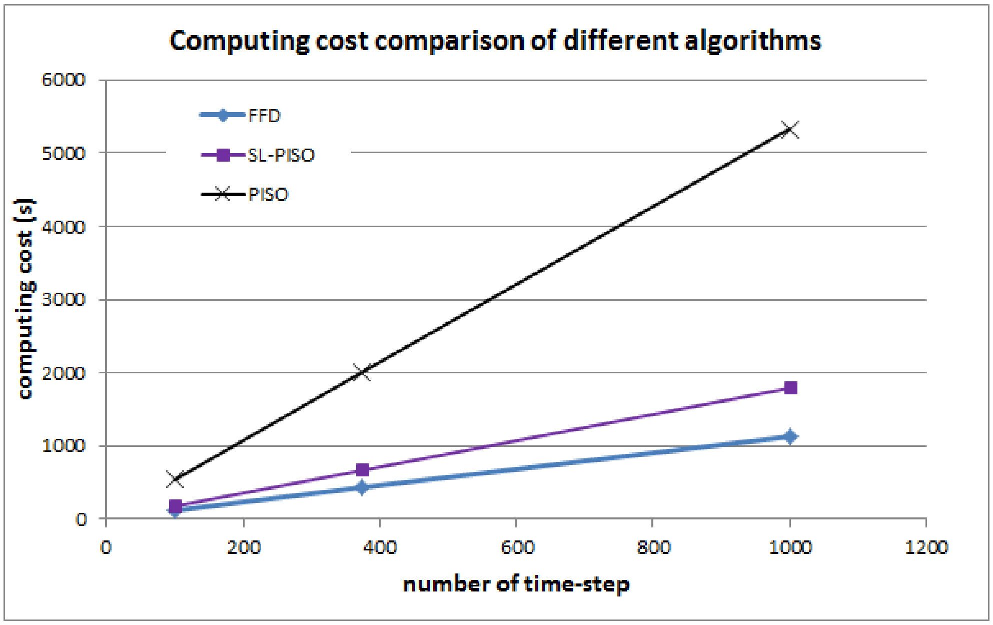


Figure 76. Computing speed comparison of FFD, SLPISO and PISO algorithms

This well-defined oscillatory lid-driven cavity case provides good benchmark data for the validation of time-dependent CFD code as in this study. The relatively simple configuration excludes the factors from impacting the accuracy of the CFD solution other than the algorithm itself. The validation illustrates that SLPISO, brought out by this research, can provide as accurate a prediction as PISO for isothermal airflow simulation.

6.3.2 Laminar Natural Convection Flow

In order to validate the transient simulation performance of the new algorithm on temperature linked problems, a differential heated cavity with time-dependent benchmark data is employed. The configuration of such a case is shown in Figure 77 [113]. Two vertical walls have different temperatures, causing a circulation in the cavity. Dimensionless parameters for this cavity are: Gr=2x10^6 and Pr=0.02. The time step used in the benchmark is 1x10^{-6}. Different grid resolutions of 241×241, 121×121 and 61×61 are tested. The grid resolution 121×121 result is very close to that of 241×241, thus it is used in the validation test. The horizontal velocity component variation with time at the center of the cavity is monitored and shows oscillatory shape with time elapses. Strong oscillation during time 0-1 is observed and the velocity stabilizes at around zero after time 2.

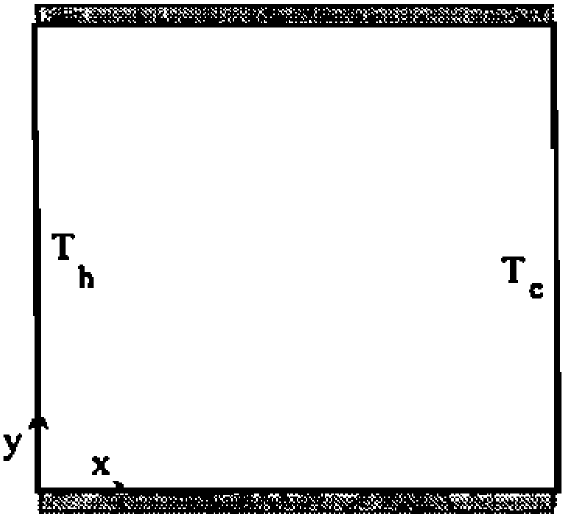


Figure 77. Configuration of natural convection flow cavity for transient simulation validation

Different time steps are tested to compare the performance of algorithms, especially the tolerance on the increase of time step size. Figure 78 shows the solution of FFD and SLPISO with time step 2×10^{-6} compared with benchmark data. It shows the variation of the horizontal velocity component at the center of this cavity. With time elapses, such velocity asymptotically approaches constant amplitude; however, the transient period between dimensionless time 0 to 1 is not possibly being captured in a steady-state CFD simulation.

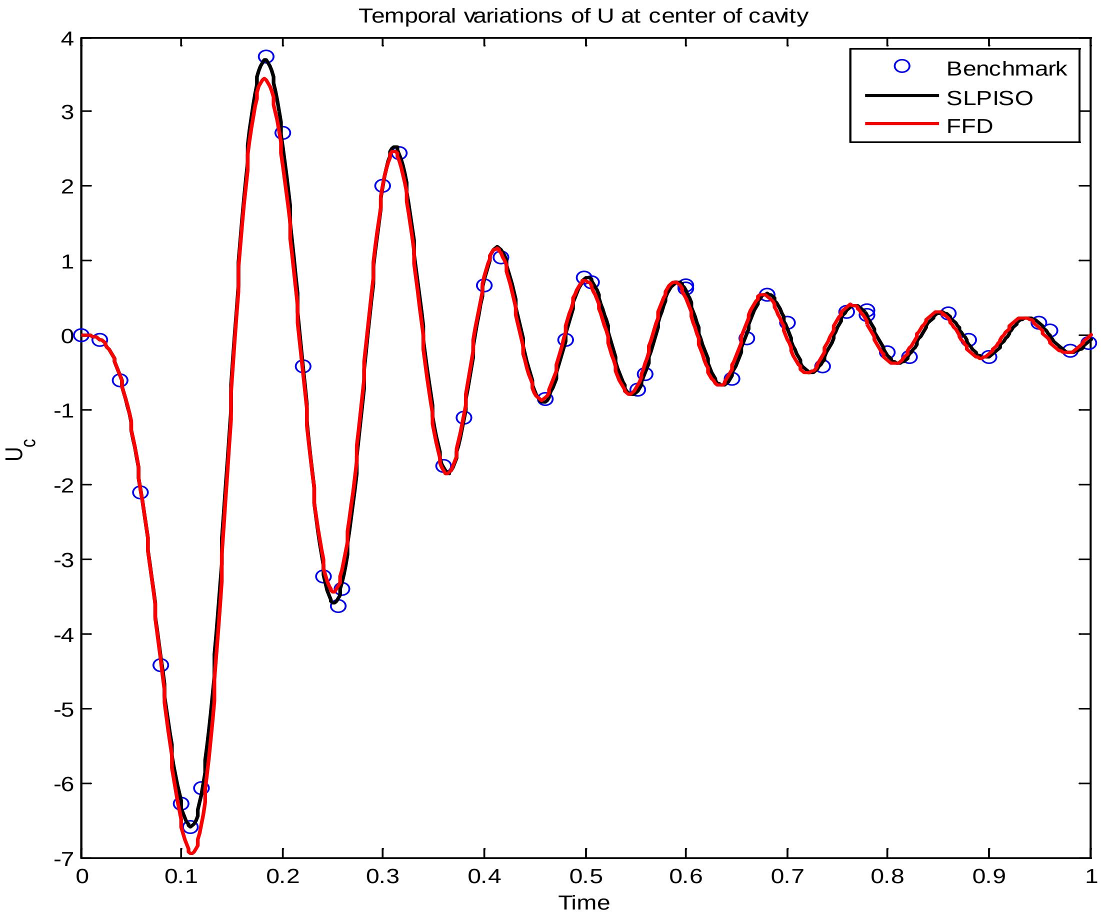


Figure 78. Comparison of FFD and SLPISO result on natural convection cavity at time step = 2×10^{-6}

This plot shows at the beginning of the oscillation, SLPISO solution is closer to benchmark data than that of FFD when time step size is as small as 2×10^{-6}. But the two solutions show no visible difference after around time 0.4.

