A Computational Fluid Dynamics Analysis of Shock Wave-Boundary Layer Interactions

by

Jiayue Hu

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 4, 2018

Keywords: Computational Fluid Dynamics, SWBLI, Turbulence Closure,
Supersonic Aerodynamics

Approved by

D. Stephen Nichols, Assistant Professor of Aerospace Engineering
Brian S. Thurow, Department Chair, Professor of Aerospace Engineering
David E. Scarborough, Assistant Professor of Aerospace Engineering

Abstract

With the continuous development of Computational Fluid Dynamics (CFD), many different numerical methods and turbulence models have been modified to enhance the capability of prediction of complex turbulence flow. This study investigates the characteristics of the shock wave-boundary layer interactions (SWBLI) generated by the fin with an angle of attack of 15°in a Mach 2 flow. Reynolds-averaged Navier-Stokes equations (RANS) are employed in this research with two different turbulence closures, which are Wilcox $k-\omega$ Model and Blended $k-\varepsilon, k-\omega$ Turbulence Model, using TENASI as the CFD solver. The simulation results are compared with the data from wind tunnel experiments using the Plenoptic PIV method so that the performance and limitations of different turbulence models could be estimated. The simulation of the fin tested in the same flow condition but in a non-obstacle surrounding is also introduced to show the difference of the supersonic flow field between the in-tunnel case and the free stream case.

Dedication

To my love, Xinlian.

Acknowledgements

I could not have reached the point in which I am today without the help and guidance of so many people. First and foremost, my deepest gratitude goes to my supervisor, Dr. Stephen Nichols, who offered me the opportunity to complete my master's thesis in the Department of Aerospace Engineering at Auburn University. Dr. Nichols significantly inspired me and dedicated himself to enlighten and teach me. Such a painstaking guidance made me learn how to simulate supersonic flow using CFD. At the beginning of the research, I felt anxious about the uncontrolled experiments and almost lost my bearings. It is Dr. Nichols the one who was always there to encourage me. He told me that uncertainty is common in research, and that experiments rarely obtain good results in the first tries, especially in the CFD field.

I would also like to express my gratefulness to two collaborators of the CFD laboratory, Michael Siemon and Daniel Stubbs, who helped me so much with my research.

Besides, I owe my deepest gratitude to my girlfriend Xinlian Ye, who has stood by me and brought me happiness all the time.

A special thank goes to my parents. Words cannot express how grateful I am for the sacrifices you have made for me. Education is the ultimate wealth parents can give to their children. You have so lovingly walked this journey with me. Thank you very much.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AMN	Arbitrary Mach Number
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrich-Lewy
DDES	Delayed Detached Eddy Simulation
DES	Detached Eddy Simulation
D/LES	Detached/Large Eddy Simulations
DNS	Direct Numerical Simulation
ENO	Essentially Non-Oscillatory
GR	Growth Rate
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
HLLC	Harten-Lax-van-Leer Contact
LES	Large Eddy Simulation
MOC	Method of Characteristics

MUSCL	Monotone Upstream Conservative Limited
PIV	Particle Image Velocimetry
RANS	Reynolds-averaged Navier Stokes
RSM	Reynolds Stress Model
SAS	Scale Adaptive Simulation
SGS	Subgrid Scale
SST	Shear Stress Transport
SWBLI	Shock-Wave/Boundary-Layer Interaction
WENO	Weighted Essentially Non-Oscillatory

List of Symbols

A	Cross-Sectional Area
A^*	Nozzle Throat Area
$\vec{A}_i$	Cell Face Normal Vector
a_r	Reference Speed of Sound
a_t	Grid Speed
C_f	Skin Friction Coefficient
C_{lim}	Stress-Limiter Strength
C_p	Specific Heat Capacity at Constant Pressure
C_v	Specific Heat Capacity at Constant Volume
d_v	Distance to the Viscous Surface
E	Energy
E_c	Eckardt Number
$\vec{F}^c$	Convective Fluxes Vectors
$\vec{F}^v$	Viscous Flux Vectors
$F(\varphi)$	Blending Function
$\vec{f}_i$	Vectors from Cell Centroid to Face Centroid

Gr	Grashof number
h	Specific Enthalpy
h_r	Reference Enthalpy
k	Turbulence Kinetic Energy
L_r	Characteristic Length Scale
l_{sep}	Primary Separation Distance
$l_0^{\,i}$	Left Eigenvectors
M	Jacobian Matrix
M_r	Reference Mach Number
$\hat{\vec{n}}$	External Pointing Unit Normal Vector
P	Static Pressure
Pr	Prandtl Numbers
Pr_d	Dynamic Prandtl Numbers
Pr_t	Turbulent Prandtl Numbers
Q	State Variables Vector
R	Specific Gas Constant
Re	Reynolds Number
S	General Source Term
S_{ij}	Conventional Mean Strain-Rate Tensor
S_L	Smallest Speed Wave

S_R	Largest Speed Wave
S_*	Contact Wave Speed
T	Temperature
t_{ij}	Molecular Stress Tensor
$\vec{U}_{i\pm1/2}$	Reconstructed Conserved Variables
U_{fric}	Friction Velocity
U_r	Reference Velocity
u	Velocity Components in x Axis
$\vec{V}_s$	Velocity of Control Volume Surface
$\mathcal{V}$	Control Volume
v	Velocity Components in y Axis
W	Characteristic Variable Vector
w	Velocity Components in z Axis
X	Right Eigenvector Matrix of A
Λ	Eigenvalue Matrix
Γ	Preconditioning Matrix
Θ	Velocity Normal to Control Volume Surface
ψ_K	Calculative Limiters
Ω	Boundary Conditions
Ω_{ij}	Rotational Strain Rate Tensor

β^*	Turbulence-Model Closure Coefficient
ρ	Density
ρ_r	Reference Density
λ_i	Eigenvalues
μ	Molecular Viscosity
μ_r	Reference Viscosity
μ_t	Eddy Viscosity
μ_{tot}	Total viscosity
ε	Tiny Perturbation
ω	Turbulence Dissipation
τ_{ij}	Reynolds Stress Tensor
τ_{wall}	Skin Shear Stress
γ	Particular Heat Ratio
ν	Local Kinematic Viscosity
Δ_{max}	Local Maximum Space
Δs	Spacing
Δt	Time Step
Δy	Initial Spacing
δ_{ij}	Kronecker Delta Function

Chapter 1

Introduction

1.1 Purpose of the Study

Shock wave-boundary layer interactions (SWBLI) are one of the most significant flow phenomena to be regarded in the design of supersonic and hypersonic aircraft. These interactions ubiquitously exist in the flowfield around the aircraft, such as wing-body junctions, engine inlet corners and deflected control surfaces, and can extremely change the performance of the vehicle component. In some cases, the increase in acoustic and thermal loads caused by SWBLI might lead to the failure of structures in a short time [1]. To overcome the detrimental effect of SWBLI, one easy solution without fully understanding the interaction is to strengthen the structure of components by using more materials and put a thermal protection coating on the surface. However, this method is hard to accomplish due to both cost and weight. Therefore, it is important for the aerospace industry to discover the underlying physics and influence factors of shock wave-boundary layer interactions so that the deleterious effects owing to interactions could be minimized. There is extensive research about SWBLI, which can be divided into laminar, transitional, and turbulent interactions according to the state of the incoming boundary. The overwhelming majority of SWBLI research is focused on either laminar or turbulent interactions for various geometry. In

the previous studies, laminar interactions have been revealed to be relatively steady, and numerical simulations have been employed to indicate the flow behaviors, such as thermal and acoustic effects very well. The research about laminar interactions is concentrated on non-equilibrium and reacting flows [2]. Since turbulence is unsteady, and people still have no consensus about the reason of this unsteadiness over several decades of research, the feature of turbulent interactions is difficult to characterize and predict. Many different turbulence models have been developed to describe turbulence characteristics, but they have their own limitations. So, numerical simulations still cannot provide an absolutely accurate result. All these reasons make turbulent interactions a very active area of research. The purpose of this thesis is to use CFD techniques to simulate a shock wave-boundary layer interactions process in order to expand the comprehension about turbulent interactions, discuss the merits and limitations of different turbulence models in numerical simulations, and facilitate future works.

1.2 Shock Wave-Turbulent Boundary Layer Interaction

The interaction of a shock wave and a turbulent boundary layer is a universal occurrence in the supersonic aircraft's flight. As usually assumed, in 1939, Antonio Ferri wrote the first reference to this possible interaction [3][4]. The incentive of this phenomenon has been studied for over 60 years as the first step of the discovery of this complex interaction. The subsonic flow, such as the boundary layer, and the supersonic flow, such as the shock wave interaction can be a special study area. SWBLI always forms at the location of complex geometry, such as engine inlet, diffuser, and

wing-fuselage junctions. These areas usually experience detrimental effects, such as the losses of pressure, excessive heat transfer, and the separation of flow. The detrimental effects on the supersonic flight is apparent in the development of supersonic aircrafts, such as the hypersonic X-15. The component mounted below the X-15 fuselage was burned by an impinging shock that caused the engine to separate from the plane during the high-speed engine test. It is possible that a hole would have burned through the fuselage, which would lead the X-15 to fall apart in flight [5][6]. The topics of shock waves and boundary layers interactions have become important throughout the continuous progress of the supersonic aircraft.

The study detailed here is a numerical simulation of SWBLI generated by an unswept fin. An unswept fin is placed in a Mach 2 flow which generates a planar oblique shock wave. Herein, the interaction happening at the base of the fin is the object of research. Along the oblique shock angle, the interaction is swept downstream. The assumption is that the fin leading edge is perfectly orthogonal to the surface of the test section. The Fig. 1.1 graphically demonstrates the characteristics of the fin and interactions. This figure comes from a previous study about Shock Wave-Turbulent Boundary Layer Interaction using a planar PIV technique [7]. The free stream flow is at Mach 2, and the fin's angle of attack is 15°. Such combination leads to the planar oblique shock wave at a wave angle of approximate 45°with respect to the free stream.

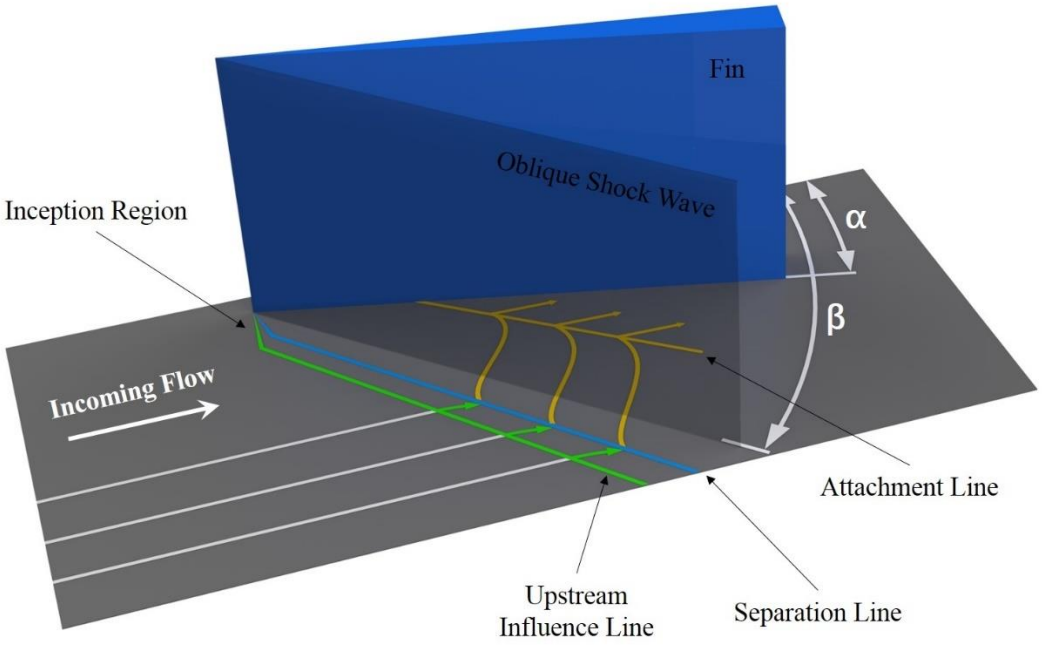


Figure 1.1: Diagram of the Fin-Generated SWBLI Flow Field

A λ-shaped structure is formed by the interaction between the shock wave and the boundary layer on the test section floor. Fig. 1.2 is the λ -shock structure's diagram. The λ -shock has an upper shock corresponding to the inviscid shock created by the model. Because of the presence of the shock wave, the jump in the pressure gradient leads to the separation of the boundary layer, as shown as a recirculation region in Fig. 1.2. The deflection of the flow incurred by the separated region results in the separation shock, which is the upstream portion of the λ structure. The separation shock turns some of the flow upward around the separation bubble and the flow encounters the horizontal flow behind the inviscid shock wave at the "triple point", in which a third shock appears to align the two flows[8]. This third weaker shock, as the bottom leg of the

λ structure, refers to the rear shock. The separation shock and this rear shock straddle a region of separation flow. The slip lines are generated by interactions between the flow behind the rear shock and the inviscid shock [9]. Although almost all of the details of this structure (e.g., separation length scale, unsteadiness, acoustic and thermal loading, etc.) depend on model geometry and incoming flow properties, the general structure is consistent. Additionally, according to Chapman et al.[10], the pressure increase leading to the separation of the flow does not rely on the model geometry for either laminar or turbulent 2-D interactions. This "free-interaction" phenomenon enlarges the pressure separation region at the centerline for blunt fin geometry [11].

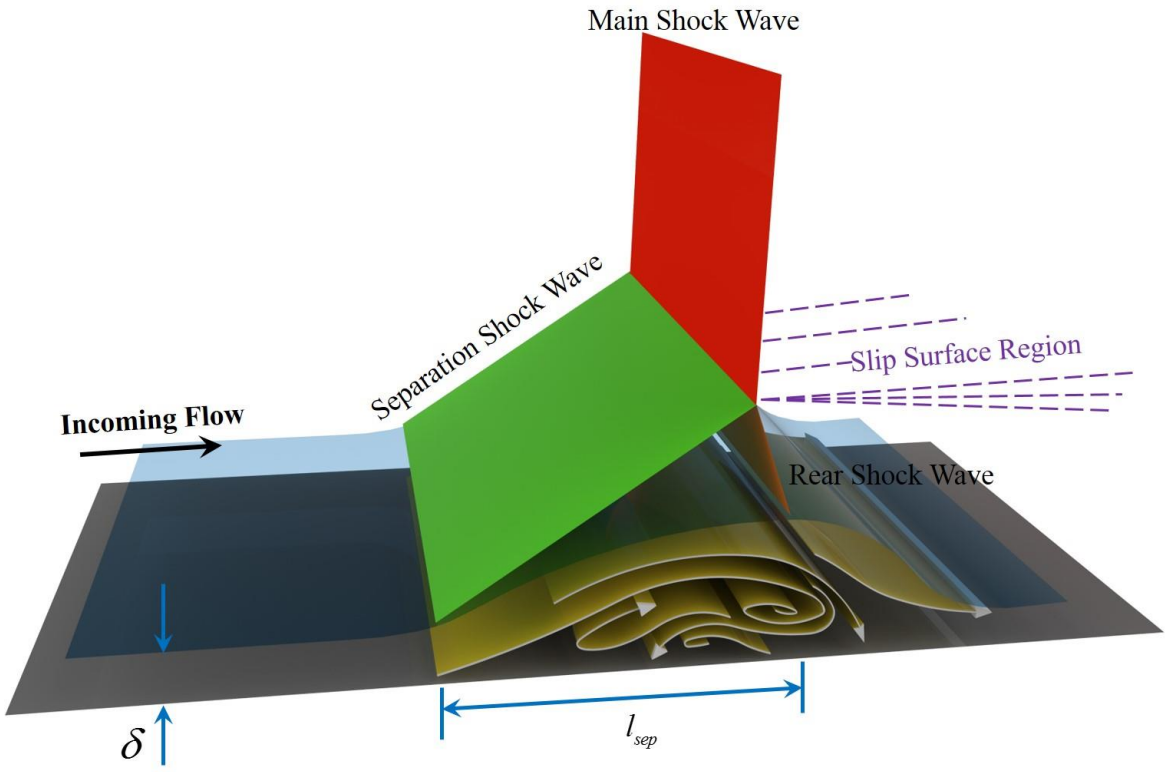


Figure 1.2: Diagram of the λ -Shock Structure

While some SWBLI are quasi-2D structures, the large majority of SWBLI encountered on supersonic aircrafts are inherently 3D. The largest portion of prior research into SWBLI has been concentrated on the quasi-2D interactions. While the structure of quasi-2D interactions is a significant portion of the necessary knowledge regarding SWBLI, the fact that a large number of the commonly encountered SWBLI are inherently 3D should encourage studies of these truly 3D interactions. The entirety of past research into the quasi-2D interactions has provided the base of knowledge for progress in 3D studies. However, the structure of the three-dimensional interaction has a few other important characteristics, as shown in Fig. 1.3. The third dimension allows out flow around the obstruction, but on the centerline, there is a stagnation point on the model's surface. The flow along the centerline that is processed by the normal shock wave is necessarily subsonic, but the flow that travels just below the triple point is processed by two or more oblique shocks and remains supersonic [12]. As the consequence of SWBLI, the supersonic "jet" is produced and leads to extremely high value of pressure and heat transfer where it impinges on the non-slip surface [13]. Away from the centerline of the three-dimensional interaction, the separated region is quickly swept downstream, causing curvature in the separation line. This primary vortex induces counter-rotating horseshoe vortices immediately downstream [11], and the number of these vortex pairs decreases with decreasing Reynold number [14]. The upstream edge of the primary vortex demarcates the primary separation distance, l_{sep} [15][16]. Note that this does not necessarily correspond to the farthest upstream extent of the disturbed flow, as the initial rise in pressure and heating can occur several diameters upstream of separation [15]. The location where this initial

deviation from undisturbed values occurs is termed the upstream influence (UI) location.

Classifications for swept SWBLI have been previously described by Settles and Dolling [18]. The major classifications are semi-infinite and non-semi-infinite, and within the semi-infinite classification a subdivision exists between dimensional and dimensionless interactions. Semi-infinite interactions refer to those SWBLI in which the shock generator is large enough that increases in the generator dimensions would not result in changes to the interaction. One example of these is an unswept fin-generated SWBLI. In the case of a sufficiently large fin, any increase in the fin's length or height would not change the dimensions of the SWBLI [15]. A non-semi-infinite interaction is one in which a change in the shock generator dimensions would result in changes in the interaction properties, such as small protuberances. As the height of a small protuberance is increased, the SWBLI takes a very different structure and dimension.

Dimensional interactions are those SWBLI in which the shock generator imposes a length scale on the interaction. In contrast to dimensionless interactions such as sharp fins, these generators cause a significant scaling of the interaction. For a sharp unswept fin, the major lengths such as the fin length do not matter to the interaction. In this case, the flow experiences a compression, and it does not have any consideration for the length of the fin which causes that compression. Although there is an inception zone with a certain curvature, this curvature does not correspond to any dimension of the fin itself. Any change in the dimensions of the fin (length, height, etc.) results in no change in the curvature of the inception zone. In contrast, a blunt fin results in a particular curvature of the interaction which is inherently tied to the diameter of the

fin's leading edge. An increase in this diameter would result in a change in the curvature of the interaction, and this length scaling is what makes the interaction dimensional.

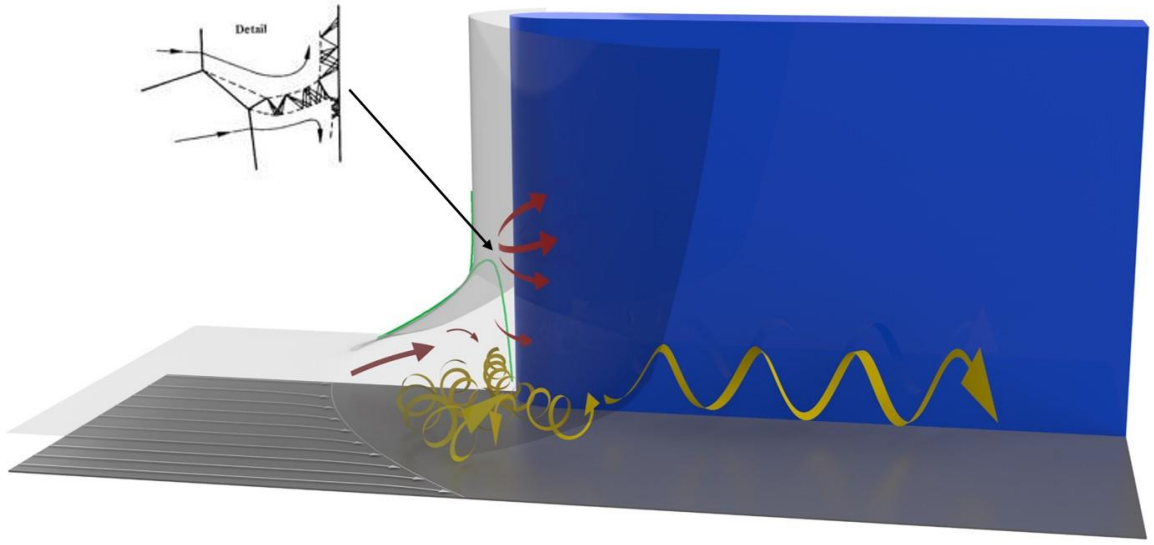


Figure 1.3: The Structure of Turbulence within the SWBLI induced by the Blunted Fin

1.3 Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) is the numerical analysis of fluid flow, heat transfer, and other related phenomena. By using a set of algorithms, CFD solvers have the capacity for modeling and simulating the flow feature about complex geometry. CFD methodology has a wide application in many technological fields such as airfoil design in aeronautics, drag simulation in automobile design, jet and thermal flow in engine design, and cooling airflow in an electronic product. CFD methodology has many different discretization approaches and turbulence models, but they share the same basic procedure. During pre-processing, the geometry of the physical

model is developed using computer aided design (CAD), and the surfaces and volumes of interest are defined. Next, the area of interest is subdivided into a large number of cells or control volumes. The mesh may be uniform or non-uniform, structured or unstructured, consisting of a combination of hexahedral, tetrahedral, prismatic, pyramidal or polyhedral elements. After defining the boundary conditions, the discretized governing equations are solved iteratively from a steady state to an unsteady state. Finally, a postprocessor is used to recompose the numerical solution and to visualize a complete description of the flow throughout the domain.

Over the past few decades, the Navier-Stokes equations, which are the governing equations for viscous flows, have been incorporated into computational fluid dynamics (CFD) codes. As the governing equations are non-linear Partial Differential Equations (PDE), the analytical solution is difficult to obtain in most of the cases. However, the numerical methods offer a possibility to solve the PDEs for a valid solution. The Navier-Stokes PDEs can be discretized and rewritten as algebraic equations that relate the velocity, temperature, pressure, density and other variables, and they are solved iteratively until the solution is converged. According to discretization methods, CFD can be categorized into the Finite Volume Method (FVM), the Finite Element Method (FEM), and the Finite Difference Method (FDM). The finite volume method (FVM) is a popular approach for most aerodynamic flows, as it has an advantage in memory usage and solution speed, especially for high Reynolds number turbulent flows and source term dominated flows (like combustion) [20]. The finite element method (FEM) is usually applied in solids mechanics analysis but is also applicable to fluids. Although FEM requires special care to ensure a conservative solution, it can

be much more stable than the finite volume approach in certain circumstance [21]. Nevertheless, compared with FVM, FEM requires more memory and time costs [22]. The finite difference method (FDM) [23] utilizes embedded boundaries or overlapping grids to resolve complex geometry with high accuracy and efficiency [24].

The most general form of governing equations, particularly those incorporated in computational fluid dynamics (CFD), has been expressed in Eq. 1.1 with appropriate boundary conditions [5].

$$\partial_t Q + \nabla \cdot \vec{F}^c - \nabla \cdot \vec{F}^v = S \quad \text{in } \Omega,\ t > 0 \tag{1.1}$$

where Q represents state variables, $\vec{F}^c$ and $\vec{F}^v$ are convective and viscous fluxes respectively, and S is a general source term.

$$Q = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix}, \quad \vec{F}_i^c = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + P\delta_{i1} \\ \rho u_i u_2 + P\delta_{i2} \\ \rho u_i u_3 + P\delta_{i3} \\ \rho u_i h \end{bmatrix}, \quad \vec{F}_i^v = \begin{bmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ u_j \tau_{ij} + \mu_{tot}^* C_P \partial_i T \end{bmatrix} \quad i, j = 1, 2, 3 \tag{1.2}$$

In the above equation, ρ represents density, ρu_1, ρu_2, ρu_3 are the flux components of x-momentum, y-momentum and z-momentum, respectively, ρE refers to the flux of total energy, P refers to the static pressure, δ_{ij} refers to the Kronecker delta function, h refers to the fluid specific enthalpy, C_P refers to the specific heat capacity at constant pressure, T refers to temperature which is determined from ideal gas law given by $P = \rho RT$, R is the specific gas constant, and τ_{ij} is the viscous stress which can be expressed as

$$\tau_{ij} = \mu_{tot}\left(\partial_i u_j + \partial_j u_i - \frac{2}{3}\delta_{ij} \nabla \cdot \vec{u} \right) \tag{1.3}$$

where μ_{tot} is the sum of dynamic and eddy viscosity (or total viscosity). The dynamic viscosity is calculated by Sutherland's law and the eddy viscosity is calculated by the turbulence models. The total viscosity is given as follows

$$\mu_{tot} = \mu_{dyn} + \mu_t$$
$$\mu_{tot}^{*} = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_t}{Pr_t} \tag{1.4}$$

where Pr_d and Pr_t are dynamic and turbulent Prandtl numbers, respectively.