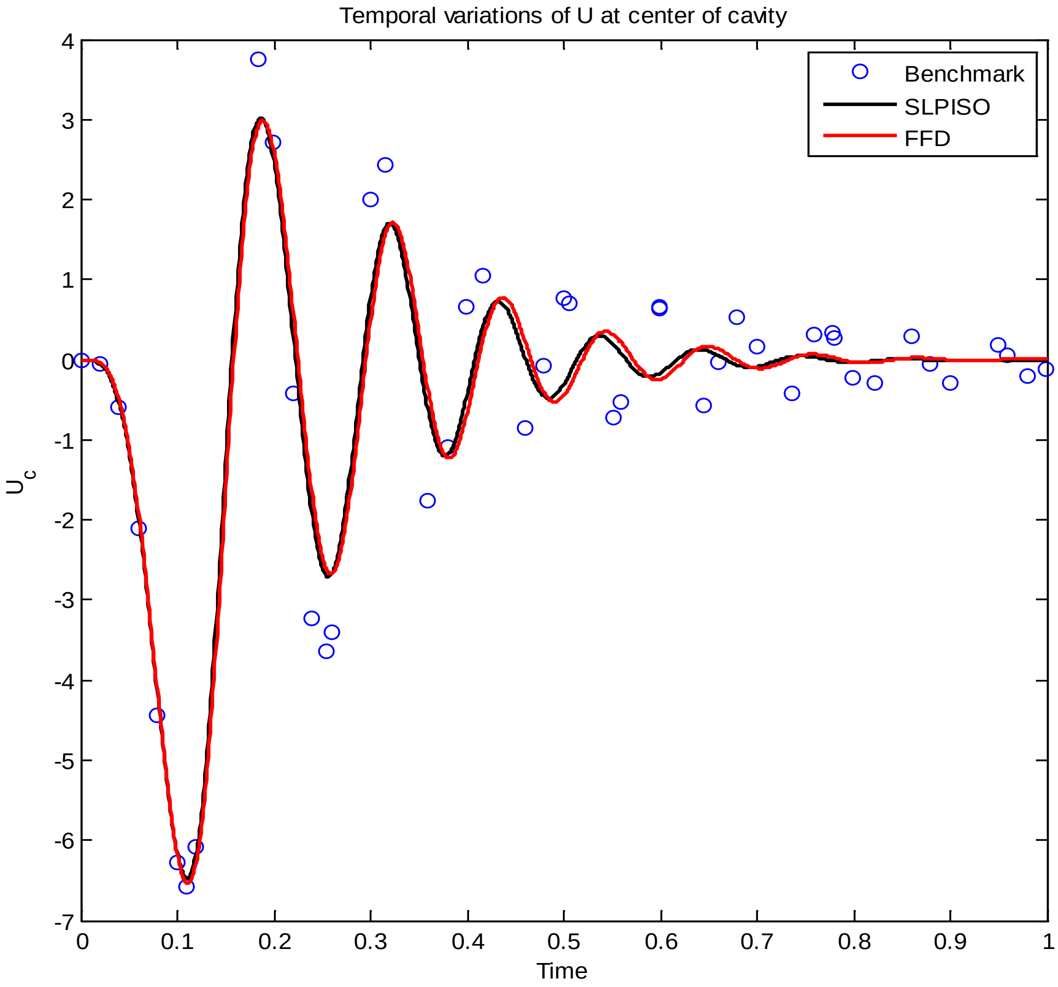


Figure 79. Comparison of FFD and SLPISO result on natural convection cavity at time step = 1x10^{-3}

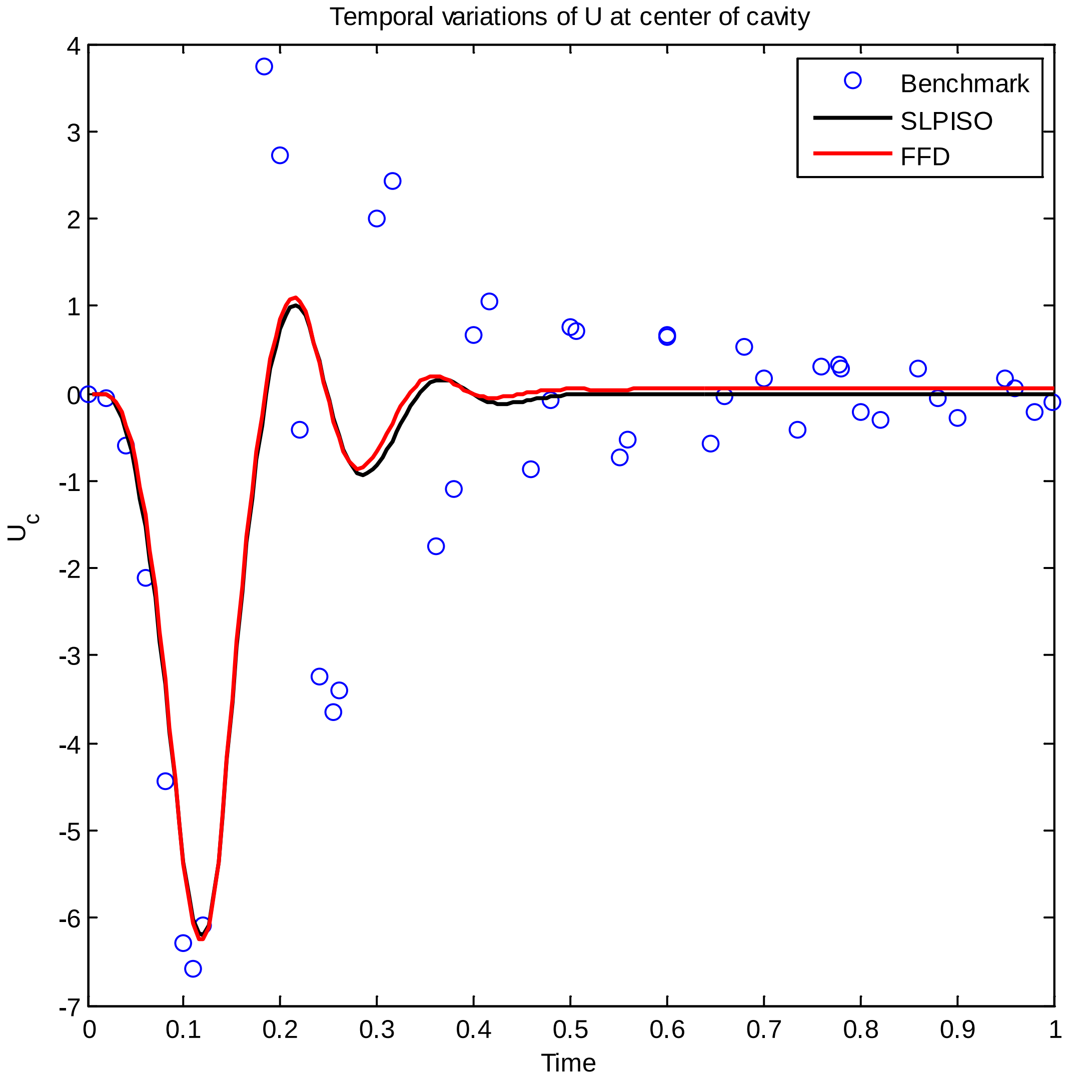


Figure 80. Comparison of FFD and SLPISO result on natural convection cavity at time step = $5x10^{-3}$

Figure 79 shows similar comparison with time step 1×10^{-3}. When the time step goes greater, both solutions deviate from the benchmark solutions. Figure 80 is the solution comparison at time step 5×10^{-3}. None of the two algorithms can capture the velocity decaying shape. This increment on time step size does not affect the prediction accuracy for steady state simulation, but when time step size is small enough, SLPISO shows better performance for capturing the evolution of transient flow at initial time periods.

The well-defined laminar flow cases used in this validation practice are ideal for testing the performance of different CFD algorithms since they can exclude the factors that impacted the CFD solution other than the algorithm itself. The result shows that for transient airflow simulation, SLPISO delivers a better solution than the FFD algorithm in capturing the transient flow pattern. For isothermal airflow simulation, SLPISO can tolerate large time step sizes as long as such time step sizes are able to resolve the transient boundary conditions. While for problems that have scalar variables such as temperature closely linked to velocity, small time steps are necessary for obtaining accurate intermediate time-dependent solutions. This agrees with the findings about the PISO algorithm in literature [55], which concludes that for velocity-temperature strongly coupled problems, PISO gives correct solutions only when the time step is small.

6.3.3 Turbulent Flow Validation

The previous two chapters have validated the SLPISO algorithm through fine grid and laminar flow cases. As for turbulent flow, it is more complex and the CFD prediction is affected by many factors such as the turbulence model. The benchmark CFD data from Ghia [111] is used for such validation. Reynolds number of this validation is 5000, which is representative for typical indoor environment airflow.

Figure 81 shows the steady-state solution of FFD, SLPISO and Phoenics validated by benchmark data along the mid-width and mid-height of the geometry. No turbulence model is used in the CFD model.

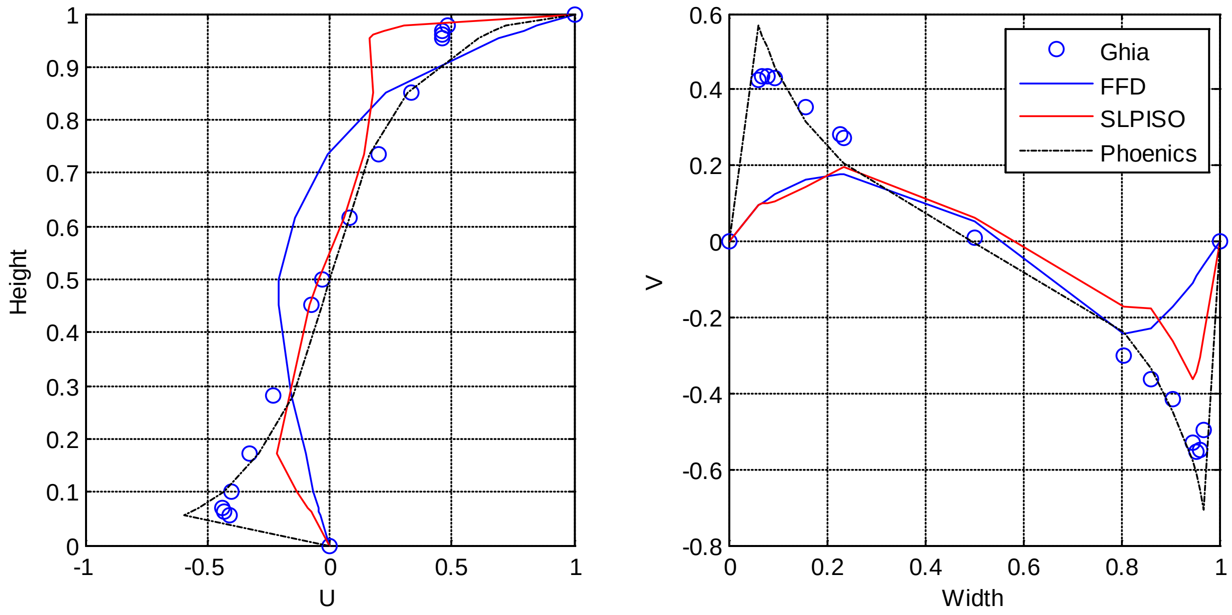


Figure 81. Comparison of the CFD solution from FFD, SLPISO and Phoenics on the lid-driven case with Re=5000

Figure 82 shows the normalized RMSE value of the three different algorithms against the benchmark data. Obviously SLPISO gives better steady-state solution than FFD method here, and Phoenics result which uses SIMPLEST algorithm gives best prediction over these three algorithms.