By solving the governing equations of the fluid flow, CFD provides important information about flow properties such as density, pressure, and temperature distribution. Furthermore, CFD methods have been modified not only to be applied in traditional aerodynamic research, but also in many different fields of investigation nowadays. Laminar flows have been the object of fluid dynamics study for a long time, but it is still a very active research area. Turbulent flows are a state of flow with a high disorder which is omnipresent and crucial for most engineering applications but requires improvement of the models to describe their characteristics. Although Newtonian fluids are predominantly studied, non-Newtonian flows are also topics of interest because the fluid's complex rheology properties greatly affect the resulting flow field. CFD analysis has become an important part of the engineering design because of its advantages of predicting the performance of novel designs or processes before manufacturing or implementation. Moreover, CFD shows a great role in shortening the investigation time costs and preventing the experiments from being too expensive and risky. On the other hand, CDF also has some disadvantages, such as the reliance on computational power, and the reliance on highly refined meshes to ensure accuracy

and convergence. Although computers have become increasingly more powerful with time, highly refined volumetric meshes could make the CFD run-times grow quite large and, in extreme cases, can take months to finish. Despite these limitations, CFD simulations are universally used in the aerospace industry today.

1.4 Organization of the Study

In order to establish a foundation for the unsteady analysis of turbulence problems of interest, Chapter 2 introduces the computational scheme of the fluid dynamic equations. The Reynolds-averaged Navier-Stokes equations, which are detailed both in the non-conservative form and the conservative form, are followed by a discussion of the turbulence closure employed. The numerical approach for the Riemann problem and the WENO higher order variable reconstruction are presented. The numerical scheme for spatial and temporal discretization of the conservation equations based on a finite volume method is presented and the boundary conditions for the equations are also discussed.

In Chapter 3, the details of the physical experiments, which were performed by the Advanced Flow Diagnostics Laboratory (AFDL) at Auburn University, are presented by discussing the initial conditions of the supersonic wind tunnel facility and the physical dimensions of the test section and experimental object. In addition, the design of converging-diverging nozzle to provide Mach 2 flow and the implementation of complete three-dimensional geometry is presented.

CFD pre-processing is fully discussed in Chapter 4 and Chapter 5. The process to generate the volume grids is shown followed by the evaluation of the grid quality by analyzing several different parameters. Next, the numerical parameters setup, the boundary conditions, and the simulation details are introduced.

The SWBLI phenomenon and supersonic flow behavior around the fin are discussed. The results of different turbulence closures are compared with the results of the physical experiment so that the applicable environment and the limitations of turbulence models can be analyzed. A non-obstacle surrounding case is developed to show the differences between in-tunnel numerical simulation and free stream numerical simulation.

The last chapter summarizes the whole study and presents the future directions to be investigated about both SWBLI and turbulence models.

Chapter 2

Numerical Scheme

In this chapter, the numerical scheme is introduced for the solution of the Navier-Stokes equations. This chapter outlines the Tenasi unstructured flow solver followed by discussions based on governing equations together with the turbulence models. Additionally, the important mathematical fluxes, along with discretization methodology and boundary conditions are also discussed.

2.1 Flow Solver

The Tenasi unstructured flow solver was introduced by SimCenter: National Center for Computational Engineering at the University of Tennessee, Chattanooga. Tenasi is a node-centered, FVM scheme that uses an implicit method applied to general multi-element unstructured grids for conducting their parallel computations. Tenasi provides the set of equations for 5 general flow systems: Incompressible [25], Incompressible Surface Capturing [26], Compressible, Arbitrary Mach Number [27], and Compressible Multi-species. The current research is dependent on the Arbitrary Mach Number (AMN) regime that applies a primitive-variable formulation for solving the governing equations. Meanwhile, the AMN flow system allows the solver to be used for flows

that range from lower subsonic (March number ≈ 0.0) to the supersonic. Moreover, Tenasi offers solutions to hybrid large eddy simulation (LES) and Reynolds-averaged Navier Stokes (RANS), in a bid to obtain closure of turbulence. Tenasi provides users more than ten different turbulence frameworks, including the one equation Spalart-Allmaras model, Menter's Scale Adaptive (SAS) model, Menter's Shear Stress Transport (SST) k-ω Model, Wilcox two equations k-ω Model, several version of Launder Shima Reynolds Stress Model (RSM), and the Classic Smagorinsky model. For these reasons, as compared with other flow solvers, Tenasi is competent for wide-variety of fluid flow simulations.

2.2 Governing Equation of Fluid Dynamics

In computational fluid dynamics, Direct Numerical Simulations (DNS), Reynolds-Averaged Navier Stokes (RANS), and Detached/Large Eddy Simulations (D/LES) are three major categories of approaches used for turbulent fluid flow simulation. Direct Numerical Simulations (DNS) [28] is used to numerically simulate the transient of Navier-stoke equations in the absence of the turbulence model, which requires highly resolved grids in order to resolve the entire range of spatial scales together with temporal scales of turbulence. As the DNS model highly oversteps the present computational power, recent studies tend to numerically simulate the Navier-Stokes equations by modeling a portion of turbulence and resolving the remaining flow field by the algorithm.

The turbulence in unsteady flow is formed by the large-scale eddies that disperse energy from

mean flow in the form of heat, and turbulent pulsation frequency is often more than 10,000 Hz. The time scale of the smallest high frequency turbulence is far less than that of the largest eddy structure within flow. This phenomenon is termed as separation of scale. The objective of the algorithm is to define this entire spectrum of turbulence. Therefore, the time scale for calculations, Δt, should be smaller compared to the rolling duration of the larger eddies and larger than the time range of the smallest vortex within the eddy. Thus, the bigger eddies are resolved while the smaller scale eddies are modeled.

Applying an average operation to the Navier-Stokes equations, Detached Eddy Simulations (DES) and Large Eddy Simulations (LES) [29] resolve large scales eddies that are strongly associated with momentum and energy transfer. The Subgrid Scale model (SGS) in the DES and LES approaches is employed to represent smaller scale eddies that are unresolved. Due to the capacity of resolving the large-scale turbulence, the DES and LES model performs well in separated flow regions when compared to RANS solutions. Nonetheless, at near wall regions, the DES and LES models are restricted and greatly rely on the refinement of grids because of small length-scales and time scales in this region. Since the DES and LES models provide a solution to the large-scale eddies and also models turbulence in small-scale, they require considerably less computational power compared to DNS, although being limited when simulating high Reynolds number flows [30].

Applying a filtering process for a whole time scale upon the laminar Navier-Stokes equations provides the Reynolds-averaged Navier-Stokes equations (RANS) [31]. As the RANS models all

turbulent scales, relative to the DNS and the DES/LES models, this modeling approach significantly declines computational requirements. Furthermore, RANS provides the velocity of mean flow that shows direct consequence and does not require averaging the instantaneous velocity in steady flow simulation. However, RANS model does not resolve the turbulent scales and therefore performs well in the near wall regions but has limitations in separated flow regions as the information is lost by applying the RANS averaging procedures [32]. This thesis would throw emphasis on the Shock-Wave/Boundary-Layer Interaction (SWBLI) and supersonic flow that behaves in a test section. The majority of the flow field is sufficiently computed by RANS but SGS modeling is needed for small portions of the flow.

2.2.1 Reynolds-averaged Navier-Stokes equations (RANS)

As regards to the Cartesian coordinates, the preconditioned unsteady three-dimensional Reynolds-averaged Navier-Stokes systems of equations [33][34] that were developed and applied in the AMN regime are defined as

$$M\Gamma^{-1}\frac{\partial}{\partial t}\int_{\Lambda} Q d\mathcal{V} + \int_{\partial\Lambda} \vec{F}\cdot\hat{\vec{n}} dA = \int_{\partial\Lambda} \frac{\vec{F}_v \cdot \hat{\vec{n}}}{Re_L} dA \tag{2.1}$$

where $\hat{\vec{n}}$ represents the external pointing unit normal vector to the control volume $\mathcal{V}$. The above equation presents the non-conservative form of Navier-Stokes equations. The vector of dependent variables is presented as

$$Q = \begin{bmatrix} \rho \\ u \\ v \\ w \\ P \end{bmatrix} \tag{2.2}$$

The components of the inviscid as well as viscous flux vectors are expressed as

$$\vec{F} \cdot \hat{\vec{n}} = \begin{bmatrix} \rho\Theta \\ \rho u\Theta + \hat{n}_x P \\ \rho v\Theta + \hat{n}_y P \\ \rho w\Theta + \hat{n}_z P \\ \left(e_t + abP\right)\Theta - a_t abP \end{bmatrix} \tag{2.3}$$

$$\vec{F}_v \cdot \hat{\vec{n}} = \begin{bmatrix} 0 \\ \Psi_x \\ \Psi_y \\ \Psi_z \\ Q_h \end{bmatrix} \tag{2.4}$$

$$\begin{aligned} \Psi_x &= \hat{n}_x\tau_{xx} + \hat{n}_y\tau_{xy} + \hat{n}_z\tau_{xz} \\ \Psi_y &= \hat{n}_x\tau_{yx} + \hat{n}_y\tau_{yy} + \hat{n}_z\tau_{yz} \\ \Psi_z &= \hat{n}_x\tau_{zx} + \hat{n}_y\tau_{zy} + \hat{n}_z\tau_{zz} \\ Q_h &= u\Psi_x + v\Psi_y + w\Psi_z + \hat{n}_x q_x + \hat{n}_y q_y + \hat{n}_z q_z \end{aligned} \tag{2.5}$$

where $a = \gamma - 1, b = M_r{}^2$; u, v, and w indicates the Cartesian velocity constituents in the x, y, and z axis, respectively, in addition to $\hat{n}_x$, $\hat{n}_y$, and $\hat{n}_z$, that represent the constituents of the normalized control volume face vector. Θ represents the velocity normal to a control volume face:

$$\Theta = \hat{n}_x u + \hat{n}_y v + \hat{n}_z w + a_t \tag{2.6}$$

where a_t indicates the grid speed:

$$a_t = -\left(V_x\hat{n}_x + V_y\hat{n}_y + V_z\hat{n}_z\right) \tag{2.7}$$

and the velocity of control volume face has been described as

$$\vec{V}_s = V_x\hat{i} + V_y\hat{j} + V_z\hat{k} \tag{2.8}$$

The reference length scale (L_r) , the reference values of density (ρ_r), velocity (U_r), and viscosity (μ_r) are introduced to normalize the variables in the preceding equations. Thus, the Reynolds number could be expressed as $Re = \rho_r U_r L_r / \mu_r$. Meanwhile, pressure normalization is done with the help of $\rho_r U_r^{\,2}$ and the equation of state becomes

$$P = \rho T / \gamma M_r^{\,2} \tag{2.9}$$

where T is non-dimensional temperature, $M_r = U_r / a_r$ indicates the reference Mach number, and $a_r = \sqrt{\gamma R T_r}$ is the reference speed of sound. As regards to a perfect gas, the expression of the reference enthalpy is done as $h_r = C_p T_r$. Besides that, the particular heat ratio definition is expressed as $\gamma = C_p / C_v$, and the Eckardt number is described as $E_c = U_r^{\,2} / h_r = (\gamma - 1) Mr^2$.

The stresses provided in Eq. 2.5 are presented as

$$\tau_{ij} = (\mu + \mu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.10}$$

where μ represents the molecular viscosity, μ_t shows eddy viscosity, correspondingly, τ_{ij} denotes the Reynolds stresses.

The Jacobian regarding the change to the primitive variables are expressed as:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ ab\phi & ab\rho u & ab\rho v & ab\rho w & b \end{bmatrix} \tag{2.11}$$

where $\phi = \left(u^2 + v^2 + w^2\right)/2$. The preconditioning matrix is defined as:

$$\Gamma = diag\left(1,1,1,1,\beta\right) \tag{2.12}$$

where $\beta = \min\left(M_r{}^2, 1\right)$. Nevertheless, the preconditioning is removed from both the Jacobian and the flux for this study, implying the term $\beta = 1$.

The Tenasi solver is a node-centered, FVM scheme that uses an implicit scheme applied to general multi-element unstructured grids. Thus, the governing equation (Eq. 2.1) is expressed as

$$N = \Gamma T + R = 0 \tag{2.13}$$

where

$$T = \frac{\partial}{\partial t}\int_{\Lambda} Q d\mathcal{V} \tag{2.14}$$

$$R = \int_{\partial\Lambda}\left(\vec{F} - \frac{1}{Re}\vec{F}_v\right)\cdot\vec{n}dA \tag{2.15}$$

R and T is the spatial components and the temporal component of the total residual N.

2.3 Turbulence Modeling

Turbulence closure is essential for the governing equations presented in the preceding RANS section. Spalart et al. [38] put forward the Detached Eddy Simulation (DES) model, switching between RANS and SGS models, which are developed on the bases of the local grid refinement.

Nonetheless, DES was shown to adversely affect the attached boundary layers in some situations [38]. In order to correct this problem, Spalart et al. [39] developed the Delayed DES (DDES) model that modifies the original length scale to compute local turbulent value using both grid length and flow features. Additional turbulence models use this approach are found in literature. The multi-scale model, developed by Nichols and Nelson [40], utilizes both grid length scales as well as turbulent length scales to define the SGS model. The Wilcox k-ω model [42] and

Menter's SST k-ω [31] turbulence model are examples of two such models that are widely used in CFD solvers.

2.3.1 Wilcox $k-\omega$ Model

In this study, the Wilcox two equations $k-\omega$ model has been applied [42]. Like all eddy viscosity models, the model applies the Bossiness approximation to calculate the Reynolds stresses. Two different equations related to turbulence kinetic energy (k) as well as turbulence dissipation (ω) are employed to create a velocity scale together with length scale for calculating the eddy viscosity. The velocity scale here is the turbulent kinetic energy (k) and the length scale as $k^{\frac{1}{2}}/\omega$. The model uses below mentioned equations [42] for calculating the molecular stress tensor, t_{ij}, along with calculation of specific Reynolds-stress tensor, τ_{ij}:

$$t_{ij} = 2\mu\overline{S_{ij}} \tag{2.16}$$

$$\rho\tau_{ij} = 2\mu_t\overline{S_{ij}} - \frac{2}{3}\rho k\delta_{ij} \tag{2.17}$$

$$\overline{S_{ij}} = S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{2.18}$$

$$S_{k\omega} = \rho\tau_{ij}\frac{\partial u_i}{\partial x_j} = \mu_t\left(2S_{ij}S_{ij} - -\frac{2}{3}\left(\nabla\cdot\vec{u}\right)^2\right) - \frac{2\rho k}{3}\left(\nabla\cdot\vec{u}\right) \tag{2.19}$$

$$\mu_t = \frac{\rho k}{\tilde{\omega}} \tag{2.20}$$

$$\tilde{\omega} = max\left\{\omega, C_{lim}\sqrt{\frac{2\overline{S_{ij}}\,\overline{S_{ij}}}{\beta^*}}\right\} \tag{2.21}$$

$$C_{lim} = \frac{7}{8} \tag{2.22}$$

where ρ and k denote mean density and turbulence kinetic energy. μ and μ_t represent molecular viscosity and eddy viscosity, correspondingly. The quantity $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ is the conventional mean strain-rate tensor, δ_{ij} denotes the Kronecker delta, ω is the specific rate of dissipation, C_{lim} indicates the stress-limiter strength and β^* depicts a turbulence-model closure coefficient. It is recommended by Wilcox [42] that the $C_{lim} = 0.95$ gives better estimation of shock-separated flows up to Mach 3.

The turbulence kinetic energy, k, together with dissipation, ω, are governed by solving the following equations

$$\frac{\partial}{\partial t}\int_{\Lambda} \rho k d\mathcal{V} + \int_{\partial\Lambda} \rho k \Theta dA = \int_{\partial\Lambda} \left(\mu + \sigma_k \frac{\rho k}{\omega} \right) \overrightarrow{\nabla k} \cdot \hat{\vec{n}} dA + \mathcal{V}\left[S_{k\omega} - \beta^* \rho k \omega \right] \tag{2.23}$$

$$\begin{aligned} \frac{\partial}{\partial t}\int_{\Lambda} \rho \omega d\mathcal{V} + \int_{\partial\Lambda} \rho \omega \Theta dA = & \int_{\partial\Lambda} \left(\mu + \sigma_\omega \frac{\rho k}{\omega} \right) \overrightarrow{\nabla \omega} \cdot \hat{\vec{n}} dA \\ & + \mathcal{V}\left[\alpha \frac{\omega}{k} S_{k\omega} - \beta \rho \omega^2 + \sigma_d \frac{\rho}{\omega} \overrightarrow{\nabla k} \cdot \overrightarrow{\nabla \omega} \right] + \Omega_G + \Omega_R \end{aligned} \tag{2.24}$$

where the turbulence-model closure coefficients

$$\begin{aligned} & \alpha = \frac{13}{25}, \quad \beta^* = \frac{9}{100}, \quad \sigma_\omega = \frac{1}{2}, \quad \sigma_k = \frac{3}{5}, \quad Pr_T = \frac{8}{9} \\ & \sigma_d = \begin{cases} 0, & \frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \le 0 \\ \sigma_{d_0}, & \frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} > 0 \end{cases}, \quad \sigma_{d_0} = \frac{1}{8} \\ & \beta = \beta_0 f_\beta, \quad \beta_0 = 0.0708, \quad f_\beta = \frac{1+85\chi_\omega}{1+100\chi_\omega} \\ & \chi_\omega \equiv \left| \frac{\Omega_{ij}\Omega_{jk}S_{ki}}{\left(\beta^*\omega\right)^3} \right|, \quad S_{ki} = S_{ki} - \frac{1}{2}\frac{\partial u_m}{\partial x_m}\delta_{ki} \end{aligned} \tag{2.25}$$

where, $\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$ represents the rotational strain rate tensor.

2.3.2 Blended SST $k-\varepsilon, k-\omega$ Turbulence Model

The Shear-Stress Transport model formulated by Menter [31] facilitates the prediction of adverse pressure-gradient-dominant flows by means of the transport of the principal shear stress. It has been successfully employed for simulating the behavior of complex turbulent flows [43]. The blended $k-\varepsilon, k-\omega$ turbulence model [44][36] which is applied in this research is determined by:

$$\frac{\partial}{\partial t}\int_{\Lambda} \rho k d\mathcal{V} + \int_{\partial\Lambda} \rho k \Theta dA = \frac{1}{Re}\int_{\partial\Lambda} \left(\mu + \sigma_k \mu_t\right)\overrightarrow{\nabla k}\cdot\hat{\vec{n}} dA + \mathcal{V}\left[\tilde{S}_{k\omega} - \frac{\rho k^{3/2}}{l_t} + c_d \overline{p'' d''} \right] \tag{2.26}$$

$$\begin{aligned}\frac{\partial}{\partial t}\int_{\Lambda} \rho\omega d\mathcal{V} + \int_{\partial\Lambda} \rho\omega \Theta dA = \\ \frac{1}{Re}\int_{\partial\Lambda} \left(\mu + \sigma_\omega \mu_t\right)\overrightarrow{\nabla\omega}\cdot\hat{\vec{n}} dA \\ + \mathcal{V}\left[\frac{Re_L \delta\rho}{\mu_t} S_{k\omega} - \beta\rho\omega^2 + c_c \beta^* \alpha_1 {M_t}^2 \rho\omega^2 + 2\rho c_D \sigma_{\omega 2} \frac{1}{\omega}\overrightarrow{\nabla k}\cdot\overrightarrow{\nabla\omega} - c_d \frac{\rho Re_L \overline{p'' d''}}{\mu_t} \right] + \Omega_G + \Omega_R\end{aligned} \tag{2.27}$$

where the eddy viscosity, production, turbulent length scale, and pressure dilatation terms are expressed as

$$\mu_t = Re_L \frac{a_1 \rho k}{\hat{\omega}} \tag{2.28}$$

$$\hat{\omega} = \max\left(a_1\omega, F_2\sqrt{2\Omega_{ij}\Omega_{ij}}\right), \qquad \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \tag{2.29}$$

$$\tilde{S}_{k\omega} = \min\left(S_{k\omega}, \frac{10\rho k^{3/2}}{l_t} \right) \tag{2.30}$$

$$S_{k\omega} = \frac{\mu_t}{Re_L}\left(2S_{ij}S_{ij} - \frac{2}{3}\left(\nabla\cdot\vec{u}\right)^2\right) - \frac{2\rho k}{3}\left(\nabla\cdot\vec{u}\right), \qquad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.31}$$

$$\overline{p'' d''} = -S_{k\omega}\alpha_2 M_t^{\,2} + \alpha_3 \rho \epsilon M_t^{\,2} \tag{2.32}$$

$$l_{k\omega} = \frac{\sqrt{k}}{\omega\beta^*\left[1+\alpha_1 M_t^{\,2} c_c\right]} \tag{2.33}$$

$$l_t = \begin{cases} l_{k\omega}: & \text{RANS mode} \\ \dfrac{a_2\sqrt{k}}{D_l}: & \text{DES mode} \end{cases} \tag{2.34}$$

$$D_l = \max\left(a_2\omega\beta^*\left[1+\alpha_1 M_t^{\,2} c_c\right], F_3\sqrt{2S_{ij}S_{ij}}, F_4\frac{a_2\sqrt{k}}{C_{DES}\Delta_{max}}\right) \tag{2.35}$$

$$c_c = c_d = c_D = \left(1 - F_1\right) \tag{2.36}$$

here, $C_{DES} = 0.65$, Δ_{max} represents local maximum space between node-to-node, $M_t^{\,2} = 2k/c^2$, $\epsilon = c_\mu k\omega$, $c_\mu = 0.09$, $\alpha_1 = 1.0$, $\alpha_2 = 0.4$, $\alpha_3 = 0.2$, $a_1 = 0.31$ and $a_2 = 2.5$. The coefficients c_c, c_d, and c_D put limitations for their related terms to the region outside of the boundary layer. In the SGS model, the turbulent length scale l_t functions as maximizing the local dissipation of k and aims to control the size of k and eventually μ_t. This property averts μ_t against huge growth or losing the spatial changes due to increase in distance from the viscous surface.

The blending functions are expressed as $F_1 = tanh\left(\varphi^4\right)$, $F_2 = tanh\left(\varphi_2^{\,2}\right)$, $F_3 = tanh\left(\varphi_3^{\,2}\right)$, and $F_4 = tanh\left(\varphi_4^{\,2}\right)$ where

$$\varphi = \min\left(\max\left(\frac{\sqrt{k}}{0.09\omega d_v}, \frac{500\mu}{Re_L\rho\omega d_v^{\,2}}\right), \frac{4\rho k\sigma_{\omega 2}}{D_{k\omega} d_v^{\,2}}\right) \tag{2.37}$$

$$\varphi_2 = \max\left(\frac{2\sqrt{k}}{0.09\omega d_v}, \frac{500\mu}{Re_L\rho\omega d_v^{\,2}}\right) \tag{2.38}$$

$$\varphi_3 = d_v / \left(C_{DES}\Delta_{max}\right), \qquad \varphi_4 = \varphi_3 / 2 \tag{2.39}$$

$$D_{k\omega} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\overrightarrow{\nabla k}\cdot\overrightarrow{\nabla \omega}, 10^{-14} \right) \tag{2.40}$$

where d_v indicates the distance to the viscous surface. However, application of F_3 as well as F_4 is made for removing the effect of grid-induced separation caused by the high grid dependence of the original DES scheme [36].

The blending function F_1 plays an important role as an indicator function for near-wall and farfield regions of the flow. The $k-\omega$ model form is recovered near the wall ($F_1 = 1$), and the model trends to $k-\varepsilon$ model form far from the wall ($F_1 = 0$) [31]. F_1 is applied for blending the model constants, which are among $k-\omega$ as well as $k-\varepsilon$ regions of the flow by applying $\phi = F_1\phi_1 + (1-F_1)\phi_2$ where ϕ_1 and ϕ_2 suggests the general constants as regards the $k-\omega$ as well as $k-\varepsilon$ regions, correspondingly.

The model constants for the SST $k-\omega$ field are presented as:

$$\sigma_{k1} = 0.85,\ \ \sigma_{\omega 1} = 0.5,\ \ \beta_1 = 0.075,\ \ \beta^* = 0.09,\ \ \delta_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} \tag{2.41}$$

whereas, the model constants for the $k-\varepsilon$ field are presented as:

$$\sigma_{k2} = 1.0,\ \ \sigma_{\omega 2} = 0.856,\ \ \beta_2 = 0.0828,\ \ \beta^* = 0.09,\ \ \delta_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} \tag{2.42}$$

and $\kappa = 0.41$ is applied on both regions. The boundary conditions used on the turbulent variables are

$$k_{farfield} = 1.0\times 10^{-7},\ \ \omega_{farfield} = 10.0,\ \ k_{viscous} = 0.0,\ \ \omega_{viscous} = \frac{C_v\mu}{\rho d_v^{\,2}} C_v = \frac{38.4}{\beta_1 Re_L} \tag{2.43}$$

In order to prevent division by zero, $\omega \geq 0.1$ is imposed for all control volumes. Moreover,

$k \geq 0.0$ is forced for maintaining physical kinetic energy of turbulence.

The source term Ω_G creates a sensitivity to spatial variations in the strain rate for the dissipation rate. This is particularly useful for determining the shock wave location. Ω_G is defined as

$$\Omega_G = \mathcal{V}\left[\frac{\mu_t}{Re_L}\frac{\nabla S \cdot \nabla S}{S}\right] \tag{2.44}$$

where

$$S = \sqrt{\hat{S}_{ij}\hat{S}_{ij}/2}, \qquad \hat{S}_{ij} = 2S_{ij} - \frac{2\delta_{ij}}{3}\nabla \cdot \vec{u} \tag{2.45}$$

whereas δ_{ij} indicates the Kronecker delta. In some incidence, the additional ω develops high dissipation on not only k but also μ_t that causes low drag calculations. As a result, this expression needs to be selectively applied.

The source term Ω_R applies realizability constraints to the turbulence model [45] to ensure adequate levels of dissipation. Ω_R is defined as,

$$\Omega_R = \max\left[\frac{\rho(\tilde{\omega}-\omega)\mathcal{V}}{\Delta t}, 0\right] \tag{2.46}$$

where Δt indicates the time step and $\tilde{\omega}$ is the smallest $\tilde{\omega} \geq 0.1$ that satisfies both

$$\tilde{\omega} \geq \frac{3}{2}S_{ii} \tag{2.47}$$

and

$$\tilde{\omega}^2 - \frac{3}{2}\left(\hat{S}_{ii} + \hat{S}_{jj}\right)\tilde{\omega} + \frac{9}{4}\left(\hat{S}_{ii}\hat{S}_{jj} - \hat{S}_{ij}^{\,2}\right) \geq 0.0 \tag{2.48}$$

Ω_R only activates if $\tilde{\omega} < \hat{\omega}$. By means of Ω_R, the turbulence model offers a smooth procedure for maintaining appropriate levels of dissipation.

2.4 Numerical Approach

2.4.1 HLLC Numerical Flux

High resolution near discontinuities, where shocks are a common occurrence, is of the most importance for Tenasi solvers. These cases come down to a classic Riemann problem. Numbers of estimations regarding Riemann solvers commonly used in CFD can be found. One such methodology was introduced by Roe [46]. However, it requires the entire decomposition of the eigensystem. Rather, the application of the approximate Riemann solver of Harten, Lax, and Van leer (HLL) [47] for two distinct contact waves (HLLC) is made to avoid the development of a complete eigensystem while still providing accurate results. The development presented hereunder follows the discussion of Toro [48][49][50].

Edge is reconstructed on the conserved variables, for example:

$$\vec{F}_{i+1/2} = \vec{F}\left(\vec{U}_{i\pm1/2_L}, \vec{U}_{i\pm1/2_R}\right) \tag{2.49}$$

wherein, $\vec{U}_{i\pm1/2_L}$ as well as $\vec{U}_{i\pm1/2_R}$ denotes the reconstructed conserved variables, subjected to both the left and right sides of the $i \pm 1/2$ cell edges, respectively (see Fig. 2.1).

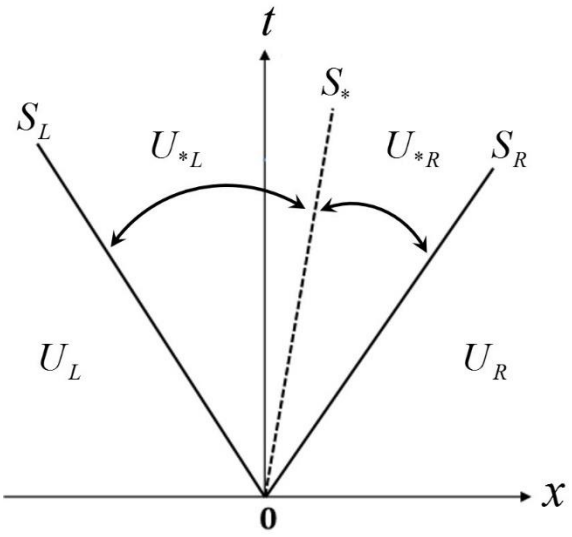


Figure 2.1: Schematic of the HLLC Approximate Riemann Solver

The HLLC Riemann solver [51] is used to estimate the Riemann solution by means of analyzing three wave speeds; S_R represents the largest speed wave, S_L represents the smallest speed wave, S_* shows contact wave speed. The contact wave then separates the two intermediate state vectors that are $\vec{U}_{*L}$ and $\vec{U}_{*R}$, which are restrictive by left as well as right waves. Appling these intermediate state vectors to calculate the intermediate fluxes $\vec{F}_{*L}$ as well as $\vec{F}_{*R}$ within these regions, the numerical flux HLLC is described as follows

$$\vec{F}_{i+1/2}^{HLLC} = \begin{cases} \vec{F}_L, & if \quad 0 \le S_L \\ \vec{F}_{*L}, & if \quad S_L \le 0 \le S_* \\ \vec{F}_{*R}, & if \quad S_* \le 0 \le S_R \\ \vec{F}_R, & if \quad 0 \ge S_R \end{cases} \tag{2.50}$$

By integrating over appropriate control volumes, the flux is given as

$$\begin{aligned} \vec{F}_{*L} &= \vec{F}_L + S_L\left(\vec{U}_{*L} - \vec{U}_L\right) \\ \vec{F}_{*R} &= \vec{F}_L + S_*\left(\vec{U}_{*R} - \vec{U}_{*L}\right) \\ \vec{F}_{*R} &= \vec{F}_R + S_R\left(\vec{U}_{*R} - \vec{U}_R\right) \end{aligned} \tag{2.51}$$

Some extra conditions are imposed to solve the algebraic problem.

For pressure and normal velocity,

$$\begin{aligned} p_{*L} &= p_{*R} = p_* \\ u_{*L} &= u_{*R} = u_* \end{aligned} \tag{2.52}$$

For tangential velocities,

$$\begin{aligned} v_{*L} &= v_L, \quad v_{*R} = v_R \\ w_{*L} &= w_L, \quad w_{*R} = w_R \end{aligned} \tag{2.53}$$

It is also assumed that

$$S_* = u_* \tag{2.54}$$

thus, is an estimate for S_* is known, the normal velocity component u_* in the star region is known. By means of right and left state waves, the intermediate wave speed can be identified as

$$S_* = \frac{p_R - p_L + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)} \tag{2.55}$$

As it is expressed in the Eq. 2.50, one can carry out the flux determination by means of various speeds of waves. $\vec{F}_L$ and $\vec{F}_R$ are considered easy to find because the right and left states are already identified. In order to determine the star region flux $\vec{F}_{*K}$, star region state variables must be analyzed. The intermediate states are expressed as

$$\vec{U}_{*K} = \rho_K \left(\frac{S_K - u_K}{S_K - S_*} \right) \begin{bmatrix} 1 \\ S_* \\ v_K \\ w_K \\ \frac{E_K}{\rho_K} + (S_* - u_K) \left[S_* + \frac{p_K}{\rho_K (S_K - u_K)} \right] \end{bmatrix} \tag{2.56}$$

where $K = L$ and $K = R$.

2.4.2 Weighted Essentially Non-Oscillatory (WENO) Edge Reconstruction

The WENO schemes are taken into consideration as extending the essentially non-oscillatory (ENO) scheme that was developed by Harten et al. [52] in the year 1987. The design of their scheme makes use of an adaptive polynomial reconstruction for the avoidance of the Gibbs phenomenon. The 1st WENO scheme was put forward by Oscher, Chan, and Liu [53] in the year 1994, followed by extension upon the ENO scheme. The key concept put forward by them for obtaining a higher order approximation requires utilizing a linear combination of multiple, weighted lower order reconstructions. The use of adaptive stencils provides more weight to the stencils that have smooth regions while the stencils processing discontinuities or oscillations that avoided. Therefore, the smoothness of stencils decides the choice of weights. Extensive use of WENO schemes has been applied to structured FVM solvers, but Tenasi is the first unstructured solver to make use of this stencils. 7th order reconstruction developed by Balsara and Shu [73] is used in this work.

2.4.3 Spatial Discretization

This study uses unstructured meshes for discretizing a specific spatial domain in which a dual mesh is used to create node-centered control volume. Multiple references discuss the implementation details [55][56].

Discretizing the equation over all the control volumes is explainable as

$$\mathcal{V}_i \frac{\partial Q_i}{\partial t} + R_i = 0 \tag{2.57}$$

where R indicates the spatial residual, containing each and every contribution from the source terms and numerical fluxes [55]. "i" specifies the individual control volume, and $\frac{\partial Q_i}{\partial t}$ describes the change in the conservative variables over a particular time step. Generalization of the above expression is mandatory for the purpose of getting the direct solution of the non-conservative variables by means of the following:

$$\mathcal{V}_i M_i \frac{\partial q_i}{\partial t} + R_i = 0, \quad M_i = \frac{\partial Q_i}{\partial q_i} \tag{2.58}$$

where M_i is a mapping of the conservative variables into non-conservative variable space.

In this study, the governing equations are discretized by a finite volume technique. Thus, the integrals in R are discretely approximated by summing the fluxes through each of the faces of the control volume [74].

$$R = \sum_{i=1}^{k} \left(\vec{F} - \frac{1}{Re_L} \vec{F}_v \right) \cdot \vec{n}_i \vec{A}_i \tag{2.59}$$

where k represents the number of faces on the control volume and A denotes the area of the face. Higher order flux evaluations are achieved by extrapolating the solution at the vertices to the faces by

$$\begin{aligned} F_{FL} &= F_L + \overrightarrow{\nabla F_L} \cdot \overrightarrow{r_{LF}} \\ F_{FR} &= F_R + \overrightarrow{\nabla F_R} \cdot \overrightarrow{r_{LR}} \end{aligned} \tag{2.60}$$

where $\vec{r}$ is a vector from the vertex to face.

2.4.4 Temporal Discretization

The transient formulations applied are the second-order implicit scheme. The differential form to approximate the time derivative term is express as [71][72]

$$\Delta q^n = \frac{\Delta t}{1+\theta}\frac{\partial}{\partial t}\left(q^{n+1}\right) + \frac{\theta}{1+\theta}\Delta q^{n-1} \tag{2.61}$$

where $\Delta q^n = q^{n+1} - q^n$. The first order accurate in time is given by $\theta = 0$ while the second order accurate in time is given by $\theta = 0.5$. "n" represent the present time step, "n+1" shows the next time step, whereas "n-1" represents the former time step.

The temporal component of the residual can be expressed as

$$T = \frac{\partial q}{\partial t} \approx \frac{\Delta q}{\Delta t} \tag{2.62}$$

where $q = \int Q d\mathcal{V}$. Substituted into Eq. 2.13

$$\Delta q^n - \frac{\theta}{1+\theta}\Delta q^{n-1} = -\Gamma^{-1}\frac{\Delta t}{1+\theta}R^{n+1} \tag{2.63}$$

The solution variables are assumed to be constant throughout the cell in order to evaluate volume integrals. Thus, $q = Q\mathcal{V}$.

$$\begin{aligned} \Delta q^n &= \mathcal{V}^{n+1}\Delta Q^n + Q^n\Delta\mathcal{V}^n \\ \Delta q^{n-1} &= \mathcal{V}^{n-1}\Delta Q^{n-1} + Q^n\Delta\mathcal{V}^{n-1} \end{aligned} \tag{2.64}$$

Since the value $\frac{\partial q}{\partial t}$ for each cell is dependent on the value at neighboring cells, the solution must be determined iteratively.

2.5 Boundary Condition

The application of the Navier-Stokes equations needs the specification of boundary conditions for non-periodic problems. The solution depends highly on the boundary conditions. The choice

of boundary conditions is decided by the physics of the problem of interest. In current formulation, the boundary conditions are implemented implicitly. In order to apply boundary conditions to the exact point at a boundary, an additional ghost cell is linked with each boundary cell. This inclusion of the ghost cells assists the boundary cells to be treated like inner cells. It allows the boundary cell unknown primary variable vector to be calculated with the inner cells. The ghost cells fill the outside of the boundary surface for a non-reflecting outflow or supersonic outflow state. Four cells are used in this case because the seventh-order WENO scheme asks for eight nodes. Subsequent to that, the outermost cells are copied to the four ghost cells. Moreover, sometimes, and accordingly, the velocity continues to follow the similar direction as in the case of the outermost cells regarding the ghost cells.

The boundary conditions are shaped with the help of starting with the quasi-linear form of the system, in addition to writing the equations in the direction that is normal to the boundary, η:

$$\frac{\partial Q}{\partial t} + A\frac{\partial Q}{\partial \eta} = 0 \tag{2.65}$$

Since $A = \frac{\partial F}{\partial t}$ is a constant-coefficient matrix. It can be diagonalized using the relation:

$$A = X\Lambda_A X^{-1} \tag{2.66}$$

where Λ is the matrix of eigenvalues and X is the right eigenvector matrix of A. Then, the quasi-linear system then becomes:

$$\frac{\partial Q}{\partial t} + X\Lambda_A X^{-1}\frac{\partial Q}{\partial \eta} = 0 \tag{2.67}$$

$$X^{-1}\frac{\partial Q}{\partial t} + \Lambda_A X^{-1}\frac{\partial Q}{\partial \eta} = 0 \tag{2.68}$$

Evaluating X^{-1} at some reference state which shows a constant matrix, it is able to place in the derivative as,

$$\frac{\partial X_0^{-1}Q}{\partial t} + \Lambda_A \frac{\partial X_0^{-1}Q}{\partial \eta} = 0 \tag{2.69}$$

however, 0 subscript points out the evaluation is constant. Subsequently, the characteristic variable vector W is described as $W = X_0^{-1}Q$, and W could be assessed as dependent on the sign of the eigenvalues on the boundary for the purpose of computing the values of Q_b which are then used in the evaluation of the numerical flux through the boundary. Q_b means the ghost cells' variables.

2.5.1 Inflow Boundary Condition for Subsonic In-Tunnel Inlet

The eigenvalues to be used in the system of equations of inflow are $\lambda_1 = \lambda_2 = \lambda_3 = \theta$, $\lambda_4 = \theta + c$, and $\lambda_5 = \theta - c$. The inflow is distinguished with the help of the foremost three eigenvalues that are negative, and the 4th eigenvalue that is positive, whereas the 5th eigenvalue is again negative. The dependent variables within the ghost cell for that boundary have been expressed as

$$\begin{bmatrix} l_0^1 \\ l_0^2 \\ l_0^3 \\ l_0^4 \\ l_0^5 \end{bmatrix} \begin{bmatrix} P \\ u \\ v \\ w \\ T \end{bmatrix}_b = \begin{bmatrix} l_0^1 \cdot Q_a \\ l_0^2 \cdot Q_a \\ l_0^3 \cdot Q_a \\ l_0^4 \cdot Q_l \\ l_0^5 \cdot Q_a \end{bmatrix} \tag{2.70}$$

where $l_0^i \quad (i = 1, 2 \cdots 5)$ are the left eigenvectors being assessed as fixed. (Note these are the rows of the matrix R_0^{-1}). Vectors Q_a and Q_l indicate the dependent variable vectors. Together with that, a subscript of "a" signifies the evaluation from a freestream location, whereas a subscript of

“l” signifies the evaluation inside of the boundary directly. Additionally, the “b” subscript signifies the ghost cells’ variables.

2.5.2 Inflow Boundary Condition for Supersonic Farfield Inlet

The supersonic inflow is characterized by the all five eigenvalues that are negative. The dependent variables within the ghost cell for that boundary have been expressed as

$$\begin{bmatrix} l_0^1 \\ l_0^2 \\ l_0^3 \\ l_0^4 \\ l_0^5 \end{bmatrix} \begin{bmatrix} P \\ u \\ v \\ w \\ T \end{bmatrix}_b = \begin{bmatrix} l_0^1 \cdot Q_a \\ l_0^2 \cdot Q_a \\ l_0^3 \cdot Q_a \\ l_0^4 \cdot Q_a \\ l_0^5 \cdot Q_a \end{bmatrix} \tag{2.71}$$

2.5.3 Outflow Boundary Condition for Supersonic Outlet

The supersonic outflow is differentiated by means of all five eigen values are positive. The dependent variable for the ghost cell for this boundary is given as

$$\begin{bmatrix} l_0^1 \\ l_0^2 \\ l_0^3 \\ l_0^4 \\ l_0^5 \end{bmatrix} \begin{bmatrix} P \\ u \\ v \\ w \\ T \end{bmatrix}_b = \begin{bmatrix} l_0^1 \cdot Q_l \\ l_0^2 \cdot Q_l \\ l_0^3 \cdot Q_l \\ l_0^4 \cdot Q_l \\ l_0^5 \cdot Q_l \end{bmatrix} \tag{2.72}$$

2.5.4 Impermeable Surface with Viscous Flow

Either Dirichlet boundary condition or Neumann boundary condition are used for the impermeable surfaces in a viscous flow in this study. A Dirichlet boundary condition is used

throughout the entire simulation generally for the nondimensional temperature at the boundary. $T = 0$ or $T = 1$ represent the wall is constantly heated or cooled, respectively. Moreover, the non-slip condition is applied on the velocity which is a subset of the Dirichlet condition. A non-slip condition is achieved by enforcing fluid for having a zero-velocity subjected to the fixed surface all the time. Therefore, the following variables can be set and kept constant. The ghost cells within this condition just acquire a straight copy of the variables from the boundary.

$$\begin{bmatrix} T_b \\ u_b \\ v_b \\ w_b \end{bmatrix} = \begin{bmatrix} T_l \\ u_l \\ v_l \\ w_l \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.73}$$

where "b" subscript is once again a representation of the ghost cells' variables whereas "l" subscript is a representation of the boundary cells' variables.

The second boundary condition applied for the impermeable surface in viscous flow is the Neumann boundary condition. The two Neumann boundary conditions used in this study are: (1) enforcing a zero gradient for both temperature and pressure off the wall, (2) requiring the pressure gradient off the wall to be equal to the source term.

Zero gradient

$$\frac{\partial T}{\partial \eta} = 0, \quad \frac{\partial P}{\partial \eta} = 0 \tag{2.74}$$

$$T_l = T_i, \quad P_l = P_i \tag{2.75}$$

Source term gradient

$$\frac{\partial P}{\partial \eta} = \frac{Gr}{Re^2} T \tag{2.76}$$

$$P_l = P_i + \frac{Gr}{Re^2} T_i(\eta_l - \eta_i) \tag{2.77}$$

where “l” subscript denotes the variables of the node on the boundary and “i” subscript denotes the variables of the node directly interior of the boundary. Gr is Grashof number.

Chapter 3

Wind Tunnel Facility and Model Configuration

In this chapter, the details of the physical experiments, which were performed by the Advanced Flow Diagnostics Laboratory (AFDL) at Auburn University, have been introduced through the wind tunnel establishment. The configuration of the supersonic nozzle is discussed as well as the introduction of the test section and the object assembly.

3.1 Wind Tunnel Facility

The experiments were carried out by means of the Supersonic Wind Tunnel (SST), which is situated in the Florida Center for Advanced Aero-Propulsion (FCAAP) within Florida State University (FSU). The facility was provided with compressed air at approximately 500 psi (~3,400 kPa) from a reservoir that possesses a volume of $113\ m^3$. The SST test section has a height of 3 inches (~76.2 mm), a width of 4 inches (~101.6 mm), and a length of 15.5 inches (~393.7 mm), correspondingly. The tunnel was designed for the purpose of generating free stream at a Mach number of 2 approximately. The pressure of stagnation was maintained at 51 psi (~350 kPa), whereas the stagnation temperature was maintained at 296 K approximately. The facility has an accumulated seeder that is considered as modified Wright nebulizer, which was mentioned by

Alkislar [61] that generates seeding particles with a mean diameter of 0.3 μm. Under these circumstances, the facility is able to generate runtimes of about 3 minutes, although the runtimes for data were restricted to about 2 minutes, which happens because of the accumulation and condensation of seeding particles on the test section window.

The fin model used for the shock wave-turbulent boundary layer interactions (SWBLI) was aluminum that possesses 37.5 mm height and 72mm length. The fin was positioned at an angle of attack of 15° that produces an oblique shock at 45° in Mach 2 flow. The supportive arrangement of the wind tunnel occludes the bottom part of the test section window from view; thus, in order to carry out experiments with ease, the fin was placed top-mounted as shown in Fig. 3.1.

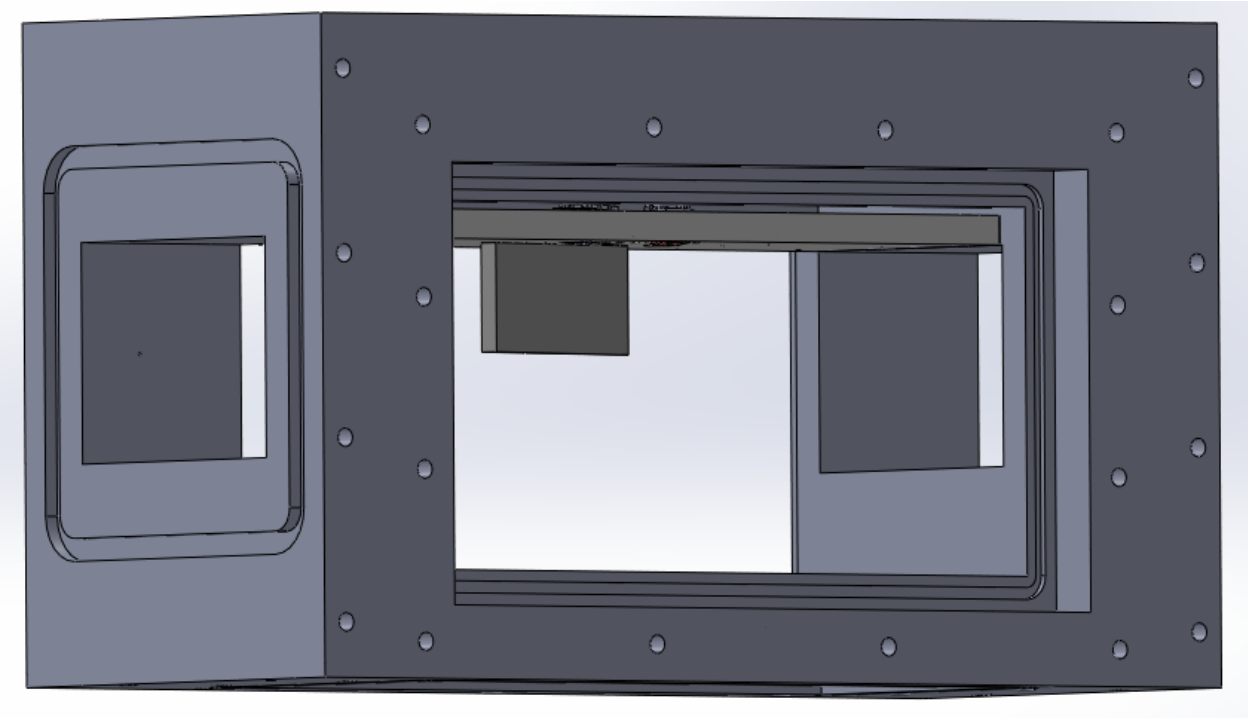


Figure 3.1: The Demonstration of Top-Mounting of the Fin in the Test Section

3.2 SolidWorks Model Configuration

In the physical wind tunnel experiment, the wind tunnel always consists of four sections ordered from upstream to downstream whose description is explicitly as follows:

1. Stagnation chamber: The stagnation chamber usually has a big volume to conserve gases and set up the flow's initial conditions. In the chamber, a uniform, low-speed flow is provided to the nozzle inlet.

2. Nozzle: The nozzle accelerates the incoming flows to their respective target velocities. The supersonic nozzles were designed with a convergent-divergent contour, while the subsonic and sonic nozzles were designed with a purely convergent contour.

3. Test section: The test section in this research is a rectangular pipe with a height of 3 inches and a width of 4 inches. The supersonic flow reaches the design Mach number at the nozzle outlet and flows into the test section. The fin (experimental object) was mounted in the test section.

4. Diffuser: In the diffuser, the high-speed flow from the test section is decelerated to low-speed.

For the numerical simulation, the initial flow conditions, such as temperature, pressure, density, velocity, and flow direction, can be determined through the input of boundary conditions at the nozzle inlet surface. Consequently, the stagnation chamber portion can be neglected. The supersonic flow that passes through the test section is decelerated by means of normal shocks in the diffuser. However, the complicated shock formation makes the algorithm easy to diverge. Using a diffuser at the back of the test section increases the difficulty of controlling the back

pressure of the test section. Thus, the diffuser section has not been used in the simulations. Since the shape of the nozzle in the wind tunnel is unknown, a converging-diverging nozzle is created in this study to accelerate the flow from stagnation state to Mach 2.

3.2.1 Converging-Diverging Nozzle Design for the Computational Domain

The nozzle is the primary component involved with the acceleration of the flow to a target velocity. For a supersonic flow, the nozzle must be designed with a convergent-divergent contour so that the flow can undergo acceleration through supersonic expansion downstream of the throat. The relation between area ratio and the Mach number is given in Eq. 3.1 taken from Pope [66]. This equation shows the effect of the cross-sectional area of a nozzle on the velocity of the flow.

$$\frac{dA}{A} = \left(M^2 - 1\right)\frac{du}{u} \tag{3.1}$$

Since the area and the velocity are positive, in the subsonic portion, the velocity of the flow will increase as the area of the duct decreases and vice versa. In the supersonic portion, the velocity of the flow will increase as the area of the duct increases and vice versa. The cross-sectional area reaches a minimum at M=1, which is known as the "throat". The throat area could be calculated by the nozzle cross-sectional area and local Mach number as the following equation [66]:

$$\frac{A}{A^*} = \left(\frac{1}{M}\right)\left\{\frac{1+\left(\frac{\gamma-1}{2}\right)M^2}{1+\left(\frac{\gamma-1}{2}\right)}\right\}^{\frac{\gamma+1}{2(\gamma-1)}} \tag{3.2}$$

where A is the cross-sectional area, A* is the area of the nozzle throat, γ is the specific heat ratio of air, and M is Mach number. The nozzle is designed to create a Mach 2 flow, and the cross-sectional area of the test section (the nozzle outlet) is known, so the nozzle throat for a given Mach

number was first calculated. Because the viscous wall is used in the flow solver, it can be known beforehand that the boundary layer growth in the simulation would reduce the total energy and available area of the flow. This viscosity effect decreases the actual Mach number produced in the test section. To overcome this, the actual design Mach number is 2.1 for the nominal Mach 2 condition. Assuming $\gamma = 1.4$, Eq. 3.2 was utilized for $M = 2.1$ to give $A/A^* = 1.84$.