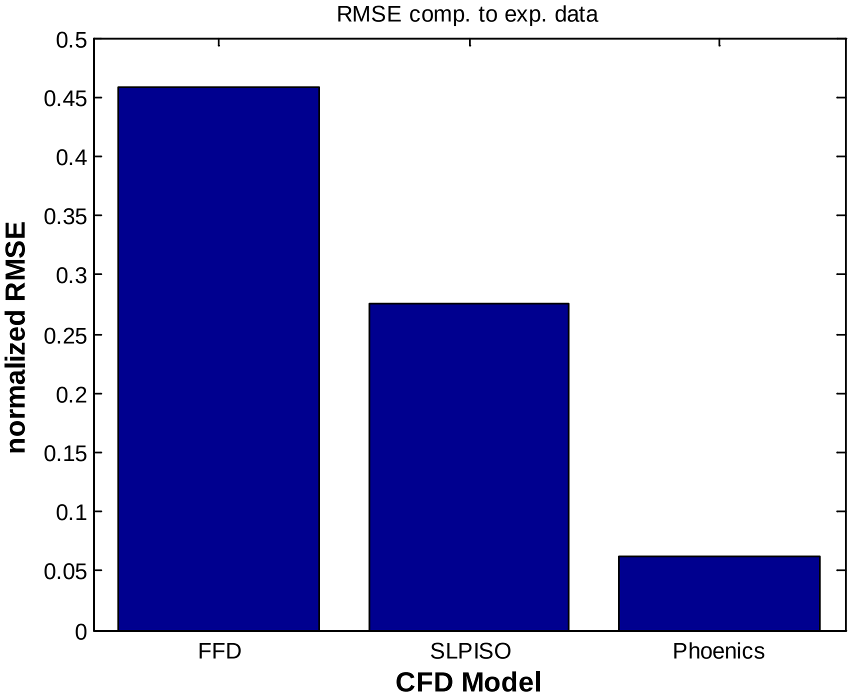


Figure 82. Comparison of the normalized RMSE value of FFD, SLPISO and Phoenics on the lid-driven case with Re=5000

The next validation utilizes the forced convection case as in chapter 5.2.1. Coarse grids together with constant viscosity models are used. Figure 83 shows the validation of the FFD and SLPISO predictions (steady-state) by both experimental data and grid independent results using RNG k-ε model with PHOENICS. The grid resolution used here for the FFD and SLPISO algorithm is 40×12 with constant viscosity 100ν. The time step used in this transient simulation is 1s. As a comparison, PHOENICS results with the same grid resolution and the constant viscosity is included. The results show good agreement of the solutions of two transient simulation algorithms with experimental data and PHOENICS results. Specifically for the location of X=2H, SLPISO and PHOENICS results indicate marginal improvement over FFD result.

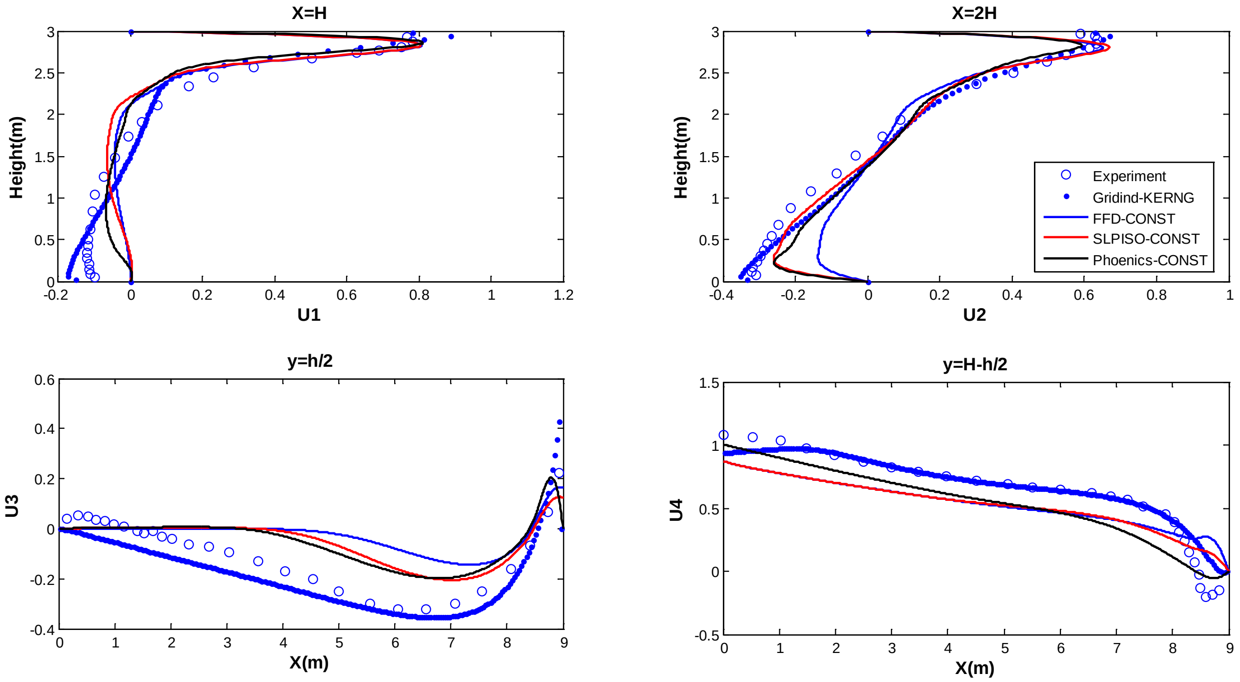


Figure 83. Validation of steady state solution of FFD and SLPISO on the FC case

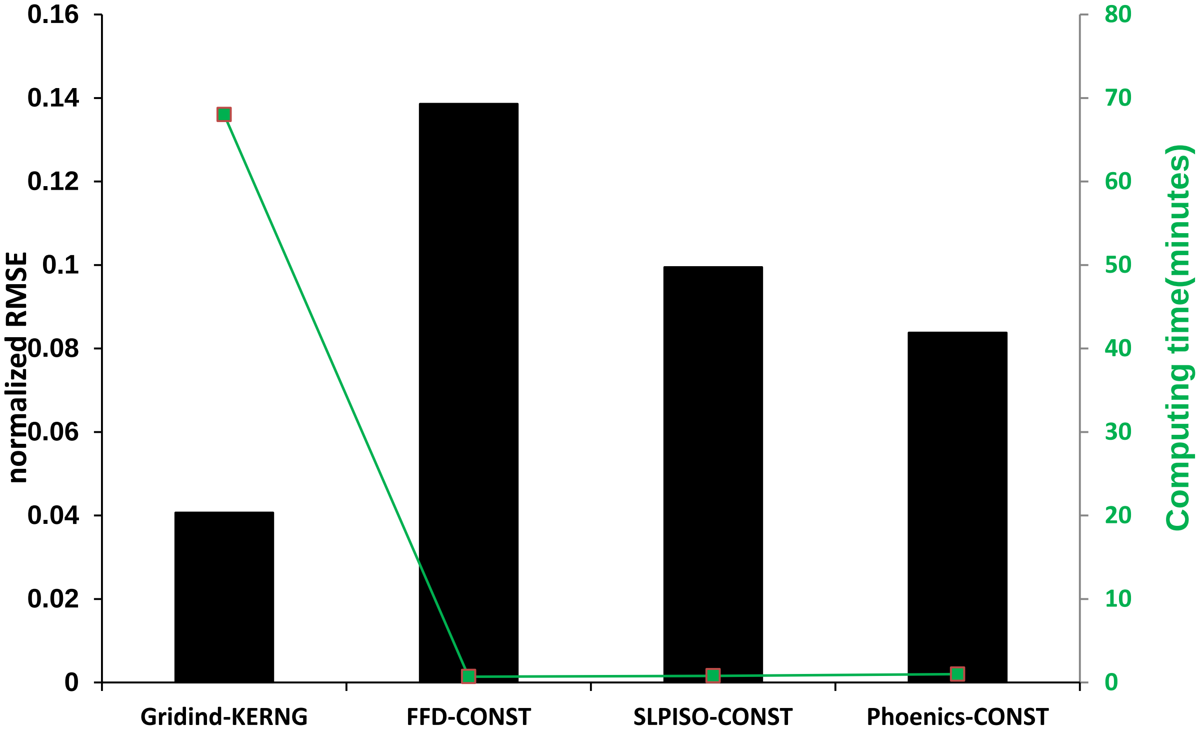


Figure 84. Normalized RMSE and computing time comparison of the tested CFD models on FC case

Figure 84 presents the performance of each CFD model on this FC case in terms of computing time and normalized RMSE indicated accuracy. Obviously grid-independent result using sophisticated RNG k-ε model gives the best CFD solution comparing to the other CFD models, but consumes much more computing resource. The other three coarse grid configurations, using FFD, SLPISO and Simplest (Phoenics) algorithm and constant viscosity model saves computing time dramatically with some sacrifice on the accuracy from the normalized RMSE value calibration of solutions against experimental data.

Figure 85, Figure 86 and Figure 87 present the solutions of velocity variation at different locations of the domain (position (3, 1.5), (4.5, 1.5) and (6, 1.5)). A general pattern of these comparisons illustrates that at the beginning period of the flow evolution, SLPISO provides a prediction closer to the grid independent result from PHOENICS than FFD for these three locations; as time elapses FFD has a better prediction than SLPISO, especially when steady-state is achieved. However, it is difficult to draw any conclusion from such comparison as many other factors such as grid size and distribution and turbulence model can all affect the CFD solution.