For the converging portion of the supersonic nozzle, the shape of the nozzle was designed by fitting the contour according to the fifth order polynomial equation [67],

$$y(x) = h - h\left[10\left(\frac{x}{L}\right)^3 - 15\left(\frac{x}{L}\right)^4 + 6\left(\frac{x}{L}\right)^5\right] \tag{3.3}$$

with the constrains,

$$\begin{aligned} y'(0) &= y''(0) = 0 \\ y'(L) &= y''(L) = 0 \end{aligned} \tag{3.4}$$

where x and y are the Cartesian coordinates, h is the height of the converging portion of the nozzle, and L is the total length converging portion of the nozzle. In this research, the height and length of the converging portion of nozzle is 5 inches and 8 inches, respectively. The contour of the converging portion of nozzle is shown in Fig. 3.2.

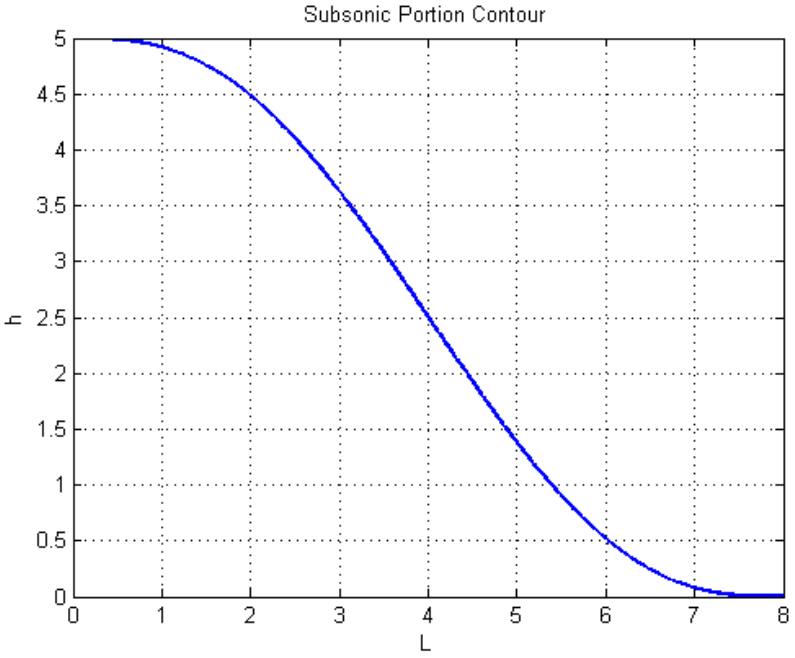


Figure 3.2: Subsonic Nozzle Portion Spline

For the diverging portion of the supersonic nozzle, the quasi-one-dimensional theory indicates nothing about the contour of the nozzle. If the variation respect to the flow direction $A = A(x)$ is not appropriate, the shock waves will coalesce and propagate into the test section. For this reason, the Method of Characteristics (MOC) [68], a technique for solving the hyperbolic partial differential equations for supersonic flow, has been employed in order to provide a proper design of the nozzle contour for shock free, isentropic flow. By using this technique, flow properties, such as direction and velocity, can be calculated at distinct points throughout a flow field. A MATLAB code is applied to this research to generate a gradual-expansion nozzle (Fig. 3.3 a) through the process of the Method of Characteristics. The gradual-expansion nozzle is a type of expansion nozzle that maintains a high-quality flow at the exit of the nozzle with desired conditions. Compared with the minimum-length nozzle (another type of expansion nozzle, Fig. 3.3 b), which

utilizes a sharp corner to provide the initial expansion, the gradual-expansion nozzle is better suited for use in supersonic wind tunnels. The behavior of these characteristic lines is shown for both a gradual expansion nozzle and a minimum-length nozzle in Fig. 3.3. The Mach waves reflect and intersect with each other in the non-simple region.

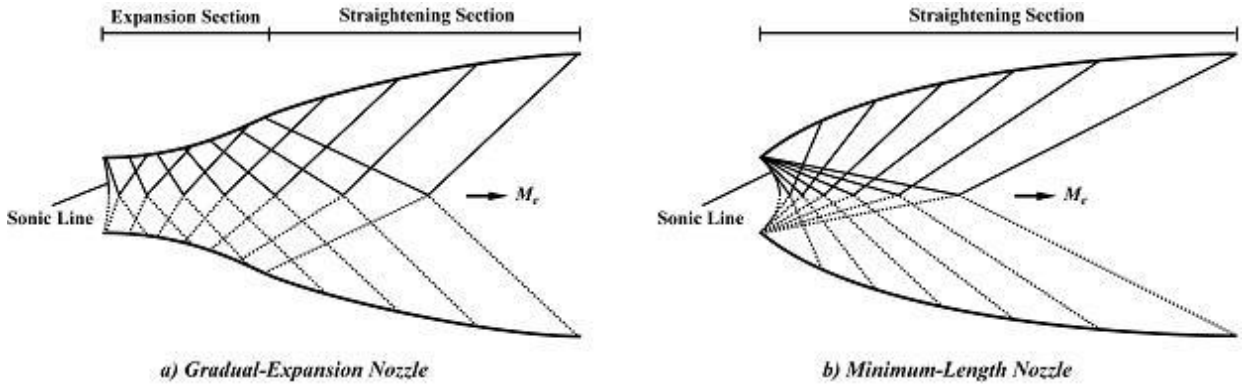


Figure 3.3: Characteristic Lines in Expanding Supersonic Flow

The MATLAB program provided contour coordinates of the diverging portion of the nozzle normalized to the throat height. To obtain the dimensional full-size coordinates, the normalized coordinates were simply multiplied by the throat height to get the full-size coordinates. The upper contour of the diverging portion of the nozzle is shown in Fig. 3.4.

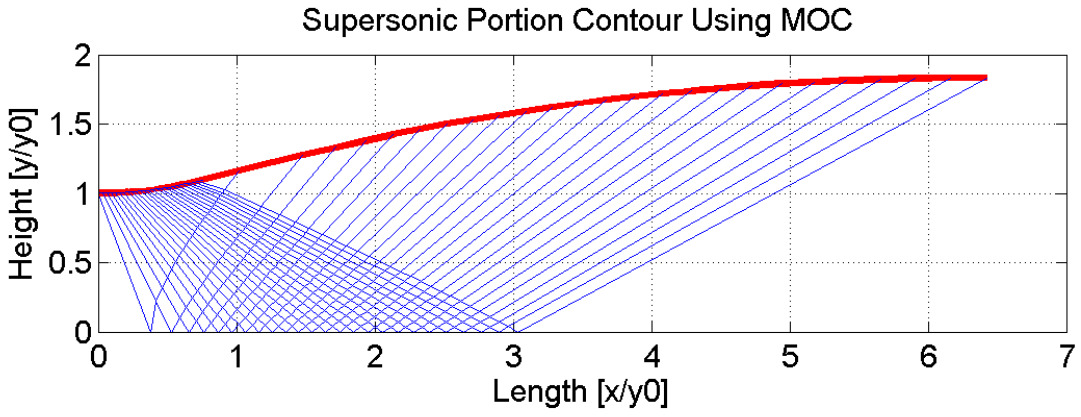


Figure 3.4: Supersonic Nozzle Portion Spline

In some cases, the flow is transonic at the throat but decelerates back to the subsonic flow in the diverging portion of the nozzle following the relation between area ratio and Mach number. Therefore, a constant area portion for 2 inches in length is given to the throat of the nozzle to let compressible flow transit from subsonic to supersonic. Notice that the cross-section of the test section is a rectangle (3 inches by 4 inches). The contour of upper and lower surfaces of the nozzle depends on the height of the test section while the contour of front and back surfaces of the nozzle depends on the width of the test section. Therefore, contours of the supersonic nozzle are calculated separately, but in the same process presented above. The contour of the converging-diverging nozzle is shown in the Fig. 3.5.

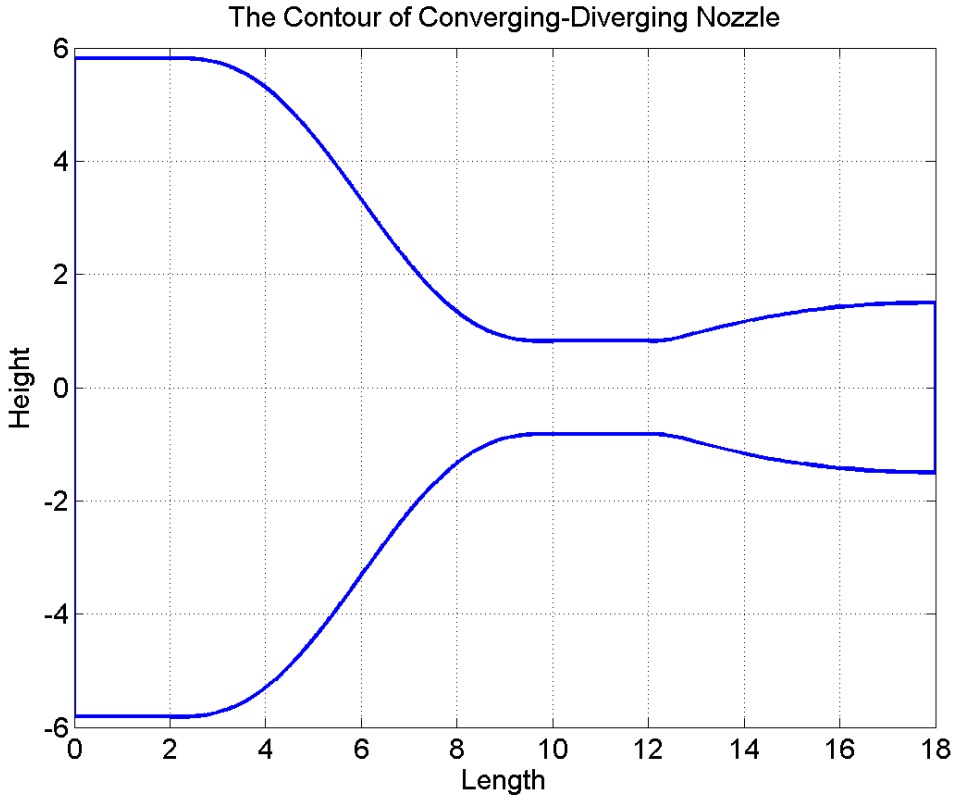


Figure 3.5: The Converging-Diverging Nozzle Spline (with Reservoir)

3.2.2 Test Section Assembly

The cross-section of the test section model in the CFD process has the same size as that in the physical experiment in section 3.1. The test section has a height of 3 inches, and a width of 4 inches. The connection between the nozzle and the test section is smooth to minimize the generation of strong oblique shock waves due to the corners on the sidewalls. The oblique shock waves produced at the inlet of the test section propagate and reflect in the tunnel, and they affect the supersonic flow properties. However, rather than eliminating oblique shock waves entirely, oblique shock waves can be only weakened by modifying the nozzle configuration. In the physical wind tunnel experiments, oblique shock waves exist and are visible in schlieren at the inlet of the test section because of the corner and a machining tolerance of about 0.1 mm (0.004 inch) in the nozzle and test section assembly. The unswept fin is positioned at the same place compared to the real experiments; the tip of the fin is 5 inches away from the inlet of the test section and 1.5 inches away from the right sidewall of the test section as indicated in Fig. 3.6. The fin has an angle of attack of 15° and is perpendicularly mounted on the bottom of the test section. The tip of the fin has been blunted by an arc with a radius of 0.002 inch (Fig. 3.7) to prevent the problems caused by a sharp point intersecting the wall. The final model configuration shown in Fig. 3.8 is used to build the unstructured meshes.

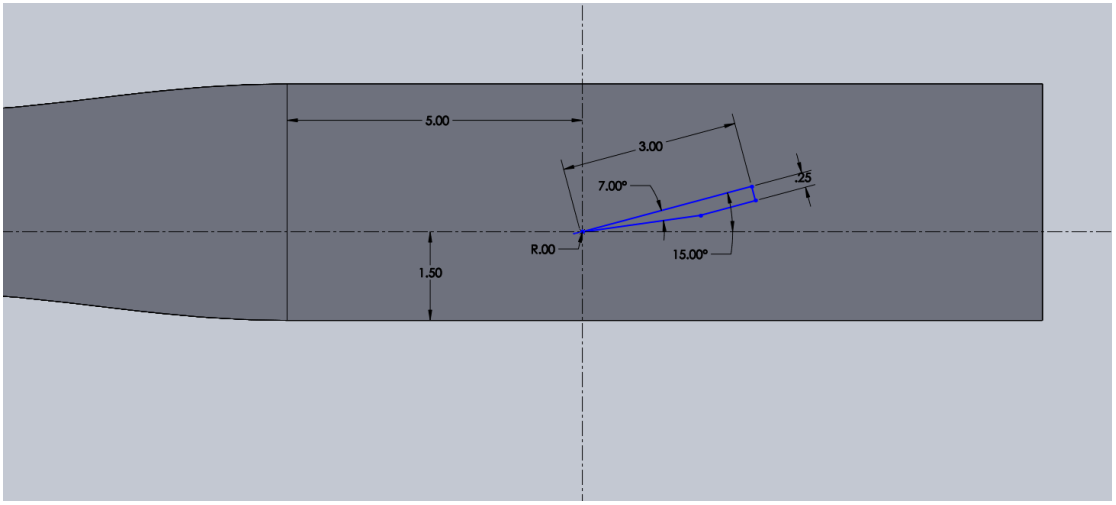


Figure 3.6: Engineering Drawing of the Fin in the Test Section

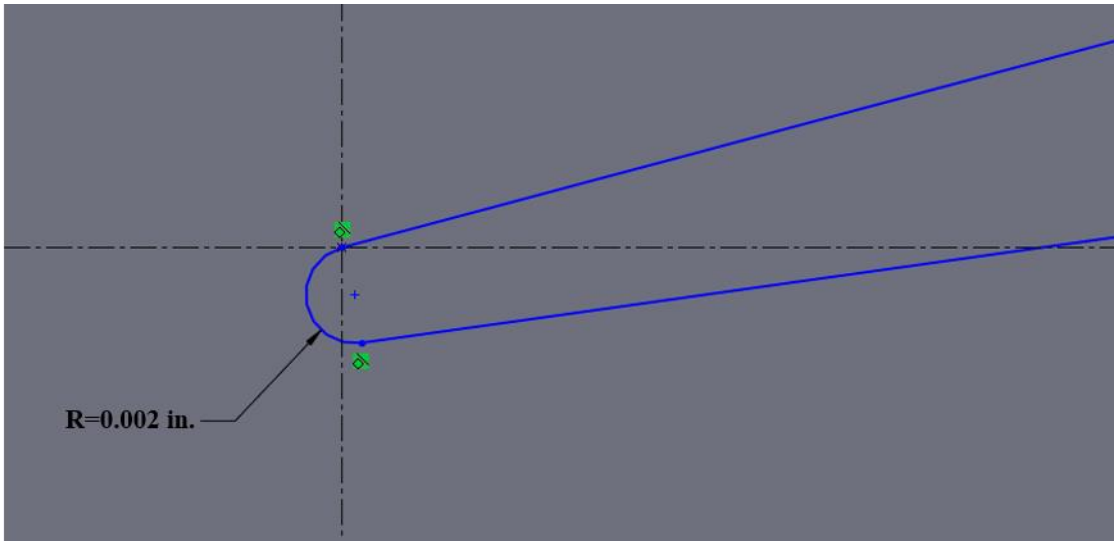


Figure 3.7: The Dimension of the Blunted Fin Apex

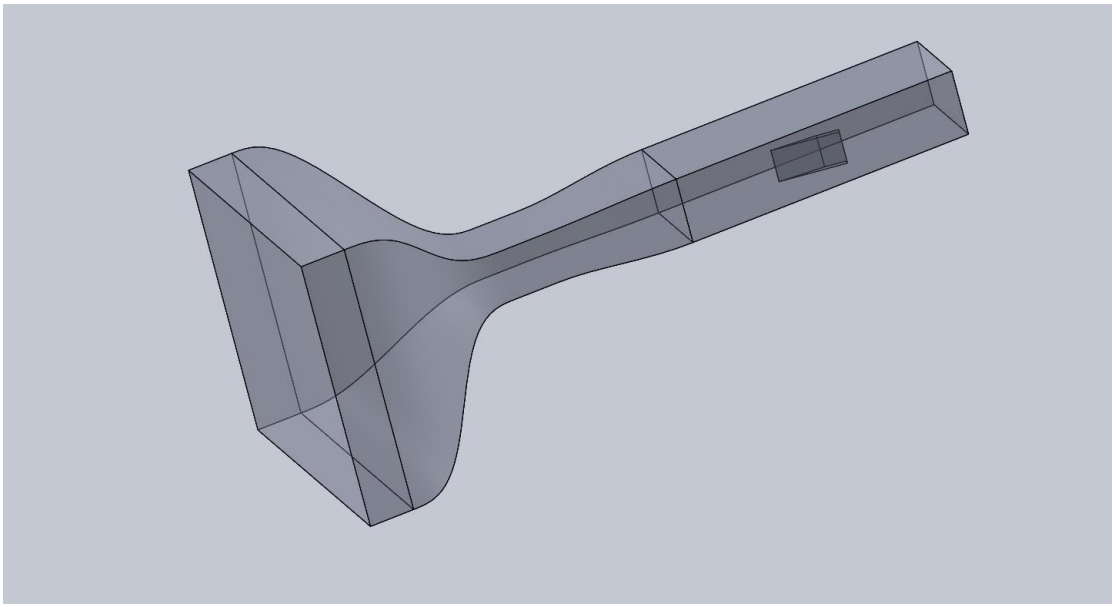


Figure 3.8: The Configuration of Final Model

Chapter 4

Grid Generation

CFD pre-processing can be divided into three major steps, which are model creation, grid generation, and numerical simulation parameters setup. In this chapter, the volume grid is generated using the wind tunnel model discussed in chapter 3. The process to achieve surface meshes and volumetric grid generation with Pointwise is presented. The evaluation of the grid quality, which affects the numerical simulation, is fully analyzed.

After configuring the model in SolidWorks, the surfaces of the model are discretized into small elements, and the computational domain should be filled by individual control volume. Using the volumetric grid solver, the Tenasi can employ numerical methods to calculate the solution of the governing equations which have been discretized into an algebraic equation. In this research, Pointwise software, a powerful meshing package, has been utilized to generate the unstructured mesh of the computational domain. Pointwise provides high quality meshes, which are essential for accurate CFD solutions, by means of structured, unstructured, and hybrid grids including an anisotropic tetrahedral extrusion (T-Rex) technique for boundary layer resolving meshes. By using Pointwise, the surface mesh generation is achieved by a bottom-up technique, and the discretization of the flow field (volumetric grid generation) is accomplished by the proper

extrusion of the surface domain meshes. Moreover, Pointwise provides several essential parameters to evaluate the quality of the model grids such as area ratio and minimum included angle, which impact the convergence of the later numerical simulation.

4.1 Surface Mesh Generation

Pointwise imports the database generated by several widely used engineering design software packages, such as SolidWorks and CATIA. The CAD model (quilts) is provided by a database to create connectors and domains. As shown in Fig. 4.1, the surfaces of the in-tunnel model have been classified into four portions: Inlet/Outlet, Converging-Diverging Nozzle, Test Section and Fin, which are indicated by different colors.

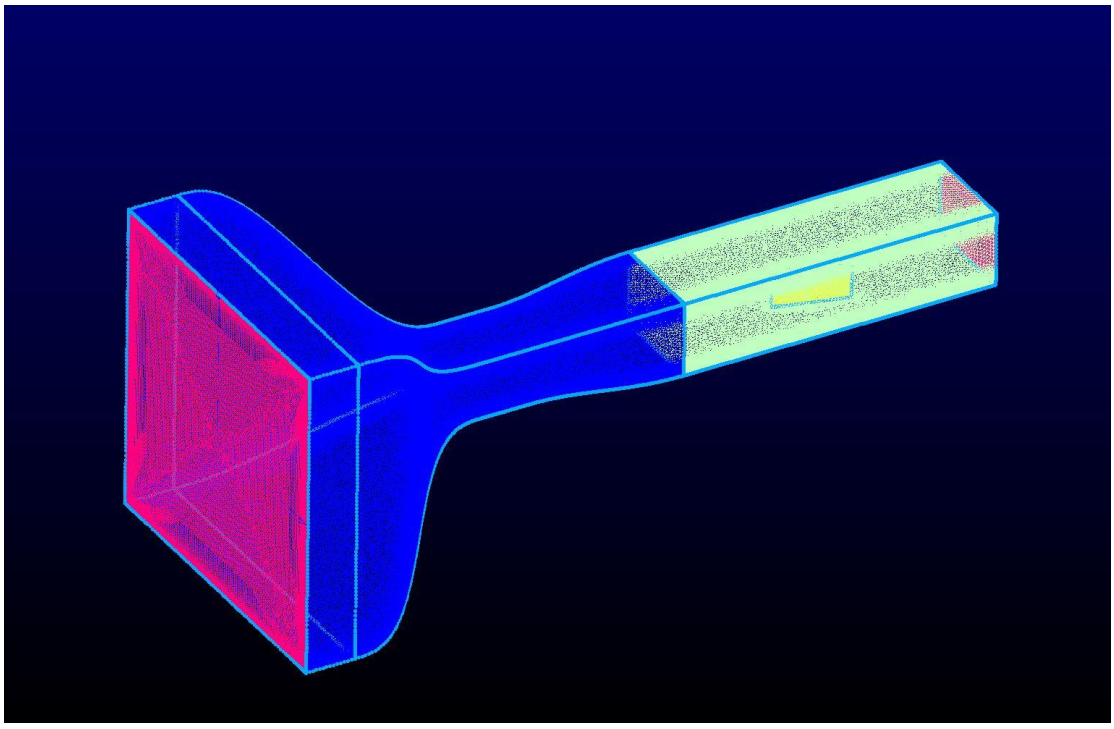
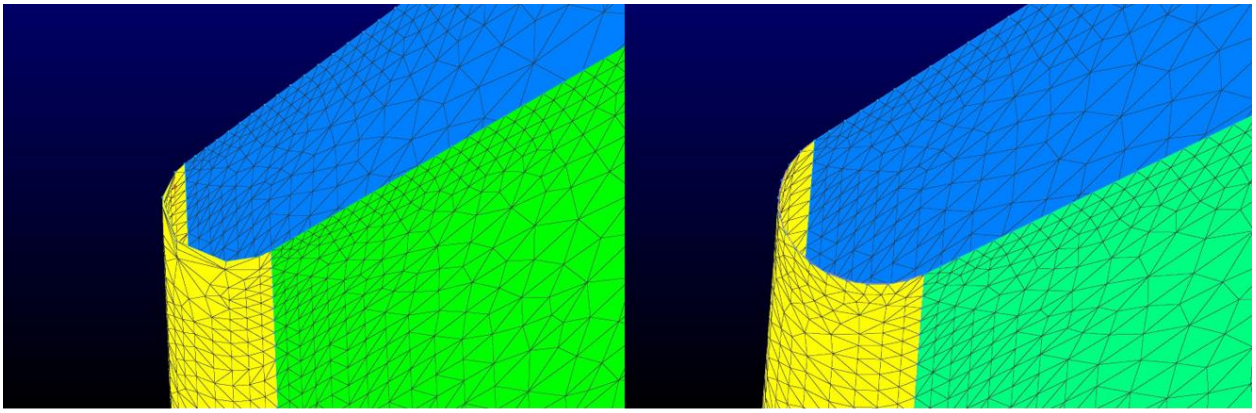


Figure 4.1: The Classification of Model Surfaces

The points distribution is determined by the flow speed and by geometric features. Since this experiment focuses on the supersonic case, the Δs used to decompose the connectors (the edges of the model) should be small enough to capture the characteristics of the supersonic flow and the shape of the geometry. Otherwise, the result of simulation will be inaccurate. However, over populating the mesh would produce too many elements in the grid and enormously increase the simulation time without improving the solution. Considering the flow field is designed to reach $Mach = 2$ in the test section, the average spacing of the test section connectors is about 0.05 inch along the surface. The converging part of the nozzle connectors has a larger average spacing because the flow is accelerating to transonic at the throat of the nozzle. The nozzle connectors near the outlet have the same spacing compared with the test section connectors in order to make the cells generate smoothly between different portions of the model and also avoid the inaccuracy caused by a huge jump in grid spacing. The smallest spacing is defined at the leading edge of the fin because more points are needed at the front of the fin to capture details of the shape. Additionally, the leading edge of the fin naturally requires more points on the wall which gives better accuracy for the oblique shock wave. Figure 4.2 shows the difference between segmenting the blunted head connector in 7 points (Left) and 11 points (Right). Some meshing details about different portions of the model are given in Table 4.1.



a) The Separation of the Connector of Blunted Fin Apex in 7 Points

b) The Separation of the Connector of Blunted Fin Apex in 11 Points

Figure 4.2 The Improvement of the Fin Apex Curve Fitting

Portions of model	Surface Domain meshes			Volume Grids	
	Average spacing in connectors (in)	T-Rex boundary conditions	Δy	T-Rex boundary conditions	Δy
Inlet section	0.120	Match		Match	
CD Nozzle	0.055	Wall	0.005	Wall	1.0×10^{-5}
Test Section	0.050	Wall	0.005	Wall	1.0×10^{-5}
Fin (blunted head domains)	0.007	Wall	0.0004	Wall	1.0×10^{-5}
Fin (other domains)	0.007	Wall	0.001	Wall	1.0×10^{-5}
Outlet section	0.055	Match		Match	

Table 4.1: The Details of Meshing Processes in Different Portions of the Model

The high aspect ratio anisotropic cells, also called T-Rex cells in Pointwise, are applied in meshing the domains to resolve the high curvature geometry at the intersection of surface domains.

Δy refers to the initial spacing of the T-Rex layers. The growth rate of 1.1 and the advancing front algorithm are used in generating T-Rex layers while meshing the domains. The different values of the initial spacing of the T-Rex layers are indicated in the Table 4.1. Figure 4.3 shows the T-Rex effects on the unstructured meshes of the domain in the blunted head of the fin.

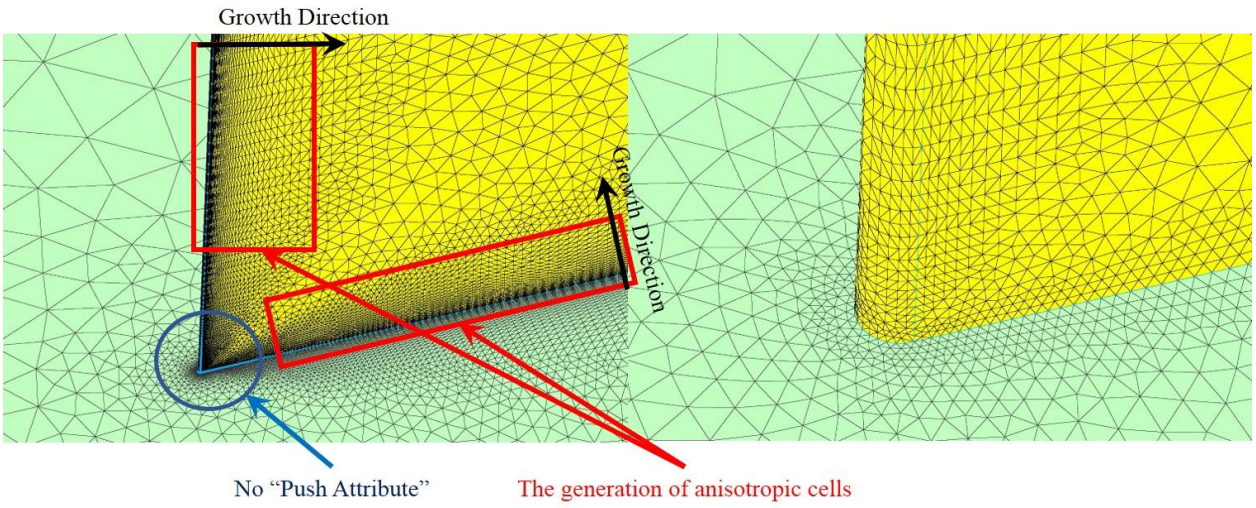


Figure 4.3: The Generation of Anisotropic Cells on Domain Meshes

The inlet and outlet surface are set to "Match" which uses the "Push Attribute" to push the viscous boundary attributes from the volume grid to the surface grids and connectors to be consistent with the volumetric T-Rex boundary conditions. The Push Attribute only applies to surface and connectors that have the Match boundary condition. By using the Push Attribute, the spacing of the inlet and outlet connectors are matched to the spacing of the adjoining volumetric points.

4.2 Volume Grid Generation

After generating the meshes for surface domains, the computational domain should be discretized with different types of cells. At first, the computational domain, which is a three-

dimensional space of interest to the research, is created by combining the surface domains into a fully enclosed volume. Domains and the face orientation are selected to generate the block as shown in the Fig. 4.4. All of the surfaces have been highlighted to define the boundary of the block and the face orientation has been chosen to point inside the tunnel.

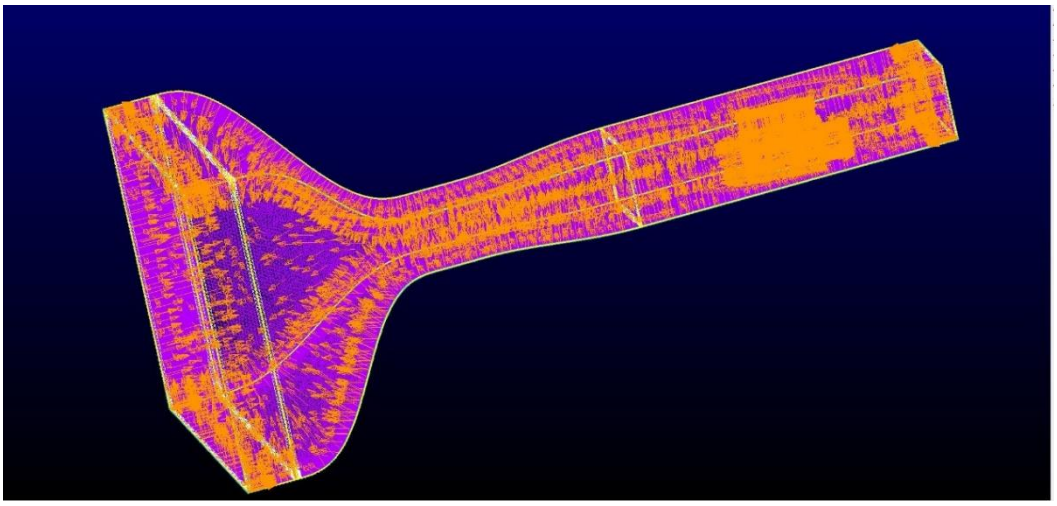


Figure 4.4: The Generation of Block for Volume Grids

The grid is built by generating anisotropic cells (T-Rex layers) with a specified initial spacing and a constant growth rate to create the appropriate viscous layers and then using isotropic tetrahedral cells to occupy the remaining space in the volume.

The T-Rex layers are generated by piling up the anisotropic cells such as prisms and pyramids elements normal to the domain in order to predict the boundary layer accurately. Those anisotropic cells are continuously produced through a specific initial spacing and a constant growth rate until the maximum layers are reached. The initial spacing Δy is determined based on flat-plate boundary layer theory from White's Fluid Mechanic [69].

$$\mathrm{Re} = \frac{\rho U_\infty L}{\mu}, \quad C_f = \frac{0.026}{\mathrm{Re}^{1/7}}, \quad \tau_{wall} = \frac{C_f \rho U_\infty^{\,2}}{2}, \quad U_{fric} = \sqrt{\frac{\tau_{wall}}{\rho}} \tag{4.1}$$

$$\Delta y = \frac{Y^+ \mu}{U_{fric} \rho} \tag{4.2}$$

In the above equation, Y^+ value is a non-dimensional normal distance to a wall which is defined as

$$Y^+ = \frac{U_{fric} y}{\mu} \tag{4.3}$$

where U_{fric} is the friction velocity at the nearest wall, y is the distance to the nearest wall, and μ is the local kinematic viscosity of the fluid. Y^+ value dictates the resolution level in the grids convergence study. Because the friction velocity which is unknown before solving the flow is used to calculate Y^+ value, the real Y^+ value requires an iterative approach that is difficult to achieve. For simplicity, $Y^+ = \frac{4}{9}$ is recommended to get a acceptable grid resolution level. Based on the physical wind tunnel experiments, the supersonic flow parameters are provided as follows:

$$\begin{aligned} &\rho = 0.9525\ kg/m^3, \quad U_\infty = 514.0068\ m/s, \quad a = 257.0034\ m/s \\ &L = 0.072\ m, \quad T = 164.44\ K, \quad P = 4.4940 \times 10^4\ Pa \\ &\mu = 1.846 \times 10^{-5}\ kg/(m \cdot s) \end{aligned} \tag{4.4}$$

Substituting the Y^+ value and the supersonic flow parameters, the Reynolds number and the initial spacing Δy can be determined as

$$\begin{aligned} &\mathrm{Re} = \frac{\rho U_\infty L}{\mu} = 1.909565 \times 10^6 \\ &\Delta y = 4.12935 \times 10^{-7}\ m = 1.6275 \times 10^{-5}\ in \end{aligned} \tag{4.5}$$

In this research, the initial spacing is chosen as 1.0×10^{-5} *inches*, which is slightly smaller than the calculation to get more layers in the viscous sub-layer region. The growth rate is required to be no larger than 1.25 to prevent layers from over expanding. The height of the cell is 1,000 times larger

than that of the initial cell after thirty layers when using the growth rate $GR_0 = 1.25$. The appropriate growth rate to improve grid quality can be decided based on the expression

$$Growth\ Rate = GR_0^{\frac{1}{F^n}} \tag{4.6}$$

where $F \approx 1.5$ and $n = 2$ are used for this study of a fine grid level. Thus, the growth rate used in this research is equal to 1.1. The boundary conditions and the initial spacing Δy for generating T-Rex layers in the volume grid is presented in the Table 4.1. Moreover, the advanced criteria for T-Rex layers are stated in the Table 4.2.

T-Rex Layers Criteria	**Value**
The Growth Rate	1.1
Max Layers	100
Isotropic Seed Layers	2.0
Collision Buffer	2.0
Aniso-Iso Blend	1.0
Maximum Angle	160
Centroid	0.7

Table 4.2: Advanced Criteria for T-Rex Layers Generation in Volume Grids

The T-Rex Layers are stopped at the location where either the maximum layers are reached, or the anisotropic cells violate the advanced criteria. The Push Attribute is also involved in volume grid generation so that the T-Rex layers could grow correctly near the inlet of the nozzle and the outlet of the test section. Figure 4.5 shows the difference between the grids with and without the Push Attribute. The Push Attribute in the volume grid generation will change the surface meshes to allow for the creation of the anisotropic cells.

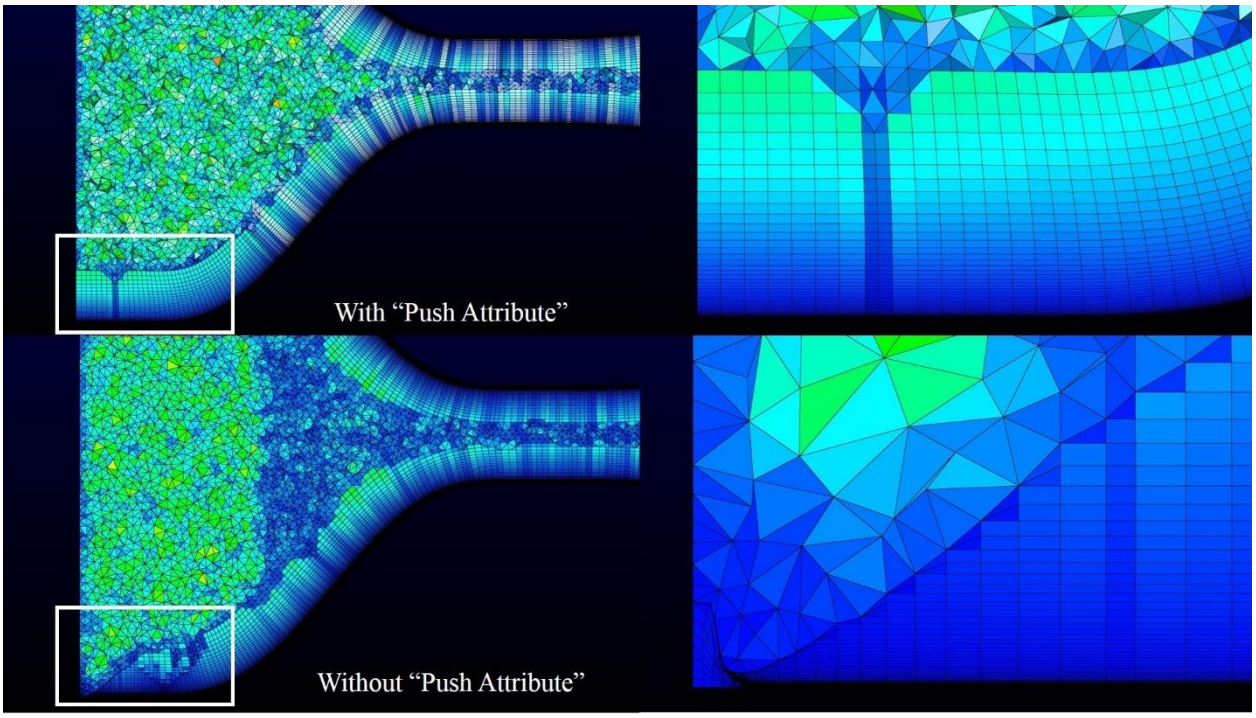


Figure 4.5: The Inlet Portion Grids with/without the Push Attribute

After the generation of anisotropic cells (T-Rex layers), the tetrahedral cells are used to populate the remaining space. The size of the tetrahedral cells in the test section (the area of interest) are constrained through the "Sources" tool, which helps users to specify value to the maximum size of elements in a selected region. Figure 4.6 shows the source being used to define the supersonic region and to control the size of tetrahedral cells inside of the source at a fixed spacing of 0.1 inch. This method ensures the flow features in the interest area has a good resolution.

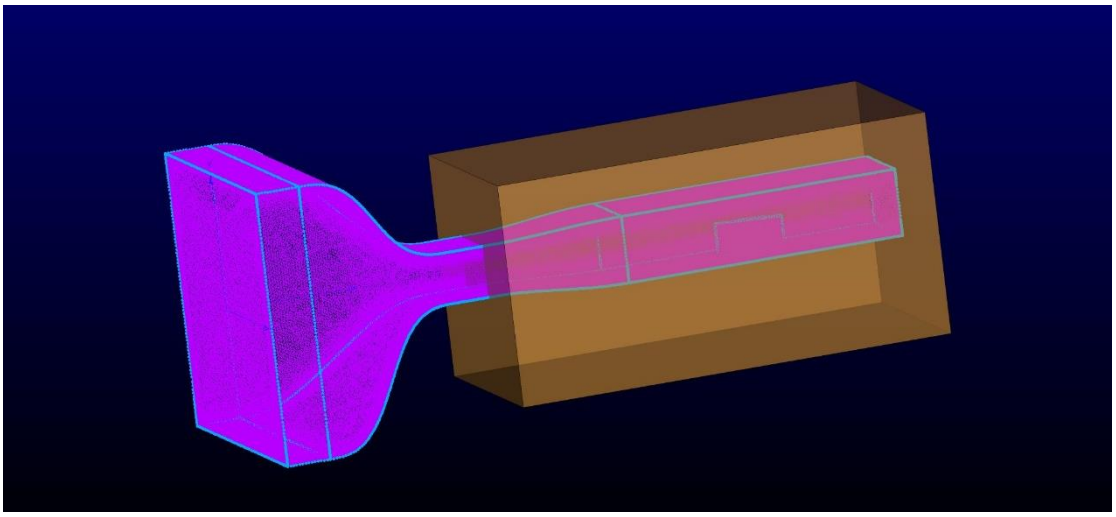


Figure 4.6: The Application of Source for Better Control on Isotropic Cells' Size

The volume grids are achieved through utilizing a combination of different types of elements, which are prisms, pyramids, and tetrahedral elements. Figure 4.7 shows the different cells distributing around the fin, where green elements are prisms, yellow elements are pyramids, and red elements are tetrahedrals. Figure 4.8 indicates the transformation from prisms and pyramids to tetrahedral cells at the leading blunted head. Pyramids are used to connect the T-Rex-layer extrusion (prisms) and tetrahedral cells.

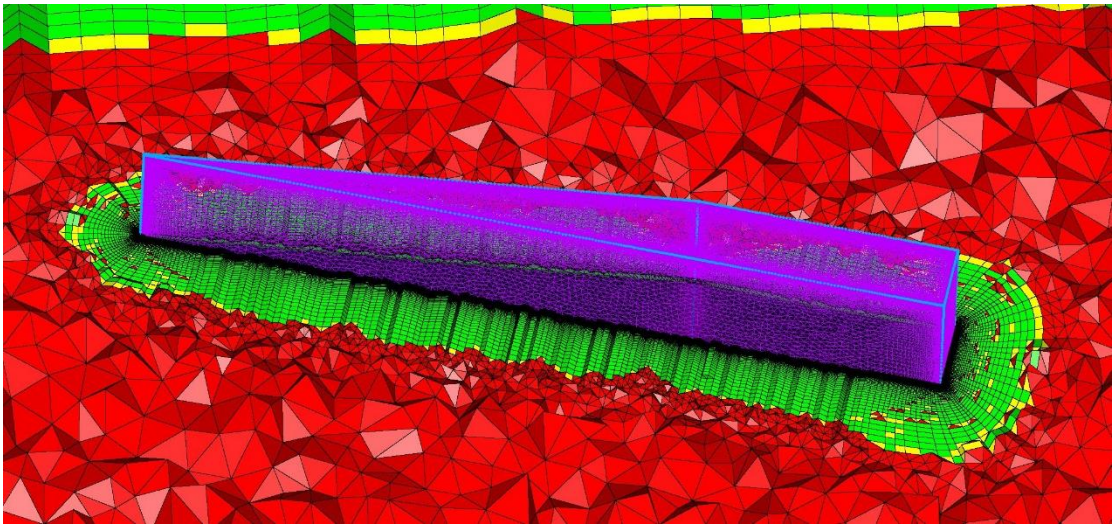


Figure 4.7: Multi-Element Grid with Prisms, Pyramids, and Tetrahedral Cells

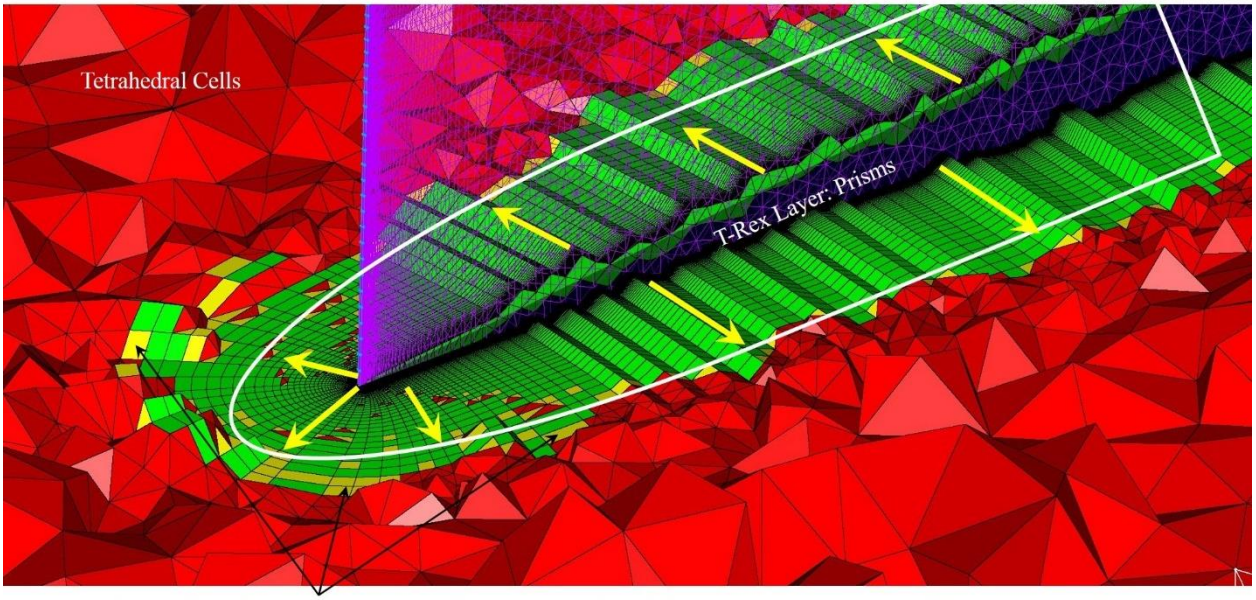


Figure 4.8: The Transformation from Anisotropic Cells to Isotropic Cells at the Leading Blunted Head in the Grid Generation

Notice that a farfield case model is also established using the grid generation procedures and parameters introduced in this section. The farfield case arranges the fin on the center of the bottom

surface of a box with the size of $2m \times 2m \times 1m$. The number of total nodes and different cells in the volume grid has been tabulated in the Table 4.3.

Model	Nodes	Prisms	Pyramids	Tetrahedrals	Total Cells
In-Tunnel	21,927,844	41,869,502	524,390	3,853,837	46,247,729
Farfield	13,170,581	15,526,687	254,725	31,649,493	47,430,905

Table 4.3: Total Number of Nodes and Different Type of Cells

4.3 Grid Export

After generating the volume grid successfully, Pointwise allows users to export the grid into an SG file that is acceptable by the Tenasi solver. The domains have been identified before export so that the Tenasi solver could apply different boundary conditions to those surfaces in the numerical solving process. Domains are divided into inlet, outlet, fin, mounting plate, and sidewalls; furthermore, appropriate boundary conditions are written in a boundary condition file.

4.4 Grid Quality

A high grid quality will significantly influence the numerical simulation. So, it is necessary to evaluate the grids generated and make some modifications to the grids before applying them into the flow solver. The quality of volume grids depends on the quality of domain meshes extruded to create blocks and the parameters used to restrict volume grid generation. The absolute definition of grid quality level is virtually absent; nevertheless, high-quality grids reveal all of the structural

characteristics of the model to maintain a maximum accuracy for solutions and to economize the numerical computation cost and time. Several important parameters used to examine the quality of grids are area ratio, minimum included angle, maximum included angle, aspect ratio, and centroid skewness.

The area ratio is the ratio of areas between the neighboring cells. Since the generation of a volume gird is achieved by inserting points normal to the domain with a distance decided by initial spacing and growth rate, the large area elements in the domain will extrude significantly faster than the small area elements, which can generate skewed cells to connect them. Generally, in order to forestall the high skewness, an area ratio up to 5 is acceptable in the tetrahedral dominant region and values up to 10 is acceptable in the prismatic dominant region. However, skewed cells always exist because small differences between elements increase exponentially through iterations. The area ratio of fin and test section domains have been demonstrated in the Fig. 4.9, and the maximum area ratio occurs in the outlet domain because of matching the extrusion of sidewalls. The inlet and outlet domain show an “X-shape” mesh because the four sidewalls use the Push Attribute to extrude normally but the extrusion stops when overlapping the extrusion in another direction as shown in the Fig. 4.10.

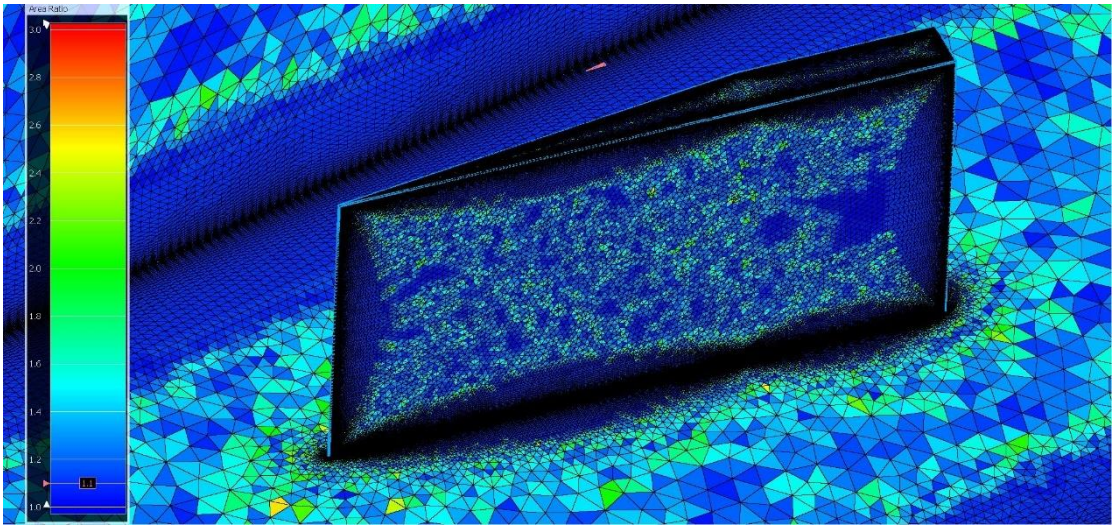


Figure 4.9: The Area Ratio of Elements around the Fin

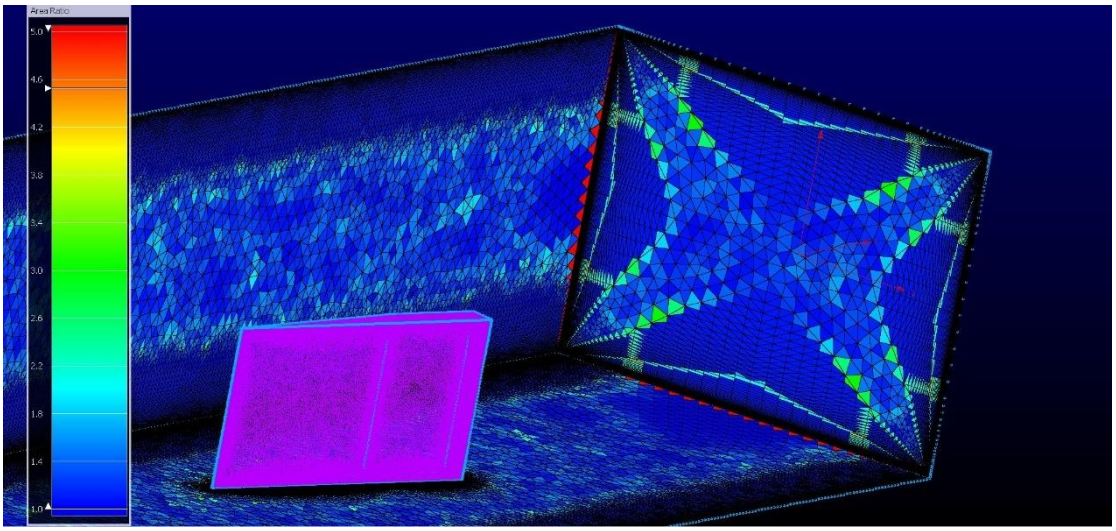


Figure 4.10: Large Area Ratio in the Elements Connecting the Sidewalls and Outlet Domain

Maximum and minimum included angle are the maximum angle and minimum angle in the grid cells, respectively. The maximum included angle in surface meshes is often controlled under 150° and the minimum included angle that is greater than 2° is acceptable. The difficult part of meshing surfaces and generating volume grids is using different parameter combinations to achieve a reasonable minimum angle in the sharp corner (leading edge of the fin). The supersonic

case is always problematic in CFD because the large gradient of flow properties need strict discretization to guarantee the validity in the iterative process. Thin boundary layers in supersonic simulations require a very small initial spacing in extruding the T-Rex layers (boundaries); however, over-segmentation along the sidewalls increases the computational time significantly so using the same magnitude value of initial spacing to decompose the connectors is meaningless. Pointwise can split the connectors and recombine them to decrease the maximum angle if the angle in cells is extremely large (larger than 179°), but this approach does not always work and leads to the failure of the grid generation. When meshing the domains, the maximum included angle is about 159° while the minimum included angle is about 1°, which are demonstrated in Fig. 4.11. The minimum angle is located at the middle of the leading blunted head of the fin. When generating the volume grids, the maximum included angle is about 179°, which happens near the bottom corner of the fin and the corner of sidewalls. This is caused by the difference between the initial spacing of T-Rex and average spacing of connectors in addition to the termination of the T-Rex extrusion in two overlapping directions. Figure 4.12 shows large angle cells distributed around the rear bottom corner of the fin.