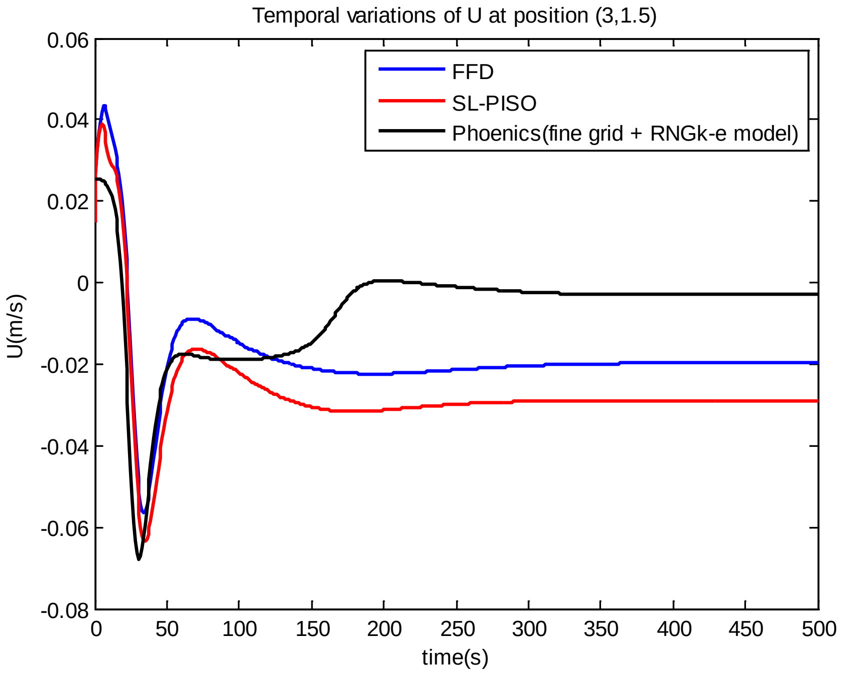


Figure 85. Temporal variation of U at position (3, 1.5) compared to grid independent result

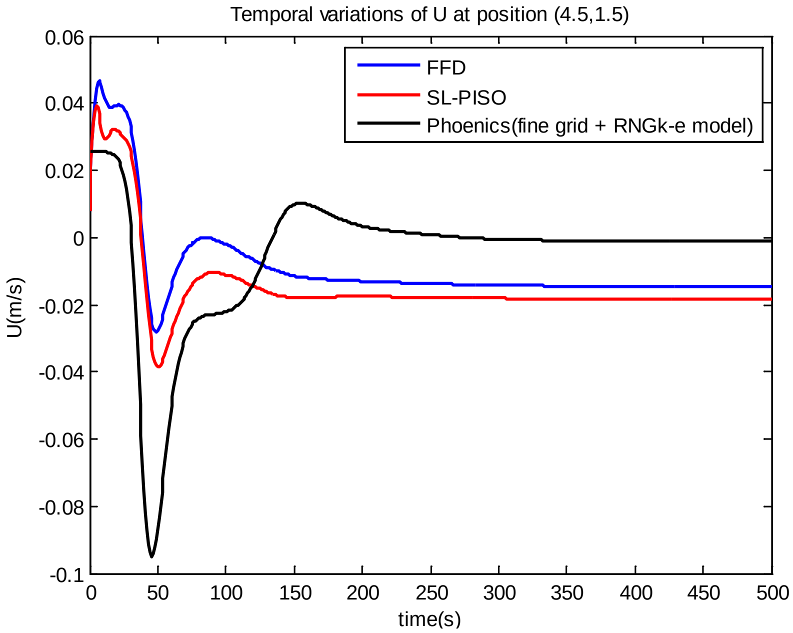


Figure 86. Temporal variation of U at position (4.5, 1.5) compared to grid independent result

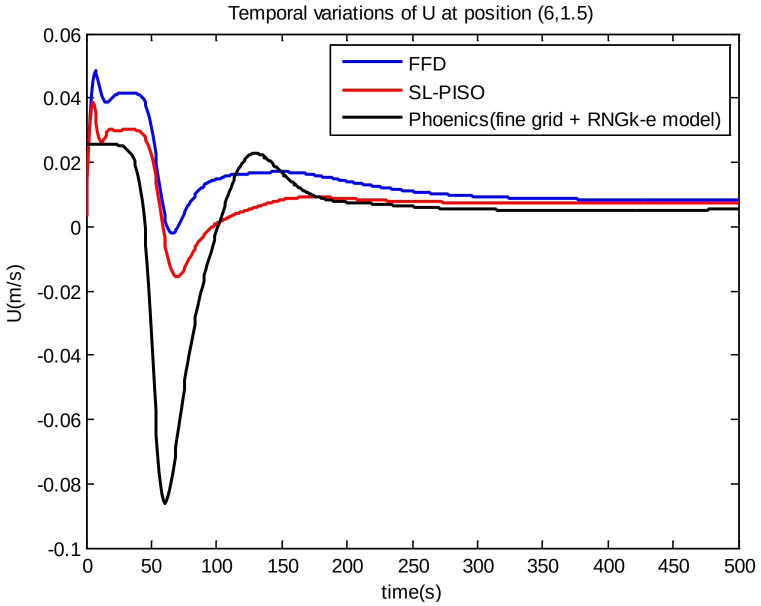


Figure 87. Temporal variation of U at position (6, 1.5) compared to grid independent result

Figure 88, Figure 89 and Figure 90 shows the transient velocity prediction comparison with the coarse grid PHOENICS result. They show similar patterns as previous comparison with grid independent results.

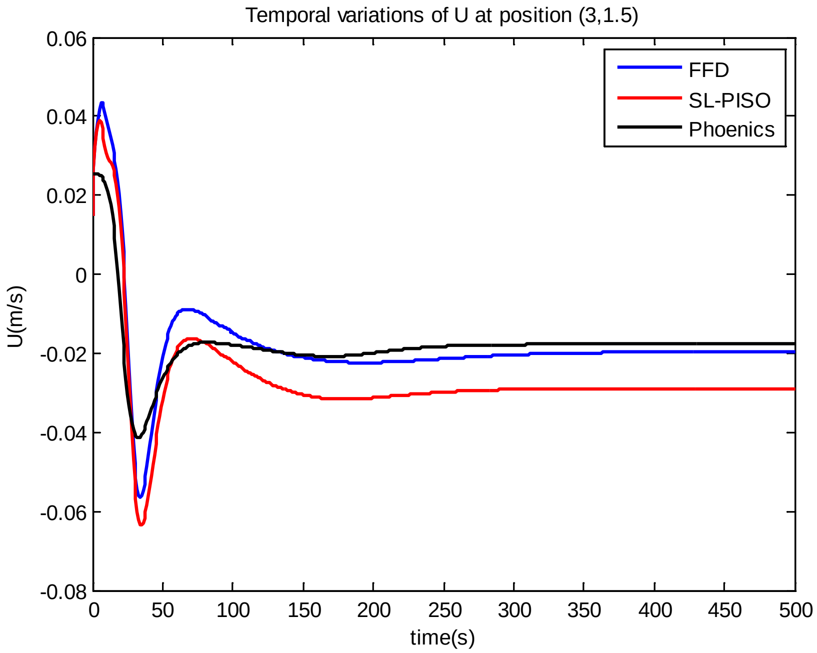


Figure 88. Temporal variation of U at position (3, 1.5) compared to coarse grid result; all three algorithms use coarse grid with constant turbulent viscosity model

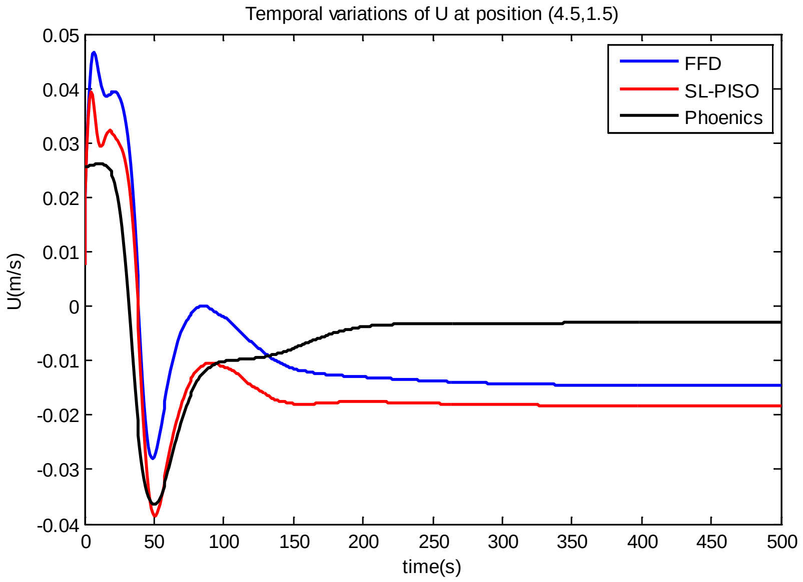


Figure 89. Temporal variation of U at position (4.5, 1.5) compared to coarse grid result; all three algorithms use coarse grid with constant turbulent viscosity model

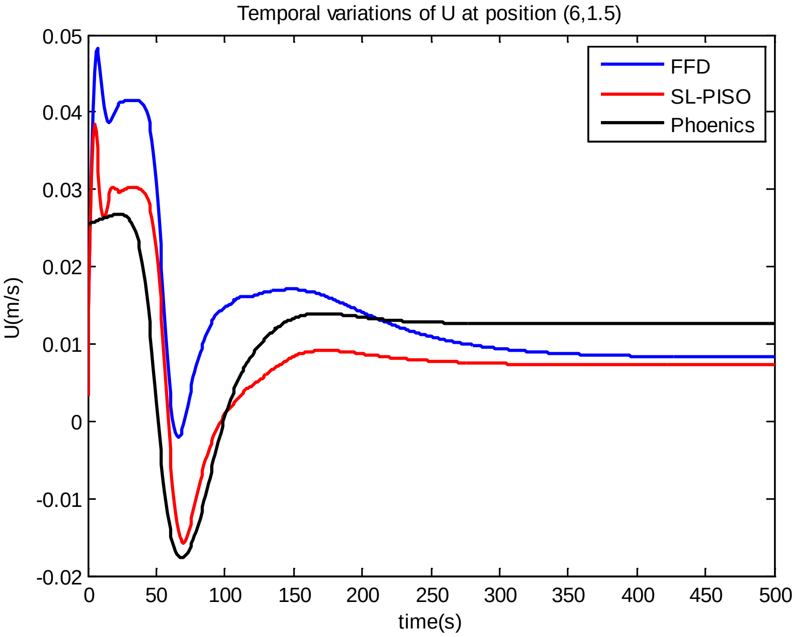


Figure 90. Temporal variation of U at position (6, 1.5) compared to coarse grid result; all three algorithms use coarse grid with constant turbulent viscosity model

The validation of the SLPISO algorithm using the turbulent flow case shows that such algorithms can provide good transient as well as steady-state solutions for turbulent flow simulations with acceptable accuracy. However, it is not conclusive from this comparison which of the two algorithms has better performance in terms of turbulent flow simulation.

6.4 Conclusions

Transient airflow and heat transfer phenomena are of great interest for IEQ study, but they draw much less attention than steady-state problems. Development of a CFD algorithm that is fast and accurate is of exceptional importance. SLPISO developed from this work shows great possibility of meeting such a requirement. This non-iterative algorithm, improved through integrating semi-Lagrangian advection into the PISO algorithm, can greatly decrease the computational intensity of PISO while keeping the accuracy of the PISO solution. As for problems in which temperature and velocity are closely linked to each other, it can only provide accurate transient solutions when the time step size is small.

CHAPTER 7
CONCLUSIVE SUMMARY AND FUTURE WORK

7.1 Conclusions

This study investigated fast CFD methods applicable to building IEQ study, identified the three most promising paths to reduce the computational intensity and applied them to typical indoor airflow validation. It is found that a coarse grid can work well with both validated sophisticated turbulence models and simplified turbulence models. A new semi-Lagrangian PISO algorithm brought out from this work improves the accuracy of FFD and the speed of PISO.

A normalized RMSE index is brought out to evaluate the overall closeness of solutions of different CFD models or solution of CFD model to measurement. A new method of grid independency analysis is brought out, using the magnitude comparison between numerical viscosity and turbulent viscosity, combined with normalized RMSE index. With total grid point number doubling, normalized RMSE value less than 10% can be considered to be an indication of grid independency. Magnitude of numerical viscosity in the computational domain overall should be much less than turbulent viscosity to claim grid independency. While doubling total grid number in CFD simulation is expensive and not always practical, viscosity magnitude comparison can be potentially integrated to commercial CFD software to inform users the status of grid independency.

7.1.1 Coarse Grid CFD Model

The coarse grid CFD model has been tested with both sophisticated turbulence models and simplified turbulence models. They both provide good CFD solutions for IEQ modeling problems with acceptable accuracy. The properly distributed coarse grid can minimize the discretization-induced truncation error and work well with sophisticated turbulence models, without significantly affecting the accuracy of the CFD solution.

Coarse grid CFD model work with sophisticated RANS based turbulence model such as RNG k-ε model. Local refinement with a grid size of 1% of the computational domain characteristic length (usually height of simulated room) is recommended at the normal direction of heated or cooled surface, as well as adjacent to inlet/outlet. Grid size around 10% of the characteristic length of the domain can be applied to the rest of directions and regions. This provides an optimized coarse grid distribution and resolution.

The grid number can be further reduced while discretization-induced error can be utilized to supplement turbulence viscosity. A constant viscosity model will work with such a coarse grid CFD model and provide a solution that is acceptable for the purpose of IEQ study. Generally grid size about 1/20 to 1/10 of the characteristic length of computational domain can be used, with a constant turbulent viscosity around 100 times the physical viscosity of air. The computing time can be reduced to as low as one percent of the grid-independent CFD model.

7.1.2 Transient Flow Simulation

A new SLPISO algorithm specifically for transient airflow simulation is brought out. The algorithm is implemented to a computer code and validated by benchmark data in the literature. This new algorithm adopts the speed advantage of FFD and accuracy advantage of the PISO algorithm and provides an alternative for the CFD velocity-pressure decoupling algorithm. Validations of such algorithms show some marginal improvement on FFD method. The problem inherited from the PISO algorithm makes the SLPISO algorithm not able to provide an accurate transient airflow prediction when temperature is closely linked to velocity.

7.2 Future Work

Great grid resolution, which provides much more data points in a flow field than necessary for analysis purposes, is the essential difference between the CFD method and other IEQ modeling methods. The CFD grid resolution directly relates to the number of the float point operation and hence restricts the computing speed. Therefore, investigation on the application of coarse grid provides great opportunity for cutting the computational intensity.

The results presented in this research provide a special path of using numerical viscosity analysis to explore the coarse grid possibilities. The numerical experimentation focusing on some specific representative cases provides some insights, but they are not solid enough to provide a universally applicable guideline for CFD users. More investigation especially needs to be done on the constant viscosity model.

The SLPISO brought out from this work can provide an accurate transient CFD solution as PISO and meanwhile cut the computational intensity significantly. However, it does not solve the problem that when temperature is closely linked to velocity, the time step shall not go large. An improved PISO for buoyancy-driven flow in the literature [56], which added a temperature predictor and corrector in the sequence, may provide some insight for future work.

APPENDICES

A.1 List of Experimental Data Used in This Dissertation

Table A 1: NC Data

Mid-height			Mid-width	
Distance(m)	V component(m/s)	Temperature(°C)	Height(m)	Temperature(°C)
0	0	50	0	24.212
0.000249998	0.0403	48.1	0.000999975	23.312
0.000500003	0.0611	47.2	0.002000025	23.012
0.00075	0.0843	46.8	0.003	22.712
0.000999975	0.1001	46.5	0.005000025	21.912
0.001250025	0.1158	45	0.006999975	21.312
0.0015	0.1325	44.74	0.009975	21.112
0.001749975	0.1457	43.492	0.015	20.948
0.002000025	0.1613	43.192	0.020025	21.16
0.00225	0.1655	42.284	0.027	21.256
0.002499975	0.1739	41.32	0.0375	21.624
0.002750025	0.1831	40.804	0.05625	22.344
0.003	0.2	40.54	0.075	23
0.005000025	0.2127	37.172	0.1125	24.284
0.006999975	0.2081	34.744	0.15	25.216
0.009975	0.1745	32.62	0.1875	26.004
0.015	0.1308	31.08	0.225	26.852
0.020025	0.0918	30.684	0.2625	27.58
0.024975	0.062	30.392	0.3	28.256
0.03	0.0374	30.18	0.3375	29.416
0.039975	0.013	30.596	0.375	30.696
0.050025	0.002871	30.7	0.4125	31.852
0.06	0.002147	31.252	0.45	32.648
0.080025	0.000263	31.5	0.4875	33.2
0.099975	0.0003728	31.6	0.525	34.272
0.12	0.000372	30.944	0.5625	34.936
0.15	0.000397	31.08	0.6	35.424
0.18	0.00331	31.004	0.6375	36.576
0.21	0.000279	31.108	0.675	37.3
0.24	0.000172	31.232	0.69375	37.964

0.27	0.000111	31.12	0.7125	38.728
0.3	0.000181	30.868	0.723	39
0.33	-0.000127	30.8	0.729975	39.2
0.375	-0.000069	30.696	0.735	39.5
0.42	-0.000487	30.752	0.740025	39.4
0.45	-0.000181	30.616	0.743025	39
0.48	-0.000606	30.66	0.744975	38.5
0.51	-0.000494	30.76	0.747	38
0.54	-0.000499	30.808	0.747975	37.4
0.57	-0.000458	30.64	0.749025	36.54
0.6	-0.000227	30.54	0.75	35.772
0.63	-0.000249	30.552		
0.650025	-0.000317	30.316		
0.669975	0.00126	30.304		
0.69	0.004408	30.244		
0.699975	0.002742	30.604		
0.710025	-0.0108	31.028		
0.72	-0.0403	30.792		
0.725025	-0.0684	31.252		
0.729975	-0.0916	30.796		
0.735	-0.1252	30.188		
0.740025	-0.1828	28.224		
0.743025	-0.2185	25.336		
0.744975	-0.2257	22.548		
0.747	-0.1884	19.736		
0.747225	-0.1803	19.272		
0.747525	-0.1736	18.544		
0.74775	-0.1604	18.2		
0.747975	-0.1488	17.46		
0.748275	-0.1268	16.432		
0.7485	-0.1056	15.296		
0.748725	-0.0835	14.732		
0.749025	-0.0588	13.74		
0.74925	-0.0314	13.108		
0.749475	-0.003389	12.628		
0.749775	-0.000412	11.236		
0.75	0	10		

Table A 2: FC Case data

X=H		**X=2H**		**Near floor**		**Near ceiling**	
Y(m)	**U(m/s)**	**Y(m)**	**U(m/s)**	**X(m)**	**U(m/s)**	**X(m)**	**U(m/s)**
0.040	-0.045	0.077	-0.141	0.152	0.019	0.012	0.492
0.090	-0.050	0.120	-0.147	0.359	0.024	0.536	0.485
0.143	-0.052	0.168	-0.144	0.534	0.022	1.021	0.471
0.206	-0.053	0.231	-0.139	0.676	0.017	1.474	0.444
0.274	-0.055	0.298	-0.135	0.829	0.014	1.987	0.417
0.345	-0.055	0.365	-0.130	0.989	0.008	2.504	0.394
0.417	-0.053	0.447	-0.126	1.163	0.004	3.004	0.375
0.503	-0.054	0.545	-0.121	1.398	-0.004	3.479	0.358
0.621	-0.051	0.672	-0.113	1.525	-0.008	3.975	0.341
0.839	-0.048	0.879	-0.097	1.678	-0.005	4.517	0.326
1.041	-0.045	1.075	-0.072	1.841	-0.014	5.058	0.315
1.252	-0.034	1.288	-0.039	2.031	-0.019	5.497	0.303
1.478	-0.020	1.505	-0.014	2.330	-0.028	6.007	0.293
1.728	-0.003	1.735	0.019	2.692	-0.033	6.558	0.281
1.912	0.014	1.936	0.041	3.035	-0.043	6.899	0.270
2.116	0.034	2.155	0.081	3.563	-0.060	7.205	0.257
2.341	0.074	2.364	0.137	4.038	-0.077	7.478	0.232
2.439	0.105	2.502	0.183	4.537	-0.092	7.875	0.206
2.560	0.156	2.629	0.226	5.005	-0.115	8.034	0.177
2.672	0.229	2.708	0.250	5.562	-0.137	8.108	0.142
2.735	0.286	2.799	0.280	6.060	-0.146	8.206	0.111
2.772	0.313	2.843	0.289	6.565	-0.146	8.300	0.067
2.811	0.343	2.893	0.289	7.078	-0.136	8.389	0.032
2.870	0.356	2.939	0.286	7.561	-0.115	8.477	-0.010
2.929	0.349	2.970	0.269	8.081	-0.074	8.499	-0.061
				8.411	-0.030	8.596	-0.091
				8.736	0.031	8.714	-0.086
				8.940	0.102	8.837	-0.067