Parameters	**Domain meshes**	**Volume grids**
Maximum included angle	159.009	179.171
Minimum included angle	1.019	0.013

Table 4.4: The Maximum and Minimum Included Angle in the Grid Generation

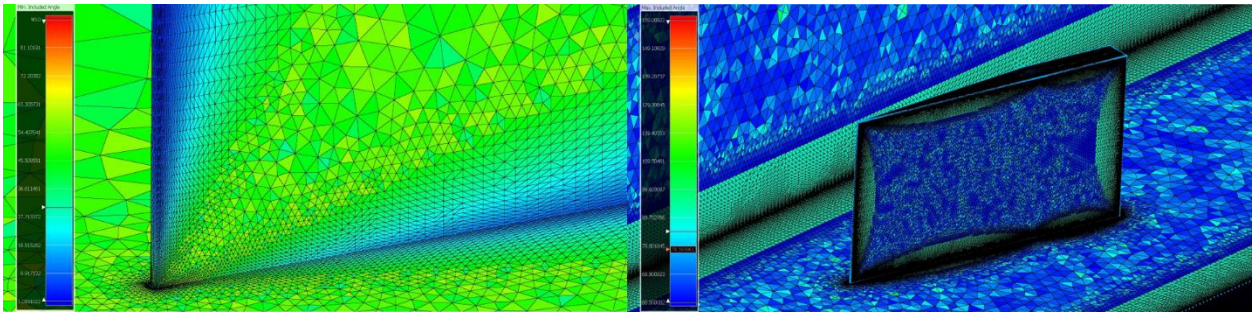


Figure 4.11: The Maximum (Left) and Minimum (Right) Included Angle in the Domain Meshes

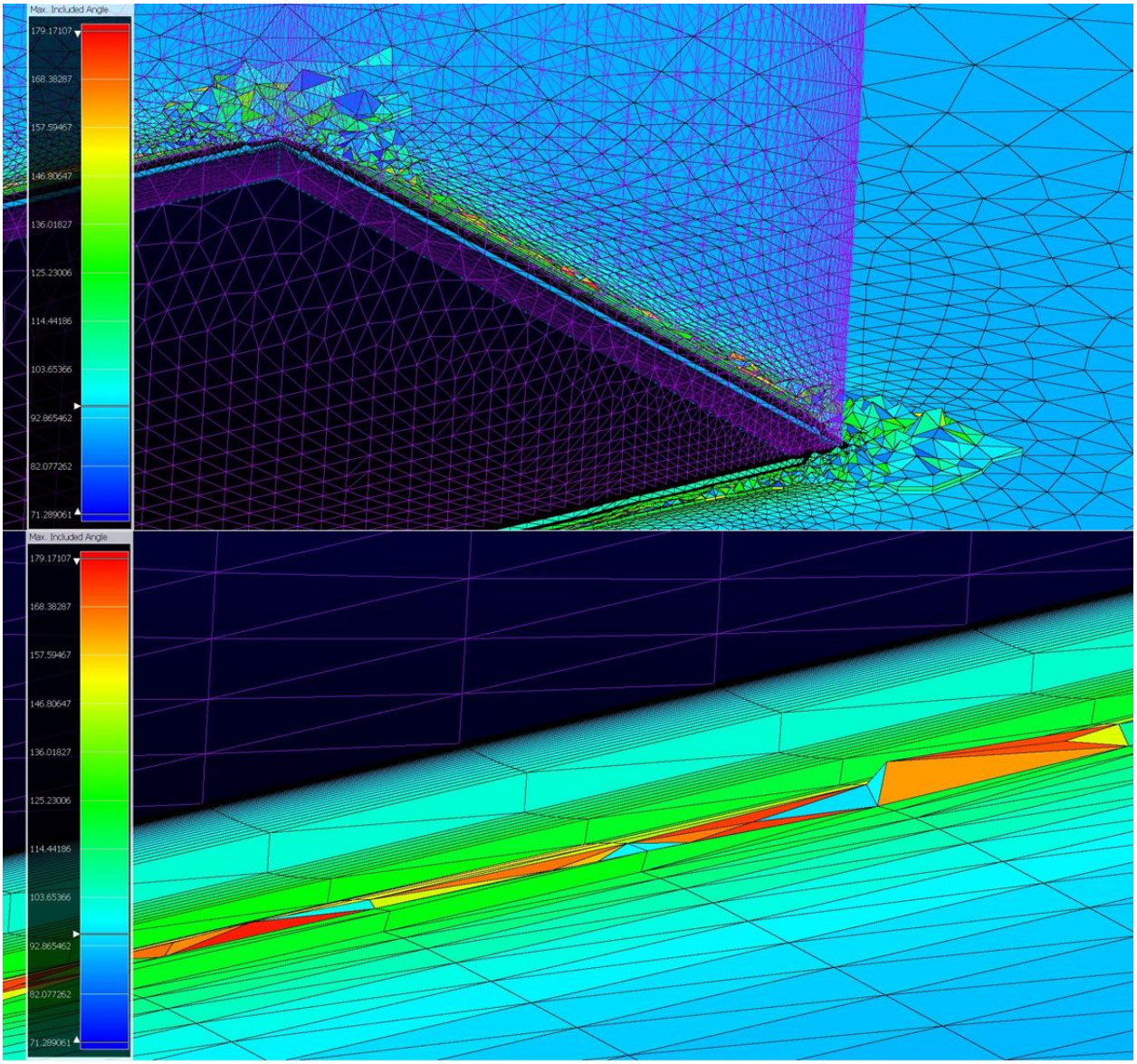


Figure 4.12: The Distribution of Large Angle Cells around the Rear Bottom Corner of the Fin.

Another quality criteria, the centroid skewness is defined as

$$Centroid\ \ Skewness = 1 - \max\left(\frac{\vec{A}_i \cdot \vec{f}_i}{\left|\vec{A}_i\right|\left|\vec{f}_i\right|}\right) \tag{4.7}$$

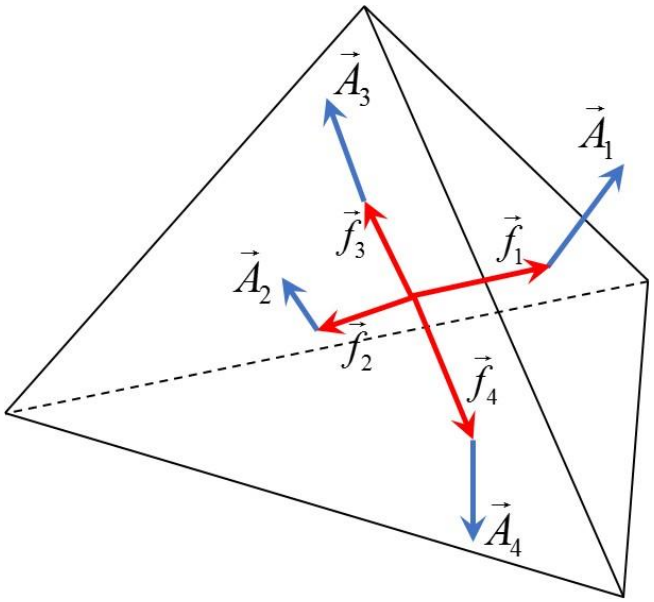


Figure 4.13 Illustration of Cell Face Normal Vectors and Vectors from the Cell Centroid to the Face Centroid

where $\vec{A}_i$ is the cell face normal vector and $\vec{f}_i$ is a vector connecting the cell centroid and the face centroid (Fig. 4.13). Values range from 0 (no-skew) to 1 (collapsed cell). It is recommended to keep the value below 0.8 for good grids and sometimes the value below 0.95 is acceptable. As shown in Fig 4.14, the centroid skewness of most of the cells is under 0.5. The extremely large centroid skewness values belong to cells situated at the corner of sidewalls. Similar to domain meshes, during the volume grid generation, if the anisotropic cell extruded normal to the domain overlaps another extrusion, the anisotropic cell layer (T-Rex layer) stops locally, and isotropic cells fill the gap that has been squeezed to elongated tetrahedral cells.

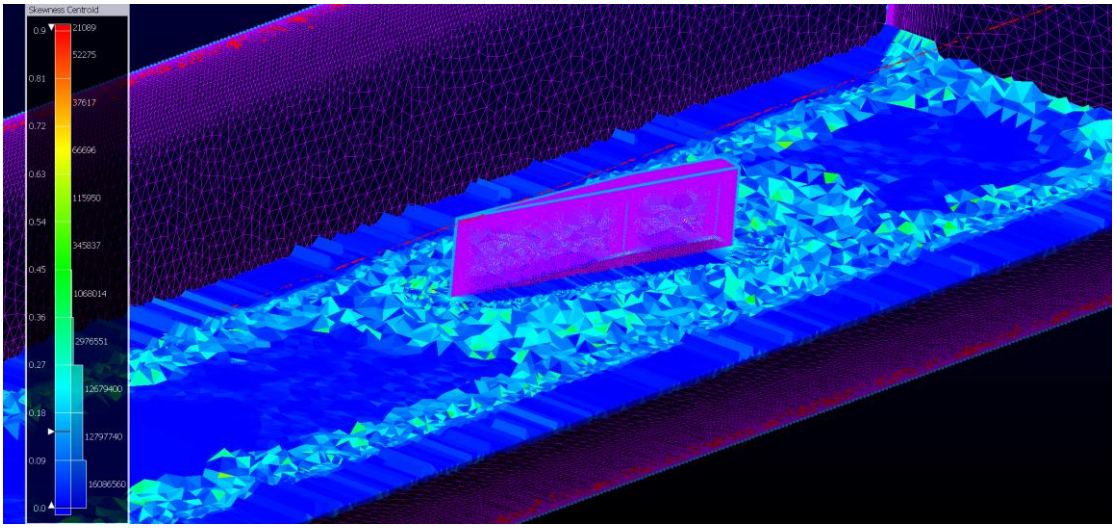


Figure 4.14: The Distribution of Skewed Cells

Through the discussion of grid quality, the grids created for this research are of high quality. The area ratio in domain meshes is below 5, which is reasonable. Most of the control volumes have a maximum included angle under 178° while 30 volumes in the grid have a maximum included angle greater than 178.5°. However, some highly skewed cells occur around the intersection of sidewalls which have not been found to adversely affect the simulations, and the number of cells that have the centroid skewness greater than 0.9 just occupy a very small percentage (0.046%) of the entire volume mesh. Hence, the grids are considered to be of high quality.

Chapter 5

Numerical Simulation

After building the volumetric grid, the processes of numerical simulations are demonstrated in this chapter. The numerical parameters setup and boundary conditions are presented and followed by the discussion of the simulation details in different cases. The CFD solver calculates the discretized governing equations iteratively, and the numerical simulations are considered to be achieved when the calculations are convergent after numbers of iterations. Nevertheless, as mentioned in the previous chapter, the absolute convergence is rarely achieved in most cases. Thus, the valid numerical simulation result is obtained when the residuals are either stable over a long iteration period or under an acceptable tolerance.

5.1 Numerical Simulation Parameters Setup

The Tenasi solver is employed to solve the Reynolds-averaged Navier-Stokes equations (RANS) and some parameters must be specified. The numerical simulation parameters setup is an important procedure because an appropriate combination of the parameter set not only determines the nature of the simulations, but also improves the accuracy of simulation results. Generally, the parameters involved in the simulation are classified into five categories: Flow conditions, Solution

algorithm, Turbulence model, Run control, and Postprocessing.

The Supersonic flow conditions used in the numerical simulations are provided as hereunder:

- Reference density: $0.9525\ kg/m^3$
- Reference velocity: $514.0068\ m/s$ (also the velocity of the flow at infinity conditions)
- Reference length: $0.072\ m$
- Reference temperature: $164.44\ K$
- Reference pressure: $4.4940\times10^4\ N/m^2$
- Reference molecular viscosity: $1.846\times10^{-5}\ kg/(m\cdot s)$

Some of the important parameters used for solving the governing equations (Eq. 2.1) in the Tenasi flow solver are listed as follows:

- Flow regime: *Arbitrary Mach*
- Solution algorithm: *Implicit*
- Number of sweeps in the linear solution algorithm: *10*
- Inviscid flux Jacobian computation method: *Approximate Jacobians*
- Temporal accuracy (for time accurate simulation): *2nd order*
- Number of Newton steps per time step: *5*
- Inviscid flux computation method: *HLLC discretization*
- High order inviscid flux controls: *Venkatakrishnan limiter*
- WENO higher-order solutions: *7th order*
- Viscous gradients computation: *Weighted least square*

The simulation utilizes both Wilcox $k-\omega$ model and the blended $k-\epsilon, k-\omega$ turbulence model. The detail about the turbulence model parameters have been listed below respectively.

Wilcox $k-\omega$ model:

- Number of linear sub-iterations: *10*
- Applies a mean field generation source term for use with aerodynamic shocks: *yes*
- Uses a large-eddy simulation variation of the turbulence model: *yes*
- k value for the run-time initialization: *1.0e-7*
- ω value for the run-time initialization: *10.0*

Blended $k-\epsilon, k-\omega$ turbulence model:

- Number of linear sub-iterations: *10*
- Applies a mean field generation source term for use with aerodynamic shocks: *yes*
- Applies the compressibility correction to the entire flow field: *yes*
- Applies the pressure dilatation term to the entire flow field: *yes*
- k value for the run-time initialization: *1.0e-7*
- ω value for the run-time initialization: *10.0*
- Coefficient for the mean field generation term: *1.0*
- Sets force function F_1 : *1.0*

5.2 Boundary Conditions

For the in-tunnel cases, the sidewalls of the converging-diverging nozzle, the sidewalls of the test section and the fin surfaces are set as non-slip, viscous, adiabatic walls. The flow direction is set along the x-axis. The total pressure of 351633.0 Pa and the total temperature of 296 K is defined as the inlet condition and the back pressure at the outlet is 44940.3 Pa.

For the non-obstacle surrounding cases, the bottom surface and the fin surface are set as non-slip, viscous, adiabatic walls. The remaining surfaces are set as the farfield. The flow direction is set along the x-axis.

5.3 Simulation Details

Most of the simulation cases in this study have over 20 million nodes, which needs a massive computation power and can be only achieved by operating in super computers. For this research, the numerical simulations are performed using High Performance Computation Linux Clusters (HPC) found in Auburn University known as Hopper. Hopper cluster offers 191 "Standard" nodes, and each node has 128GB of memory and 2 "Haswell" CPUs (10 cores/CPU). The grids are decomposed such that each core receives approximately 200,000 grid points in order to keep an optimal run speed.

The numerical simulations using the AMN regime performs steady state solutions to establish the basic flow field and then time accurate solutions to capture unsteady phenomena. In the operation process of the steady state, the simulation uses the first order iterations (flux

computations) to prevent instability followed by the higher order flux evaluations to improve the solution. Compared with the higher order schemes, the lower order schemes run the flow field with high dissipation and quickly generate an approximate mean flow. For this reason, the lower order iterations create a reasonable flow field at the beginning and offer a general flow field solution for improved higher-order iterations. The temporal accuracy is used after the steady state solutions establish the flow field, and then second order accuracy in time using five Newton iterations with 10 linear subiterations is utilized in the simulations. For unsteady simulations, time steps are $10^{-7}\ s$. Overall, the simulation process of each of the cases could be separated into five portions: first order steady state simulation, second order steady state simulation, first order temporal accuracy (unsteady state) simulation, second order temporal accuracy (unsteady state) simulation, and WENO higher order solutions simulation, which has been demonstrated in Table 5.1. The convergence history for three numerical simulation cases has been presented in Fig. 5.1, which shows the residual of all three cases is smaller than 10^{-3} and runs stably in the WENO higher-order solutions process. The Wilcox $k-\omega$ turbulence shows a better stability compared with the Menter SST turbulence model because the switch criteria of the Menter SST needs more parameters to control, and inappropriate parameters lead a huge residual value in the iterative numerical computational processes. In order to show the feature of the supersonic flow, 5,000 iterations are used to time average data for the unsteady run. Notice that the Plenoptic PIV results are an average of 500 images taken at 15 frames per second (15 Hz), so around 30 seconds of run time. Furthermore, the camera frame rate used in the physical experiment is 0.5Hz so that the

averaged results are really the mean flowfield. This run time scale is unaccessible for numerical simulation because of the memory and time costs. In the CFD process, the numerical solutions are averaged for 5,000 iterations so the averaged result indicates the flow features in $5\times10^{-4}s$. Compared with the Plenoptic PIV results, the CFD results can be consider as an instantaneous result.

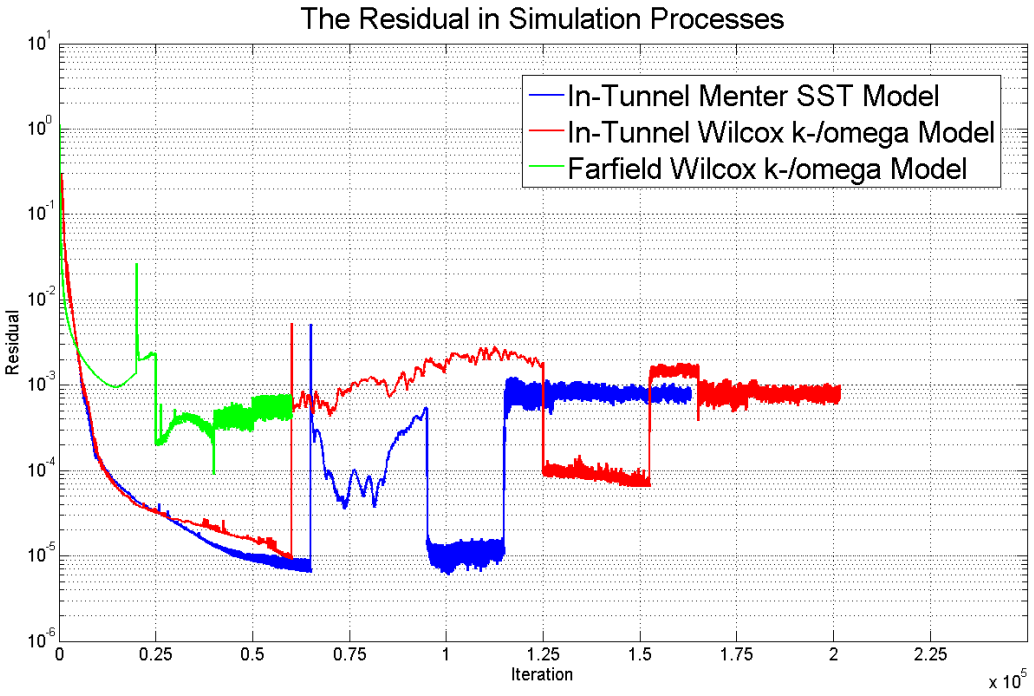


Figure 5.1: The Convergence Plots in the Different Simulation Cases

In the numerical simulation processes, there is an automatic CFL adjustment procedure for steady state simulations such that problems could be detected before arising, and the adjustment of the CFL number prevents the program from going unstable. In this study, CFL-START = 1.0, CFL-RAMP = 1,000, and CFL = 2.0, which means 1,000 time steps is used to ramp the CFL number linearly from 1.0 to 2.0. The details about runs are shown in Table 5.1 below

Details	**In-tunnel Cases**		**Farfield Cases**
Turbulence Model	Wilcox $k-\omega$	Menter SST	Wilcox $k-\omega$
Grids Size	$\approx 2.1\times10^{7}$	$\approx 2.1\times10^{7}$	$\approx 1.3\times10^{7}$
Number of Nodes (Cores) Used	9 (180)	9 (180)	9 (180)
CFL-START	1.0	1.0	1.0
CFL-RAMP	1,000	1,000	1,000
CFL	2.0	1.5	2.0
Time Step (s)	10^{-7}	10^{-7}	10^{-7}
1st Order Steady Iteration Number	60,000	65,000	20,000
2nd Order Steady Iteration Number	65,000	30,000	5,000
1st Order Unsteady Iteration Number	20,000	10,000	10,000
2nd Order Unsteady Iteration Number	7,5000	10,000	5,000
WENO Higher-Order Solutions Iteration Number	47,500	45,000	20,000

Table 5.1: Details about Numerical Simulation Processes for Different Cases

The simulation time (CPU time) is around 2-5 seconds per iteration in the steady state solutions, but this time might increase to 10-30 seconds because of Newton iterations to resolve unsteady features. The total simulation time for each case usually remains several days. However, there are some fluctuations in the CPU usage time because of the busy situation of the Hopper cluster. Heavy I/O operations among the users affect the network speed and operation speed of the computation node.

Chapter 6

Results and Discussion

The results from numerical simulations introduced in the previous chapters are analyzed by means of examining various flow characteristics, such as the velocity in the x-direction, the density, the total pressure, and the temperature. Furthermore, the velocity components from CFD simulation results are compared with the measured value of the wind tunnel experiments using the Plenoptic PIV method for validation.

FieldView version 16 is applied for the CFD post-processing and visualization. Meanwhile, for the requirement of the numerical analysis and comparison of simulation results, the data set is also extracted and processed in MATLAB. There are two methods for the data set extraction: (1) using the python script to extract results directly from the Tenasi flow solver and (2) exporting the data after the CFD post-possessing from FieldView. The data set for the analysis is extracted from a cubic region, which is identical to the region of used in the physical experiment. The cubic region has a height of 29mm, a width of 16mm, and a length of 45mm, correspondingly, and it is 6mm, 0mm, and 16mm away from the bottom point of the leading edge of the fin in the x, y, and z-axis, respectively. The location of the cubic region for the data extraction has been indicated in Fig. 6.1 (Left), and the planar oblique shock wave at a wave angle of approximate 45°passes through the

cubic region, which is shown in the Fig. 6.1 (Right). Notice that the original coordinate system of the data set from physical experiments is different than the coordinate system established in the numerical simulations, and the PIV data has been translated into the coordinate system of the numerical simulations for these comparisons.

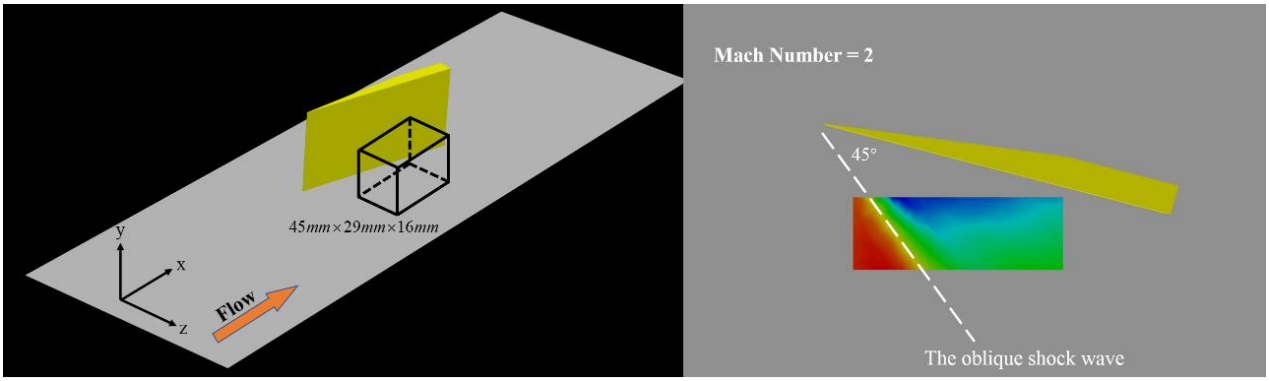


Figure 6.1: Data Analysis Region of the Numerical Simulation

The discrete data has been interpolated to create the figures of the flow properties in the MATLAB. The interpolation method employed in this study is a triangulation-based cubic interpolation, which has a continuity of C^2. From the numerical simulation of different cases, the feature of the λ-structure of the SWBLI is obtained, which is illuminated in Fig. 6.2. The velocity of the free stream reaches to $Mach \approx 2$, which is roughly $510\ m/s$. Because of the obstruction of the fin, the oblique shock wave is generated, and the separation line is curved to the downstream. The green portion is the transonic region. It can be observed in Fig. 6.2 that the boundary layer is very thin in the supersonic condition, that the thickness of the boundary layer is altered because of the oblique shock wave. The main oblique shock wave, the separation shock wave, and the rear shock wave encounter at the triple point forming the λ-structure of the SWBLI. The downstream flow

impinges the perpendicular surface of the fin and deforms the sonic line at the intersection of the fin and the bottom surface.