Table A 3: MC Case data

Mid-height				Mid-width			
X(m)	V(m/s)	X(m)	T(°C)	Y(m)	U(m/s)	Y(m)	T(°C)
0	0	0	15	0	0	0	35.5
0.020094	0.223196	0.019878	18.99175	0.025402	-0.29499	0	35
0.059947	0.2414	0.039579	19.50738	0.016372	-0.2902	0.012451	20.03384
0.099976	0.226074	0.08004	19.49484	0.041289	-0.26251	0.030403	18.90867
0.200634	0.142362	0.229398	19.213	0.060634	-0.23591	0.080823	18.80528
0.350211	0.064061	0.520266	19.28836	0.101521	-0.22343	0.181143	19.06694
0.519765	-0.00279	0.810216	19.00728	0.200995	-0.1333	0.321168	19.23453

0.700816	-0.06068	0.960104	18.20497	0.401044	-0.02001	0.520939	19.29277
0.85113	-0.12409	0.999059	17.82167	0.500251	0.014551	0.721537	19.22428
0.951025	-0.20486	1.019416	17.62706	0.621381	0.044576	0.86159	19.28268
0.991325	-0.2635	1.04	15	0.820677	0.13916	0.960568	18.61069
1.020524	-0.26565			0.920925	0.207527	1.011203	17.68655
1.04	0			0.961637	0.241995	1.026783	17.28677
				1.002115	0.28503	1.039998	14.99862
				1.028847	0.303916	1.04	15
				1.04	0		

Table A 4: Natural Ventilation Case data

Height	P2		P3		P4		P5	
m	V(m/s)	T(°C)	V(m/s)	T(°C)	V(m/s)	T(°C)	V(m/s)	T(°C)
0.05	0	24.618	0	24.552	0	26.07	0	25.377
0.1	0.197036	25.245	0.149947	25.047	0.113925	25.905	0.215915	25.641
0.5	0.04991	27.258	0.065968	26.829	0.036022	26.895	0.048174	26.796
0.9	0.040796	27.588	0.042315	27.324	0.032984	27.588	0.03038	27.291
1.3	0.031682	28.479	0.022568	29.865	0.024955	29.568	0.059458	30.03
1.7	0.038843	29.733	0.04991	32.01	0.06293	31.845	0.103075	31.812
2.1	0.217	32.802	0.157108	32.637	0.059024	33	0.238049	33.627
2.15	0	33	0	33	0	33	0	33.99

Table A 5: Displacement Ventilation Case data

Location	Height[m]	0.05	0.1	0.6	1.1	1.5	1.9	2.3	2.38
P1: (0.78,1.83)	T[°C]	20.85	20.97	23.33	24.75	25.76	26.37	26.17	25.9
	V[m/s]		0.163	0.03	0.062	0.007	0.017	0.038	
P2: (1.74,1.83)	T[°C]	21.51	21.96	23.63	25.06	25.7	26.55	26.53	26.05
	V[m/s]		0.084	0.025	0.043	0.018	0.019	0.034	
P3: (2.70,1.83)	T[°C]	22.1	22.44	23.91	25.38	26.1	26.49	26.36	26.19
	V[m/s]		0.081	0.031	0.046	0.02	0.026	0.163	
P4: (3.66,1.83)	T[°C]	22.56	22.71	23.63	25.33	25.96	26.52	26.89	26.68
	V[m/s]		0.058	0.032	0.031	0.032	0.03	0.028	
P5 (4.62,1.83)	T[°C]	22.87	22.94	23.43	25.1	26.13	26.57	26.53	26.53
	V[m/s]		0.038	0.055	0.028	0.017	0.013	0.047	
P6: (2.70,0.61)	T[°C]	22.01	22.08	23.73	25.4	25.79	26.57	26.49	26.25
	V[m/s]		0.083	0.039	0.015	0.017	0.034	0.063	
P7: (2.70,1.22)	T[°C]	22.19	23.71	25.3	25.99	26.49	26.38	26.16	24.95
	V[m/s]		0.078	0.035	0.027	0.014	0.033	0.061	
P8: (2.7,2.44)	T[°C]	22.08	22.15	23.66	25.09	26.07	26.51	26.44	26.51
	V[m/s]		0.084	0.039	0.024	0.016	0.013	0.052	
P9: (2.7,3.05)	T[°C]	22.12	22.26	23.6	25.16	25.99	26.75	26.94	26.61
	V[m/s]		0.099	0.042	0.028	0.019	0.02	0.0116	

Table A 6: Steady-state lid-driven Case data (Re=100)

grid pt	y	U	x	V
129	1	1	1	0
126	0.9766	0.84123	0.9688	-0.05906
125	0.9688	0.78871	0.9609	-0.07391
124	0.9609	0.73722	0.9531	-0.08864
123	0.9531	0.68717	0.9453	-0.10313
110	0.8516	0.23151	0.9063	-0.16914
95	0.7344	0.00332	0.8594	-0.22445
80	0.6172	-0.13641	0.8047	-0.24533
65	0.5	-0.20581	0.5	0.05454
59	0.4531	-0.2109	0.2344	0.17527
31	0.2813	-0.15662	0.2266	0.17507
23	0.1719	-0.1015	0.1563	0.16077
14	0.1016	-0.06434	0.0938	0.12317
10	0.0703	-0.04775	0.0781	0.1089
9	0.0625	-0.04192	0.0703	0.10091
8	0.0547	-0.03717	0.0625	0.09233
1	0	0	0	0

Table A 7: Transient lid-driven Case data (Re=500)

t	U_bottom	t	U_center	t	U_up
0.0365	-6.00E-04	0.0157	-0.0019	0.0365	-6.00E-04
0.7505	-0.0159	0.7413	-0.0295	0.7231	-0.0565
1.3203	-6.00E-04	1.3428	-6.00E-04	1.343	-3.00E-04
1.7589	0.0098	1.7868	0.0178	1.7755	0.0345
2.244	0	2.2434	-0.0011	2.2217	2.00E-04
2.7792	-0.0135	2.7711	-0.0254	2.7586	-0.044
3.3026	0	3.302	-0.0011	3.3251	0
3.8094	0.0112	3.7924	0.0194	3.7849	0.0417
4.2491	8.00E-04	4.2705	-0.001	4.2711	0
4.7842	-0.0127	4.7764	-0.0243	4.7878	-0.0408
5.285	6.00E-04	5.2838	-0.0013	5.3068	-5.00E-04
5.7917	0.0118	5.7975	0.0203	5.7904	0.0431
6.2531	1.00E-04	6.2977	-7.00E-04	6.2755	-2.00E-04

6.8115	-0.0123	6.7814	-0.0235	6.7701	-0.0402
7.312	6.00E-04	7.2887	-7.00E-04	7.2893	4.00E-04
7.7739	0.0121	7.7798	0.0208	7.7949	0.0431
8.3034	0.0012	8.2572	-7.00E-04	8.3032	9.00E-04
8.8162	-0.012	8.7862	-0.0232	8.752	-0.0404
9.3167	9.00E-04	9.2934	-4.00E-04	9.2939	4.00E-04
9.801	0.0121	9.8073	0.0214	9.7769	0.0432
10.0248	0.01	10.0066	0.0164	10.0171	0.032

Table A 8: Transient Natural Convection Case data (Gr=2×10^6 and Pr=0.02)

t	0	0.019975	0.039975	0.059975	0.079975	0.099975	0.109673	0.119975	0.183154
U	0	-0.05624	-0.60255	-2.11319	-4.42884	-6.2836	-6.42712	-6.07431	3.74648
t	0.199975	0.219975	0.239975	0.25483	0.259975	0.299975	0.315282	0.359975	0.379975
U	2.71771	-0.4225	-3.23559	-3.63465	-3.39156	2.00153	2.437077	-1.75359	-1.09699
t	0.399975	0.416198	0.459975	0.479975	0.499975	0.505753	0.552072	0.559975	0.599306
U	0.66165	1.05663	-0.85866	-0.06791	0.76493	0.707166	-0.72144	-0.52104	0.662484
t	0.599975	0.645686	0.659975	0.679975	0.699975	0.735409	0.759975	0.778538	0.779975
U	0.63806	-0.57862	-0.03087	0.53739	0.17011	-0.41245	0.30958	0.338559	0.28192
t	0.799975	0.821217	0.859975	0.879975	0.899975	0.950026	0.959975	0.979975	0.999975
U	-0.21679	-0.29325	0.29533	-0.05921	-0.28388	0.178516	0.06368	-0.20965	-0.10662
t	1.031882	1.059975	1.079975	1.082474	1.099975	1.125435	1.159975	1.199975	1.211198
U	0.13356	-0.11759	-0.13425	-0.1465	0.0697	0.088877	-0.1239	0.10084	0.06745
t	1.219975	1.239975	1.25403	1.259975	1.296969	1.299975	1.339808	1.359975	1.379975
U	-0.00193	-0.10294	-0.09561	-0.02869	0.06946	0.02675	-0.07016	0.04095	0.03563
t	1.399975	1.459975	1.479975	1.499975	1.559975	1.579975	1.639975	1.659975	1.759975
U	-0.04144	0.03717	-0.01985	-0.0484	-0.00276	-0.03943	0.00412	-0.02758	-0.02304
t	1.779975	1.799975	1.899975	1.919975	1.979975	1.999975			
U	-0.00143	0.00715	-0.00613	-0.01799	-0.00355	-0.01464			