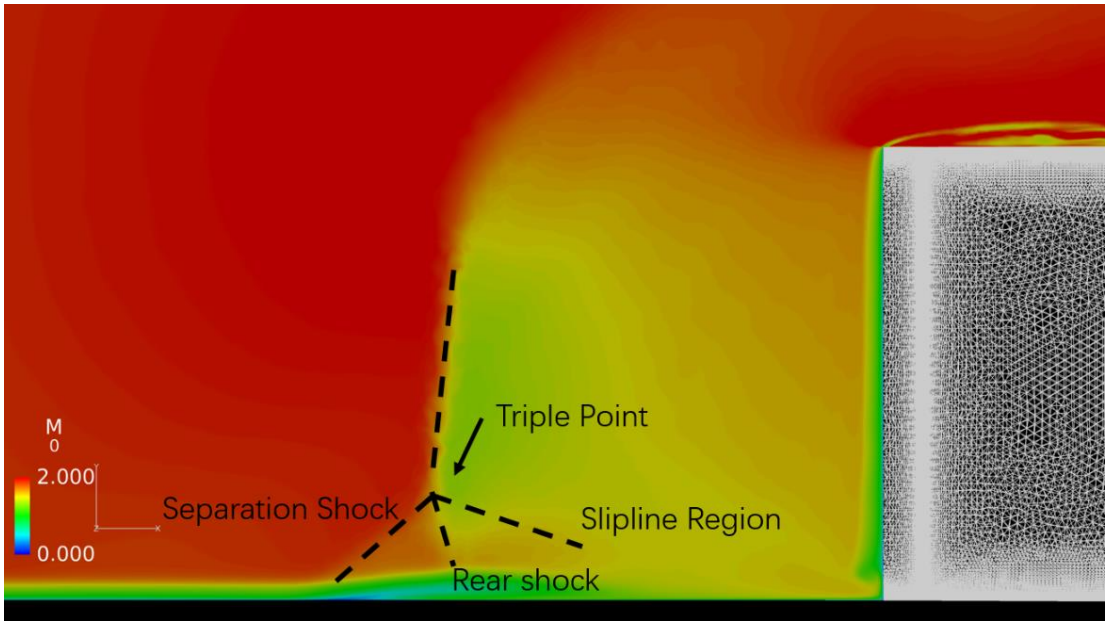
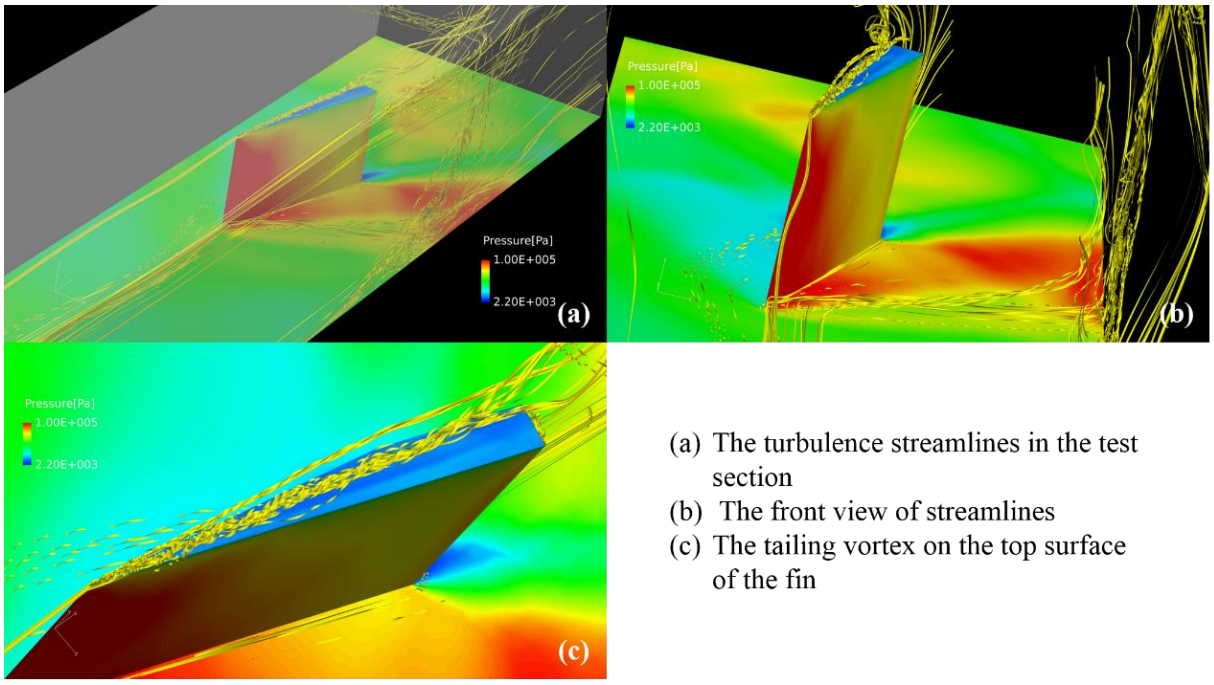


Figure 6.2: The Demonstration of the λ -Structure of the SWBLI from the Numerical Simulation (Location: Z=0)

6.1 In-Tunnel Case Using Wilcox $k-\omega$ Turbulence Model

Figure 6.3 shows the streamlines of the supersonic flow in the test section. The tailing vortexes form at the leading edge and on the top surface of the fin. The vortexes occurring and being constrained in the region near the rear surface of the fin incessantly rotate dragging the model. Because of the oblique shock wave, the stream lines are forced to turn toward the right sidewall of the test section. Since the shock wave impinges and affects the boundary layer of the sidewall, the flow near the right sidewall of the tunnel rises to the top surface of the tunnel and vortexes are generated and intermingled together. The downstream of the oblique shock wave is a high-pressure

region.



(a) The turbulence streamlines in the test section
(b) The front view of streamlines
(c) The tailing vortex on the top surface of the fin

Figure 6.3: Supersonic Flow Streamlines in the Test Section Using Wilcox $k-\omega$ Model

By using Wilcox $k-\omega$ model, the 3D structure of the SWBLI generated by the fin is demonstrated by means of the velocity iso-surfaces in Fig. 6.4. The velocity of Mach 2 flow is $514\ m/s$. However, since the shock wave is a discontinuity causing high gradient of flow properties, it is difficult to capture the iso-surface of the shock wave by setting the velocity equal to $514\ m/s$. Rather, a smaller value of velocity is selected to represent the oblique shock wave. The λ-structure is shown in Fig 6.4 by three color iso-surfaces where the main oblique shock wave and separation shock wave, the rear shock wave, and the slipline region are presented by the red, yellow, and green iso-surface, respectively. The three iso-surfaces separate at the triple point, and the portion under the separation shock wave and the rear shock wave is the separation region. With the shock wave propagation away from the fin toward the right sidewall of the tunnel, the

size of the λ -structure increases.

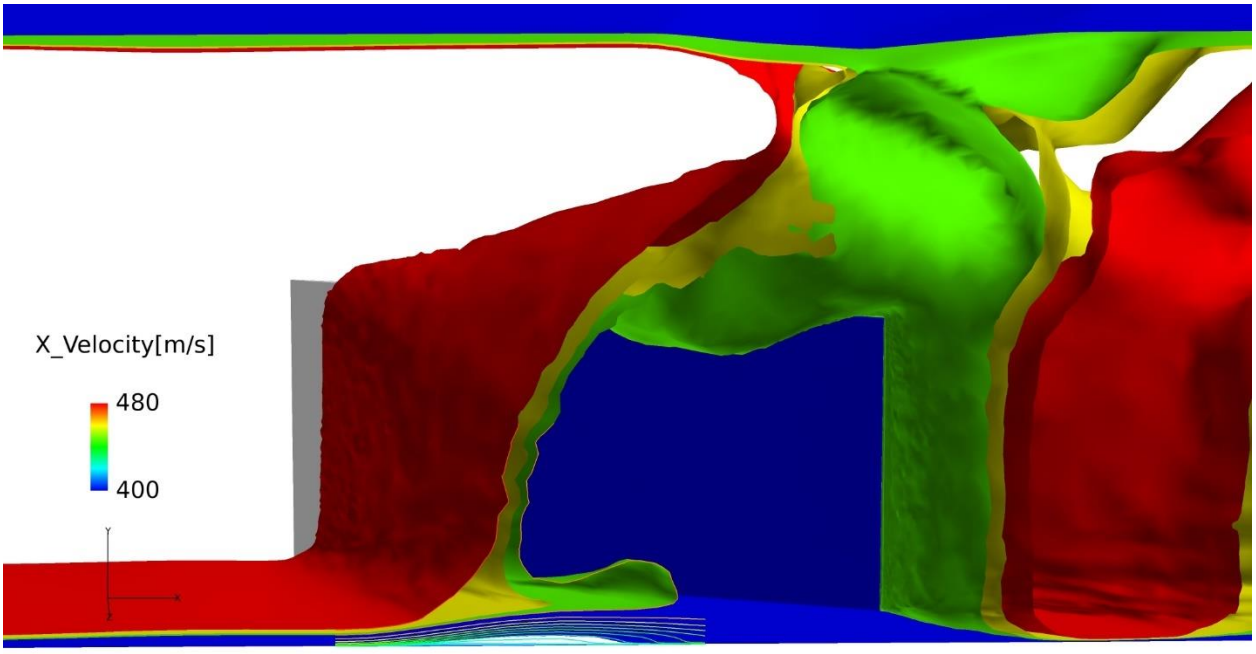


Figure 6.4: The Structure of the SWBLI Generated by the Fin in the Test Section Using Wilcox $k-\omega$ Model (Iso-Surfaces of X-Velocity)

Distributions of the different flow characteristics are provided by the slice cut figures in the z-direction in order to show the growth process of the SWBLI in the in-tunnel case using Wilcox $k-\omega$ model. The distribution of the x-direction component of the velocity, the total pressure, the density, and the temperature are presented in Fig 6.5, Fig. 6.6, Fig. 6.7, Fig 6.8, Fig. 6.9, Fig. 6.10, Fig 6.11, Fig. 6.12, Fig. 6.13, Fig 6.14, Fig. 6.15, and Fig. 6.16, respectively. The x-velocity distribution shown in Fig. 6.5, Fig 6.6, and Fig. 6.7 provides a good illustration of the development of the SWBLI structure, which demonstrates the behavior shown in Fig. 6.4. The boundary of the dark red region on the left and the orange region on the right represents the main shock wave and

the separation shock wave. The region downstream of the main shock wave is a low-velocity region which is terminated by the rear shock wave and slip surfaces generated by the SWBLI. The separation region increases and the slipline region extends backward with the increase of the z location. The thickness of the boundary behind the shock wave increases significantly and changes back to the thickness upstream of the shock wave within 40mm.

The total pressure distribution presented in Fig. 6.8, Fig. 6.9, and Fig. 6.10 shows a high-pressure region behind the main shock wave. The separation bubble in the separation region is obvious, and a pressure "jet" occurs from the triple point to the bottom along the separation shock wave. With the development of the separation region in z-axis, the separation bubble is enlarged.

The density distribution shown in Fig. 6.11, Fig. 6.12, Fig. 6.13 indicates a high-density region behind the main shock wave. An extreme low-density region, which is related to the extension of the slipline region, is located near the mounting plate and increases along the x-direction.

The temperature distribution presented in Fig. 6.14, Fig. 6.15, and Fig. 6.16 shows a high-temperature region behind the main shock wave. More important, an extreme high-temperature region occurs in the subsonic region near the bottom mounting plate which illustrates the thermal loads on the aircraft's surface caused by the SWBLI. It is also shown in Fig 6.14, Fig. 6.15, and Fig. 6.16 that the extreme high-temperature region is separated into an upper portion and a lower portion by a relatively low-temperature layer (the yellow line in Fig 6.14, Fig. 6.15, and Fig. 6.16).