RELATED PUBLICATIONS

1. Haidong Wang, John Zhai and Xiang Liu. 2012. Feasibility of Utilizing Numerical Viscosity from Coarse Grid CFD for Fast Turbulence Modeling of Indoor Environments. *Building Simulation*. Accepted.
2. Haidong Wang and John Zhai. 2012. Application of Coarse Grid CFD on Indoor Environment Modeling: Optimizing the Trade-off between Grid Resolution and Simulation Accuracy. *HVAC&R Research*. 18(5): 915-933.
3. Haidong Wang and John Zhai. 2012. Analyzing Grid-Independency and Numerical Viscosity of Computational Fluid Dynamics for Indoor Environment Applications. *Building and Environment*. 52: 107-118.
4. Haidong Wang, John Zhai and Xiang Liu. 2012. Simplify turbulence modeling in indoor environment study by using coarse grid. *The Second International Conference of Building Energy and Environment*. Boulder, Colorado. August 1-4.
5. John Zhai and Haidong Wang. 2011. Optimizing the Trade-off between Grid Resolution and Simulation Accuracy: Coarse Grid CFD Modeling. *Final Report of ASHRAE Research Project 1418*. American Society of Heating, Refrigerating and Air-conditioning Engineers, Atlanta, GA.

REFERENCES

[1] Versteeg HK, Malalasekera W. An introduction to computational fluid dynamics: the finite volume method. 2007: Prentice Hall.

[2] Nielsen PV. Flow in Air Conditioned Rooms (English Translation, 1976). 1974, Technical University of Denmark. 121.

[3] Hensen J, Djunaedy E, Trcka M, Yahiaoui A. Building performance simulation for better design: some issues and solutions. 2004, Technische Universiteit Eindhoven: Einhoven.

[4] Augenbroe G, Hensen J. Simulation for better building design. Building and Environment 2004; 39(8): 875-8.

[5] Sun H, Stowell R, Keener H, Michel F. Two-dimensional computational fluid dynamics(CFD) modeling of air velocity and ammonia distribution in a high-rise (tm) hog building. Transactions of the ASAE 2002; 45(5): 1559-68.

[6] Gousseau P, Blocken B, Stathopoulos T, Van Heijst G. CFD simulation of near-field pollutant dispersion on a high-resolution grid: a case study by LES and RANS for a building group in downtown Montreal. Atmospheric Environment 2011; 45(2): 428-38.

[7] Zerihun Desta T, Janssens K, Van Brecht A, Meyers J, Baelmans M, Berckmans D. CFD for model-based controller development. Building and Environment 2004; 39(6): 621-33.

[8] Zhai Z, Chen Q. Performance of coupled building energy and CFD simulations. Energy and Buildings 2005; 37(4): 333-44.

[9] Djunaedy E. External coupling between building energy simulation and computational fluid dynamics. 2005, Technische Universiteit Eindhoven: Eindhoven, the Netherlands.

[10] Wang L, Chen Q. Validation of a Coupled Multizone-CFD Program for Building Airflow and Contaminant Transport Simulations. HVAC&R Research 2007; 13(2): 267-81.

[11] DesignBuilder. Design Builder Simulation + CFD Training Guide.

[12] A-Tas. Manual of A-Tas: Theory.

[13] Wang LL, Dols WS, Chen Q. Using CFD Capabilities of CONTAM 3.0 for Simulating Airflow and Contaminant Transport in and around Buildings. HVAC&R Research 2010; 16(6): 749-63.

[14] Cohen J, Molemake J. A Fast Double Precision CFD Code Using CUDA. in 21st International Conference on Parallel Computational Fluid Dynamics (ParCFD2009) 2009.

[15] CHAM. PHOENICS Documentation. 2011, CHAM L.T.D.

[16] Chen GY, Xu WR. A zero-equation turbulence model for indoor airflow simulation. Energy and Buildings 1998; 28(2): 137-44.

[17] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. Journal of Computational Physics 1986; 62(1): 40-65.

[18] Chorin AJ. A numerical method for solving incompressible viscous flow problems. Journal of Computational Physics 1967; 2(1): 12-26.

[19] Andrews, Loezos PN, Sundaresan S. Coarse-Grid Simulation of Gas-Particle Flows in Vertical Risers. Industrial & Engineering Chemistry Research 2005; 44(16): 6022-37.

[20] Mora L, Gadgil AJ, Wurtz E. Comparing zonal and CFD model predictions of isothermal indoor airflows to experimental data. Indoor Air 2003; 13(2): 77-85.

[21] Feustel HE. COMIS — An International Multizone Air-Flow and Contaminant Transport Model. 1998, Lawrence Berkeley National Laboratory.

[22] Megri ACr, Haghighat F. Zonal Modeling for Simulating Indoor Environment of Buildings: Review, Recent Developments, and Applications. HVAC&R Research 2007; 13(6): 887-905.

[23] Burkardt J, Gunzburger M, Lee HC. POD and CVT-based reduced-order modeling of Navier-Stokes flows. Computer methods in applied mechanics and engineering 2006; 196(1): 337-55.

[24] Emmerich SJ. Validation of Multizone IAQ Modeling of Residential-Scale Buildings: A Review. 2001.

[25] Axley J. Multizone Airflow Modeling in Buildings: History and Theory. HVAC&R Research 2007; 13(6): 907-28.

[26] Dols WS. Tool for Modeling Airflow and Contaminant Transport. ASHRAE Journal 2001; 43(3): 35-42.

[27] Feustel H, Rayner-Hooson A. COMIS Fundamentals. Lawrence Berkeley Laboratory. 1990.

[28] Walton GN. A computer program for building airflow network modeling. NISTIR 89-4072 1989.

[29] Walton GN, Dols WS. CONTAM user guide and program documentation. NUSTIR 7251 2005.

[30] Blomsterberg A, Johansson T. Use of Multi-Zone Air Flow Simulations to Evaluate a Hybrid Ventilation System, in 9th International IBPSA Conference. 2005: Montreal, Canada.

[31] Li Y, Duan S, Yu ITS, Wong TW. Multi-zone modeling of probable SARS virus transmission by airflow between flats in Block E, Amoy Gardens. Indoor Air 2005; 15(2): 96-111.

[32] Liu X, Zhai ZJ. Prompt tracking of indoor airborne contaminant source location with probability-based inverse multi-zone modeling. Building and Environment 2009; 44(6): 1135-43.

[33] Patankar S. Numerical heat transfer and fluid flow. 1980: Taylor & Francis.

[34] ORSZAGT SA. Analytical theories of turbulence. J. Fluid Mech 1970; 41(part 2): 363-86.

[35] Smagorinsky J. General circulation experiments with the primitive equations. Monthly weather review 1963; 91(3): 99-164.

[36] Reynolds O. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. Philosophical Transactions of the Royal Society of London. A 1895; 186: 123-64.

[37] Spalart P, Jou W, Strelets M, Allmaras S. Comments of feasibility of LES for wings, and on a hybrid {RANS/LES} approach. 1997.

[38] Wilcox DC. Turbulence Modeling for CFD. 1993: DCW Industries.

[39] Schmitt FG. About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity. Comptes Rendus Mécanique 2007; 335(9-10): 617-27.

[40] Zhai ZJ, Zhang Z, Zhang W, Chen QY. Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: Part 1: Summary of prevalent turbulence models. HVAC&R Research 2007; 13(6): 853-70.

[41] Gurniki F, Bark FH, Zahrai S. Turbulent free convection in large electrochemical cells with a binary electrolyte. Journal of Applied Electrochemistry 1999; 29(1): 27-34.

[42] Xu W, Chen Q. Numerical simulation of airflow in a room with differentially heated vertical walls. in ASHRAE Transactions 1998.

[43] Yakhot V, Orszag SA. Renormalization-Group Analysis of Turbulence. Physical Review Letters 1986; 57(14): 1722.

[44] Zhang Z, Zhang W, Zhai ZJ, Chen QY. Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: Part 2: Comparison with experimental data from literature. HVAC&R Research 2007; 13(6): 871-86.

[45] Prandtl L. Uber die ausgebildete turbulenz. . ZAMM 1925; 5: 136-9.

[46] Chen Q, Xu W. A zero-equation turbulence model for indoor airflow simulation. Energy and Buildings 1998; 28(2): 137-44.

[47] Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. International Journal of Heat and Mass Transfer 1972; 15: 1787-806.

[48] Patankar SV. Numerical Heat Transfer and Fluid Flow. 1980: Hemisphere Publishing Corporation, Taylor & Francis Group, New York.

[49] Van Doormaal JP, Raithby GD. Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numerical Heat Transfer 1984; 7(2): 147-63.

[50] Acharya S, Moukalled FH. IMPROVEMENTS TO INCOMPRESSIBLE FLOW CALCULATION ON A NONSTAGGERED CURVILINEAR GRID. Numerical Heat Transfer, Part B: Fundamentals 1989; 15(2): 131-52.

[51] Dormal JPV, Raithby GD. An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows, in National Heat Transfer Conference. 1985: Denver, Colorado.

[52] Harlow FH, Welsh JE. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. The Physics of Fluids 1965; 8, N. 10: 2182-9.

[53] Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations. Journal of computational physics 1985; 59(2): 308-23.

[54] Jang DS, Jetli R, Acharya S. COMPARISON OF THE PISO, SIMPLER, AND SIMPLEC ALGORITHMS FOR THE TREATMENT OF THE PRESSURE-VELOCITY COUPLING IN STEADY FLOW PROBLEMS. Numerical Heat Transfer 1986; 10(3): 209-28.

[55] Audi E. Comparison of pressure-velocity coupling schemes for 2D flow problems. in Advances in Computational Tools for Engineering Applications, 2009. ACTEA '09. International Conference on 2009.

[56] Oliveira P, Issa R. An Improved PISO Algorithm for the Computation of Buoyancy-Driven Flows. Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology 2001; 40(6): 473-93.

[57] Foster N, Metaxas D. Realistic animation of liquids. Graph. Models Image Process. 1996; 58(5): 471-83.

[58] Foster N, Metaxas D. Modeling the motion of a hot, turbulent gas, in Proceedings of the 24th annual conference on Computer graphics and interactive techniques. 1997, ACM Press/Addison-Wesley Publishing Co.

[59] Stam J. Stable fluids, in Proceedings of the 26th annual conference on Computer graphics and interactive techniques. 1999, ACM Press/Addison-Wesley Publishing Co.

[60] Zuo W, Chen Q. Real-time or faster-than-real-time simulation of airflow in buildings. Indoor Air 2009; 19(1): 33-44.

[61] Lucia DJ, Beran PS, Silva WA. Reduced-order modeling: new approaches for computational physics. Progress in Aerospace Sciences 2004; 40(1-2): 51-117.

[62] Bui-Thanh T, Willcox K, Ghattas O. Parametric Reduced-Order Models for Probabilistic Analysis of Unsteady Aerodynamic Applications. Aiaa Journal 2008; 46(10): 2520-9.

[63] Borggaard J, Hay A, Pelletier D. Interval-based reduced-order models for unsteady fluid flow. International Journal of Numerical Analysis and Modeling 2007; 4(3-4): 353-67.

[64] Antoulas AC. Approximation of Large-Scale Dynamical Systems, in Advances in Design and Control. 2005: Philadelphia.

[65] Surana A. Reduced order modeling for contaminant transport and mixing in building systems-A case study using dynamical systems techniques, in American Control Conference. 2008: East Hartford. 902-7.

[66] Hou Y. Reduced-order modeling of incompressible jet flow using proper orthogonal decomposition and galerkin projection, in Department of Electrical and Computer Engineering. 2010, Northwesten University.

[67] D RJ. Reduced-order modeling of multiscale turbulent convection Application to data center thermal management, in Mechanical Engineering. 2006, Georgia Institute of Technology.

[68] Schetzen M. The Volterra and Wiener theories of nonlinear systems. 1980, New York: Wiley.

[69] Hall KC TJ, Clark WS. Computation of unsteady nonlinear flows in cascades using a harmonic balance technique, in Ninth International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity ofTurbomachines and Propellers. 2000: Lyon, France.

[70] Farhat DAaC. Interpolation Method for the Adaptation of Reduced-Order Models to Parameter Changes and Its Application to Aeroelasticity. Aiaa Journal 2008; 46(7): 1803-13.

[71] Borggaard J. Model reduction for indoor-air behavior in control design for energy-efficient buildings, in American Control Conference. 2012: Blacksburg, VA. 2283-8.

[72] Barnard S, Biswas R, Saini S, Van der Wijngaart R, Yarrow M, Zechtzer L, Foster I, Larsson O. Large-scale distributed computational fluid dynamics on the information power grid using globus. in Frontiers of Massively Parallel Computation, 1999. Frontiers' 99. The Seventh Symposium on the 1999: IEEE.

[73] Wang H, Zhai ZJ. Analyzing grid-independency and numerical viscosity of computational fluid dynamics for indoor environment applications. Building and Environment 2012.

[74] Weathers JW. A comparative study of room airflow: Numerical prediction using computational fluid dynamics and full-scale experimental measurements. Ashrae transactions 1993; 99: 144-57.

[75] Moulton M, Steinhoff J. A technique for the simulation of stall with coarse-grid CFD methods. in AIAA Paper2000-0277, 38th AIAA Aerospace Science Meeting 2000.

[76] Andrews IV AT, Loezos PN, Sundaresan S. Coarse-grid simulation of gas-particle flows in vertical risers. Industrial & engineering chemistry research 2005; 44(16): 6022-37.

[77] Nielsen PV. The selection of turbulence models for prediction of room airflow. TRANSACTIONS-AMERICAN SOCIETY OF HEATING REFRIGERATING AND AIR CONDITIONING ENGINEERS 1998; 104: 1119-27.

[78] Nilsson HO. Thermal comfort evaluation with virtual manikin methods. Building and Environment 2007; 42(12): 4000-5.

[79] Courant R, Isaacson E, Rees M. On the solution of nonlinear hyperbolic differential equations by finite differences. Communications on Pure and Applied Mathematics 1952; 5(3): 243-55.

[80] Courant R, Friedrichs K, Lewy H. Ãœber die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen 1928; 100(1): 32-74.

[81] Tu J, Yeoh GH, Liu C. Computational fluid dynamics: a practical approach. 2008: Butterworth Heinemann.

[82] Einstein A. The foundation of the general theory of relativity. Annalen Phys 1916; 49(769-822): 31.

[83] Goethals K, Janssens A. Sensitivity analysis of predicted convective heat transfer at internal building surfaces to diffuser modelling in CFD. in 11th International Building Performance Simulation Association Conference (Building Simulation 2009) 2009. Glasgow, Scotland, UK.

[84] CFD-online. http://www.cfd-online.com/Wiki/Introduction_to_turbulence/Reynolds_averaged_equations.

[85] Chen Q, Glicksman L, Srebric J. Simplified methodology to factor room air movement and the impact on thermal comfort into design of radiative, convective and hybrid heating and cooling systems, in ASHRAE RP-927. 1999. 184 pages.

[86] Versteeg HK, Malalasekera W. An introduction to computational fluid dynamics: the finite volume method 1995: McGraw-Hill, Loughborough University. 257 pages.

[87] Zhai ZJ, Zhang W, Zhang Z, Chen QY. Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: part 1 - Summary of prevalent turbulence models. HVAC&R Research 2007; 13(6): 853-70.

[88] Zhang Z, Zhai ZJ, Zhang W, Chen QY. Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: Part 2-comparison with experimental data from literature. HVAC&R Research 2007; 13(6): 871-86.

[89] Chen Q. Comparison of different k-e models for indoor air flow computations. Numerical Heat Transfer, Part B: Fundamentals 1995; 28(3): 353-69.

[90] Roache PJ. Fundamentals of Computational Fluid Dynamics. 1998: Hermosa publisher.

[91] Runchal AK, Anand MS, Mongia HC. An unconditionally-stable central differencing scheme for high Reynolds number flows, in AIAA, Aerospace Sciences Meeting. 1987: Reno, NV. 13.

[92] Leonard BP. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. Computer Methods in Applied Mechanics and Engineering 1979; 19(1): 59-98.

[93] Darwish FMM. A UNIFIED FORMULATION OF THE SEGREGATED CLASS OF ALGORITHMS FOR FLUID FLOW AT ALL SPEEDS. Numerical Heat Transfer, Part B: Fundamentals 2000; 37(1): 103-39.

[94] Anderson VDA, Tunnehill JC, Pletcher RH. Computational fluid mechanics and heat transfer. 1984: Hemisphere Publishing Corp., Washington.

[95] Stam J. A simple fluid solver based on the FFT. Journal of Graphics Tools 2002; 6(2): 43-52.

[96] Stam J. Real-time fluid dynamics for games. 2003: Proceedings of Game Developer Conference.

[97] Zuo W. Advanced simulations of air distributions in buildings. 2010, Purdue University: West Lafayette.

[98] Roache PJ. Quantification of uncertainty in computational fluid dynamics. Annual Review of Fluid Mechanics 1997; 29: 123-60.

[99] Celebi ME, Celiker F, Kingravi HA. On Euclidean norm approximations. Pattern Recognition; 44(2): 278-83.

[100] Wang B, Zhao B, Chen C. A simplified methodology for the prediction of mean air velocity and particle concentration in isolation rooms with downward ventilation systems. Building and Environment 2010; 45(8): 1847-53.

[101] Celik IB, Ghia U, Roache PJ, Freitas CJ, Coleman H, Raad PE. Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications. Journal of Fluids Engineering 2008; 130(7): 078001-4.

[102] Zhai ZJ, Zhang W, Zhang Z, Chen Q. Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: part 1 - Summary of prevalent turbulence models. HVAC&R Research 2007; 13(6): 853-70.

[103] Zhang Z, Zhai ZJ, Zhang W, Chen Q. Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: Part 2 - Comparison with experimental data from literature. HVAC&R Research 2007; 13(6): 871-86.

[104] Ampofo F, Karayiannis TG. Experimental benchmark data for turbulent natural convection in an air filled square cavity. International Journal of Heat and Mass Transfer 2003; 46(19): 3551-72.

[105] Nielsen PV. Specification of a two-dimensional test case. International Energy Agency 1990.

[106] Blay D, Mergui S, Niculae C. Confined turbulent mixed convection in the presence of a horizontal buoyant wall jet. Fundamentals of Mixed Convection, HTD 1992; 213: 65-72.

[107] Yuan X, Chen Q, Glicksman L, Hu S, Hu Y, Yang X. ASHRAE Research Report: Performance evaluation and development of design guidelines for displacement ventilation. 1999, Ashrae. 230.

[108] Jiang Y, Chen Q. Buoyancy-driven single-sided natural ventilation in buildings with large openings. International Journal of Heat and Mass Transfer 2003; 46(6): 973-88.

[109] Zhai Z, Wang H. Optimizing the Trade-off Between Grid Resolution and Simulation Accuracy: Coarse Grid CFD Modeling. 2011, American Society of Heating, Refrigerating and Air-Conditioning Engineers: Atlanta. 147.

[110] Zuo W, Jin M, Chen Q. Reduction of numerical diffusion in the FFD model. Engineering Applications of Computational Fluid Mechanics 2012; 6(2): 234-47.

[111] Ghia U, Ghia K, Shin C. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of computational physics 1982; 48(3): 387-411.

[112] Churchfield MJ. An Overview of OpenFOAM.

[113] Mohamad A. Benchmark solution for unsteady state CFD problems. Numerical Heat Transfer, Part A Applications 1998; 34(6): 653-72.