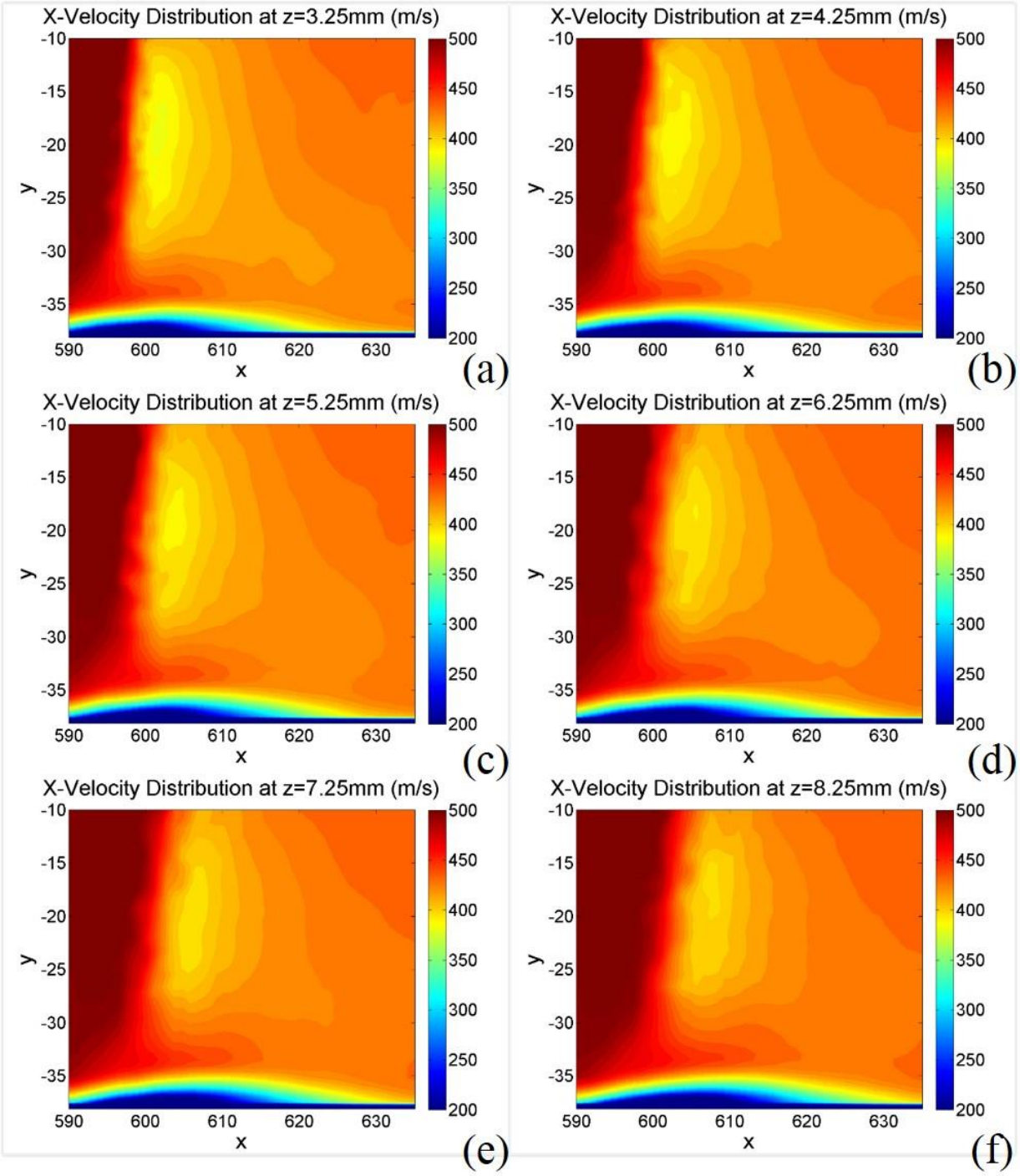


Figure 6.5: X-Velocity Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

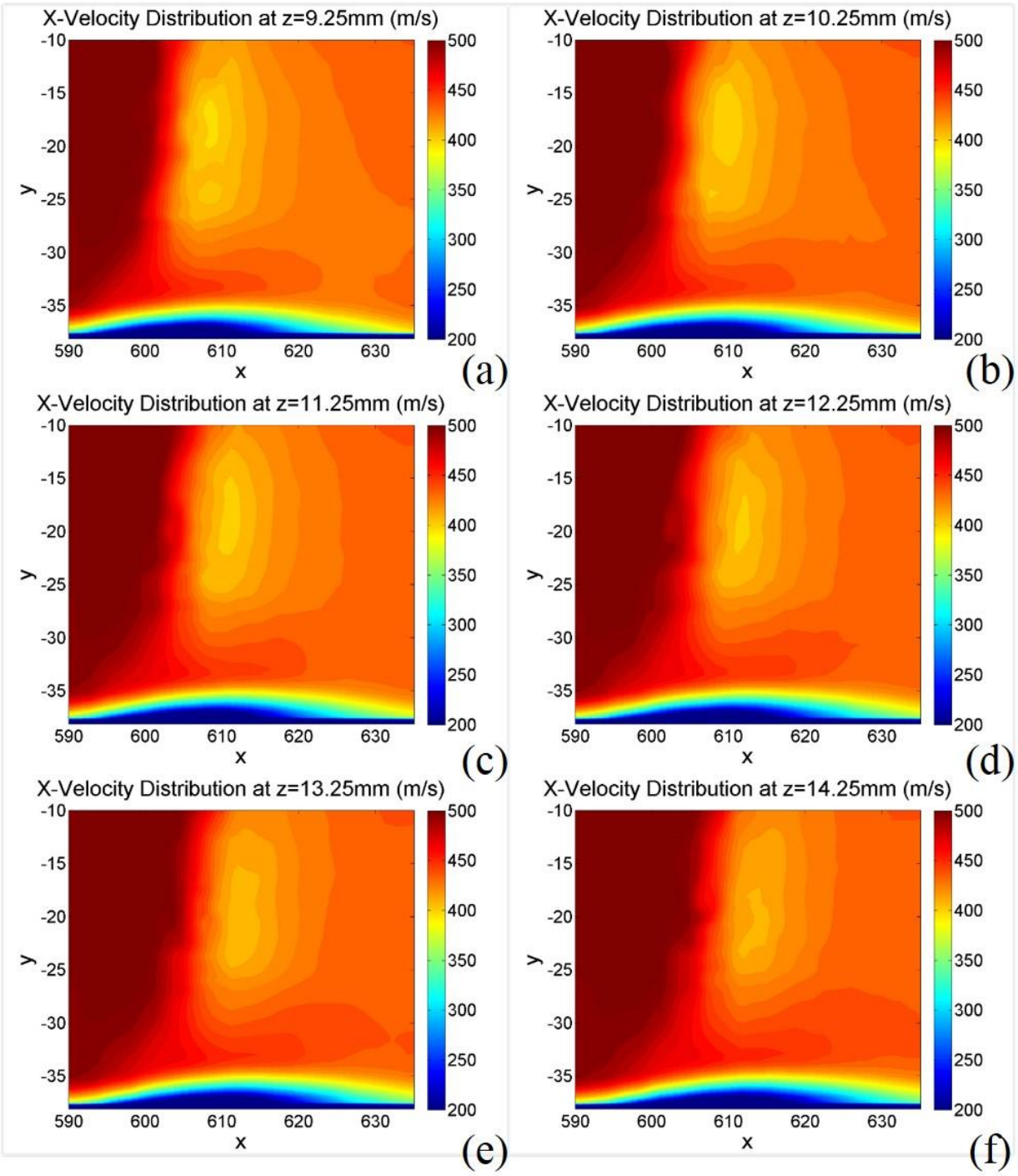


Figure 6.6: X-Velocity Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

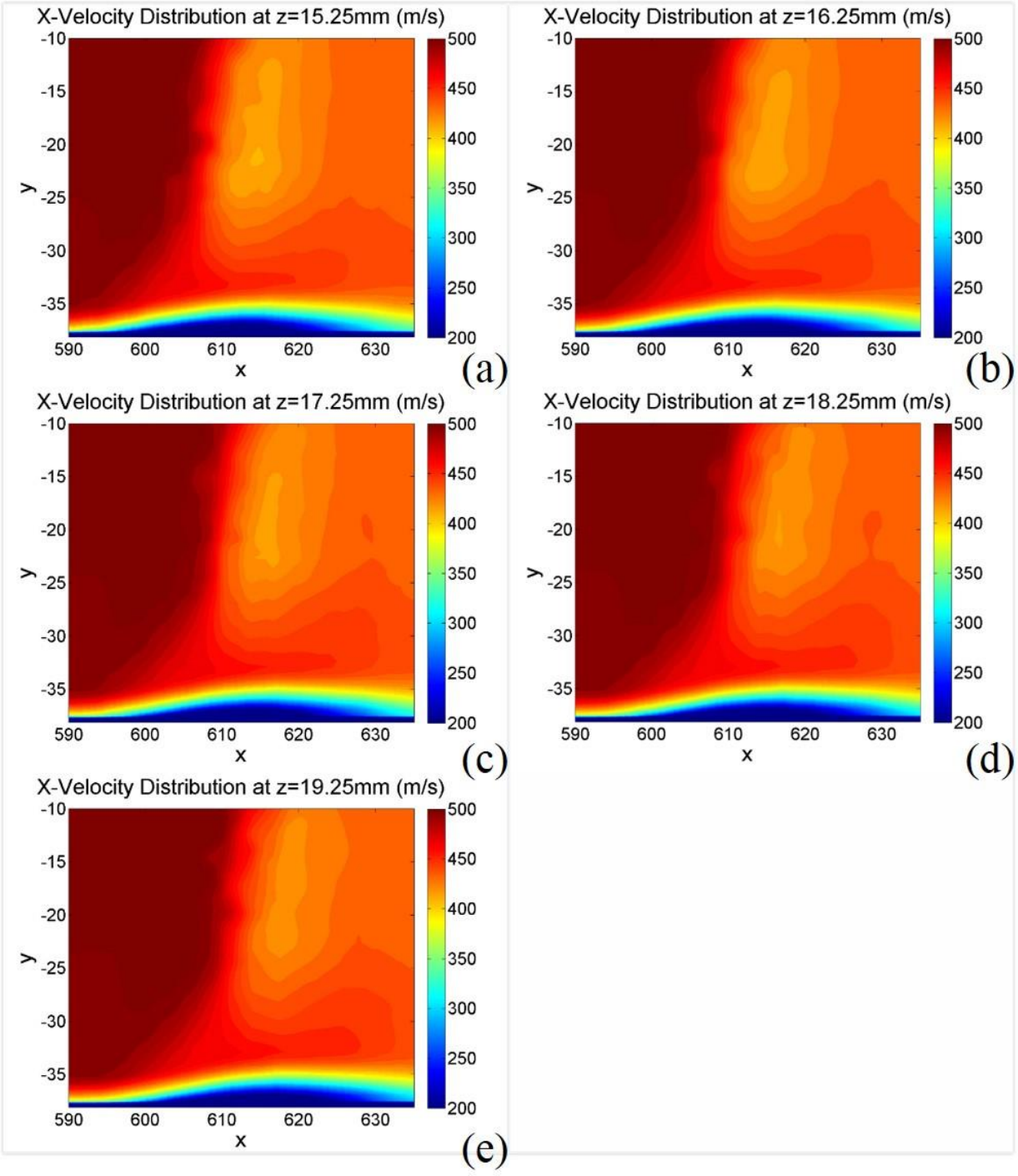


Figure 6.7: X-Velocity Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

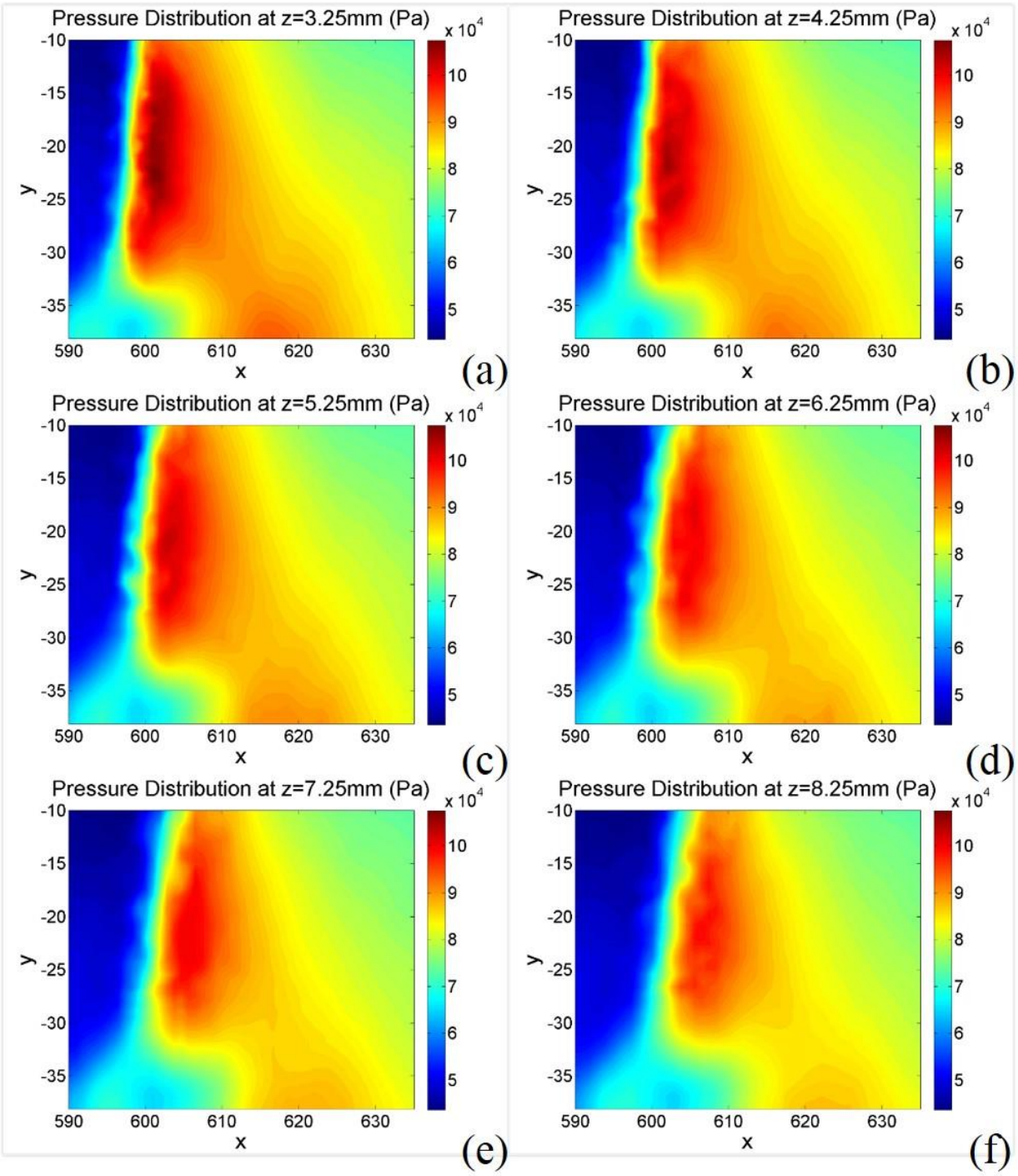


Figure 6.8: Pressure Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

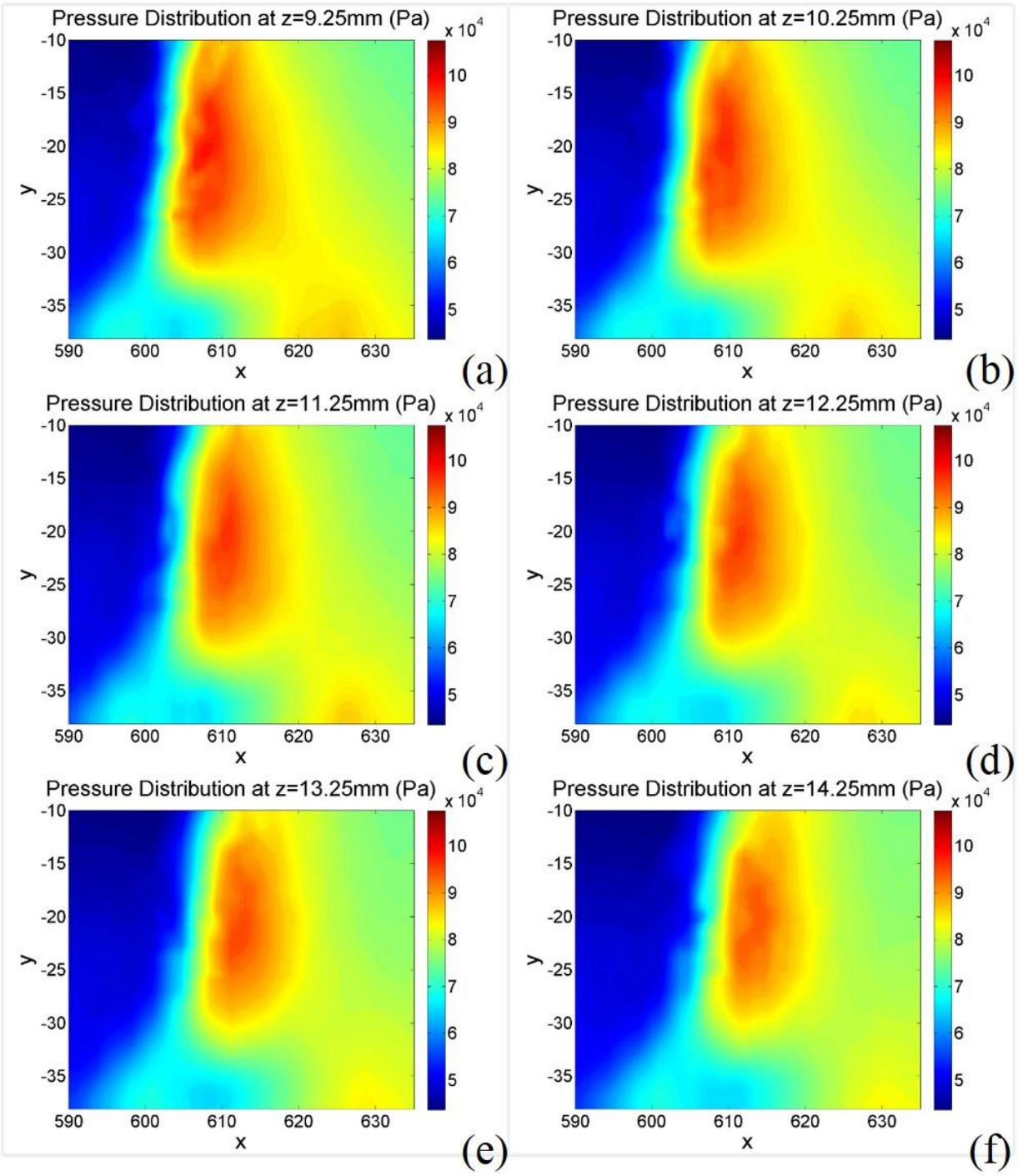


Figure 6.9: Pressure Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

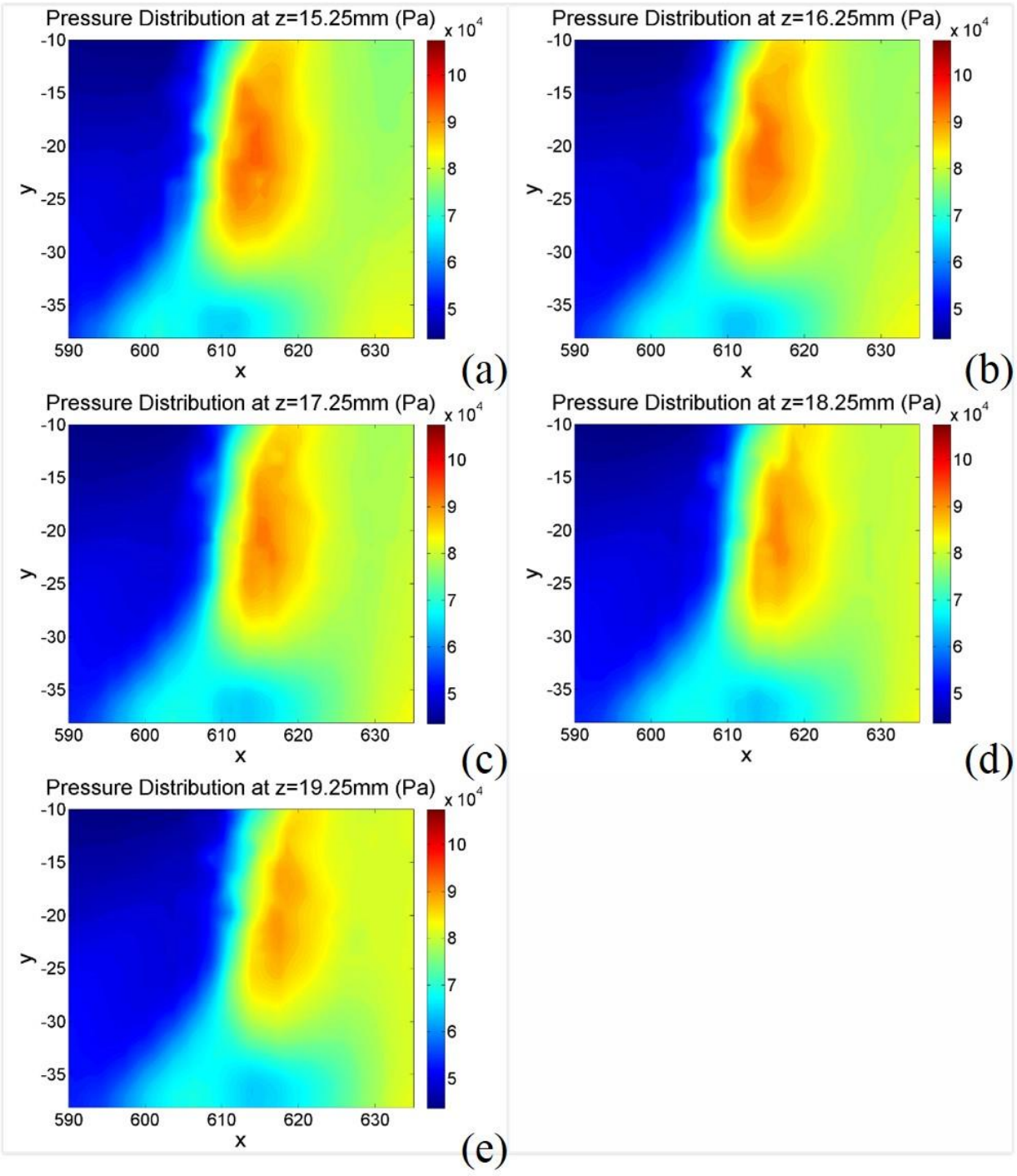


Figure 6.10: Pressure Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

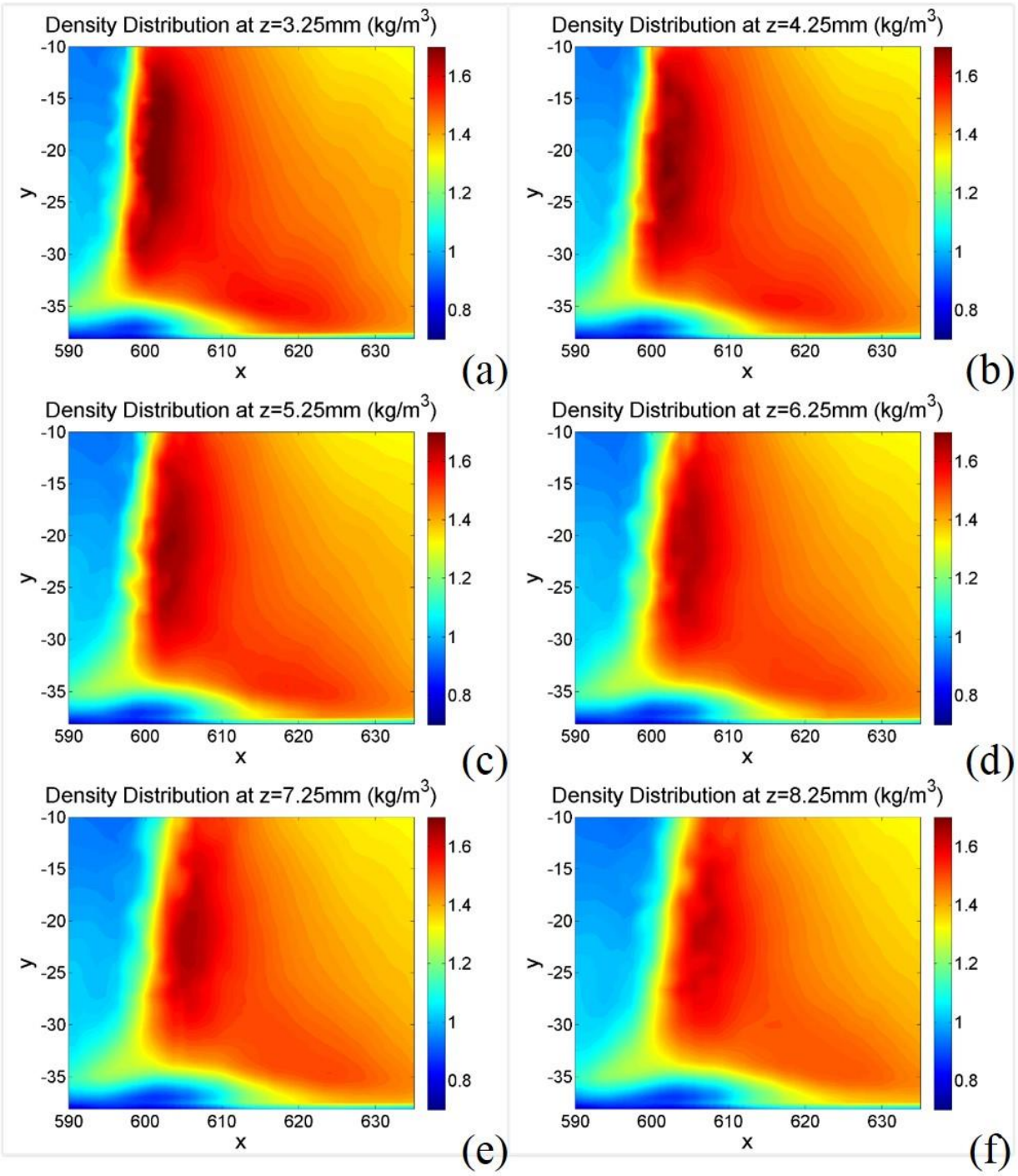


Figure 6.11: Density Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

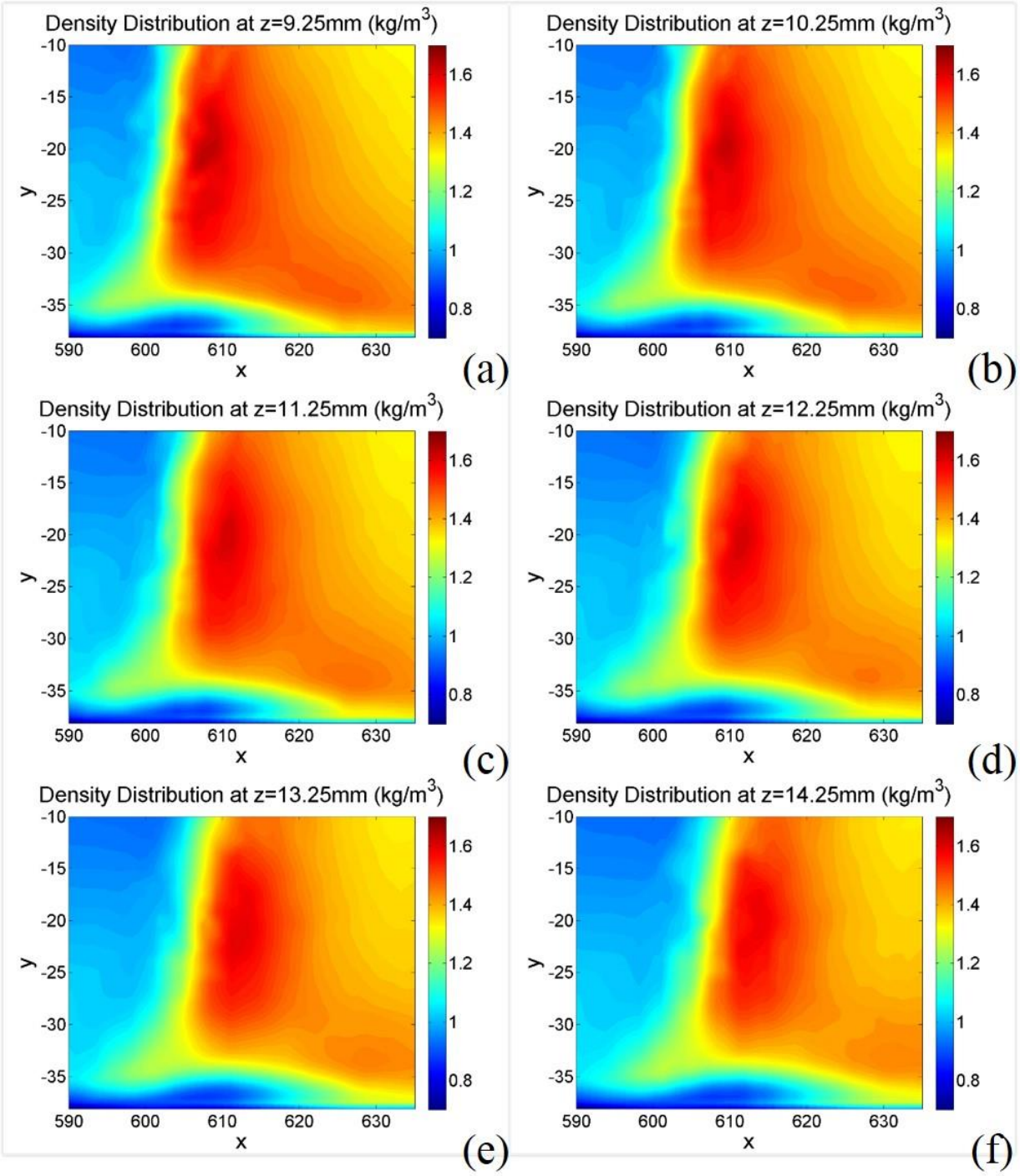


Figure 6.12: Density Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

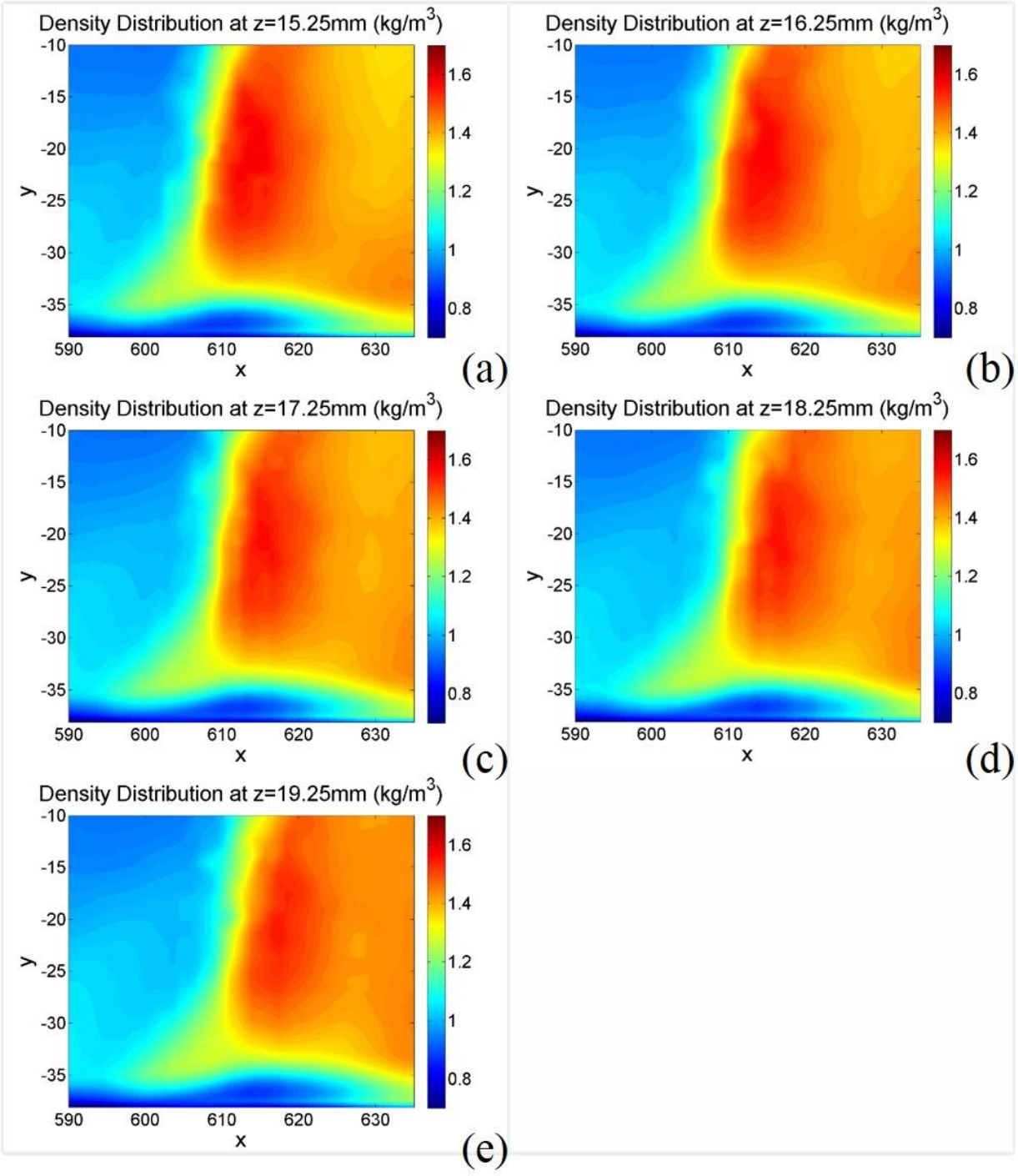


Figure 6.13: Density Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

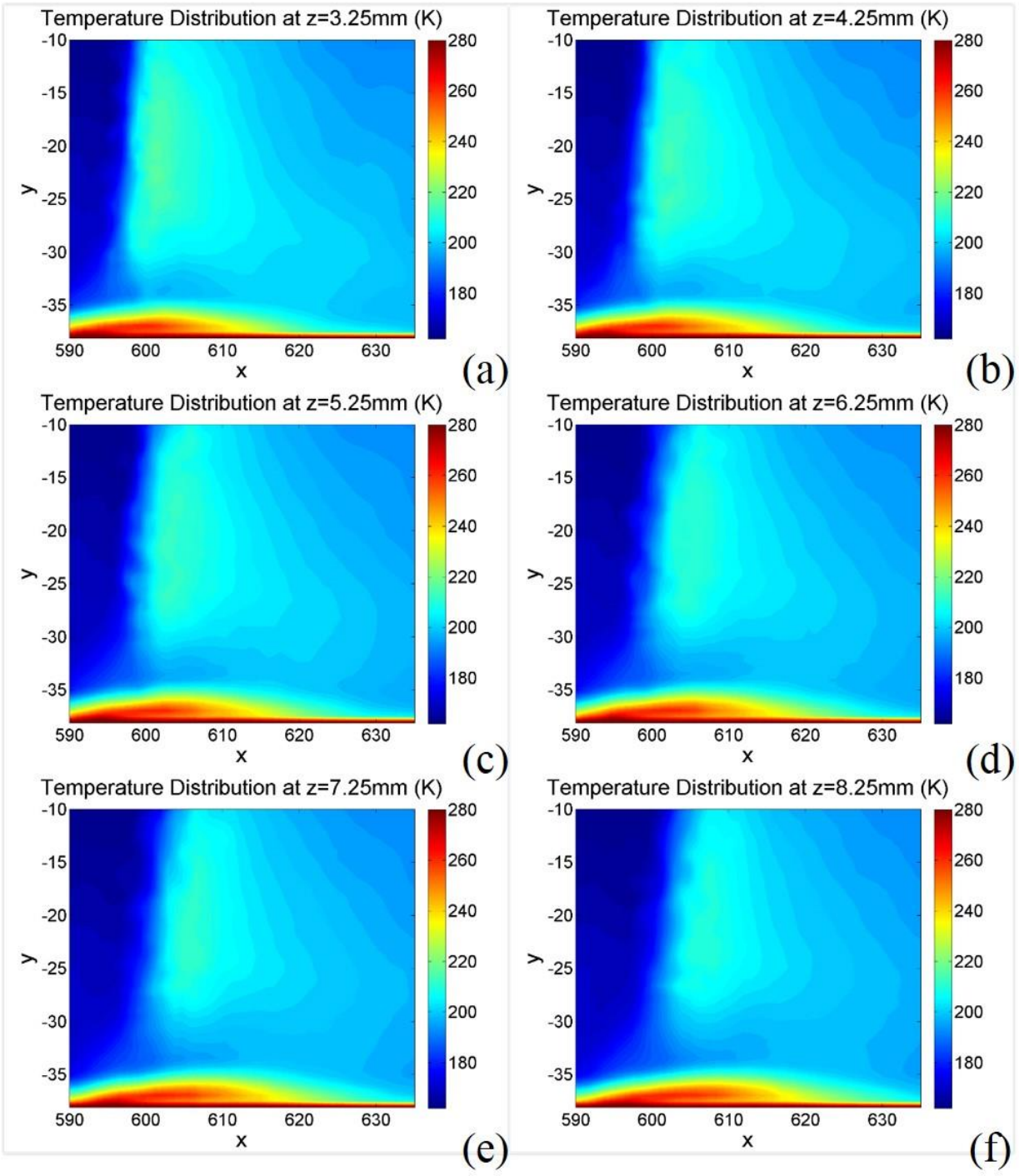


Figure 6.14: Temperature Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

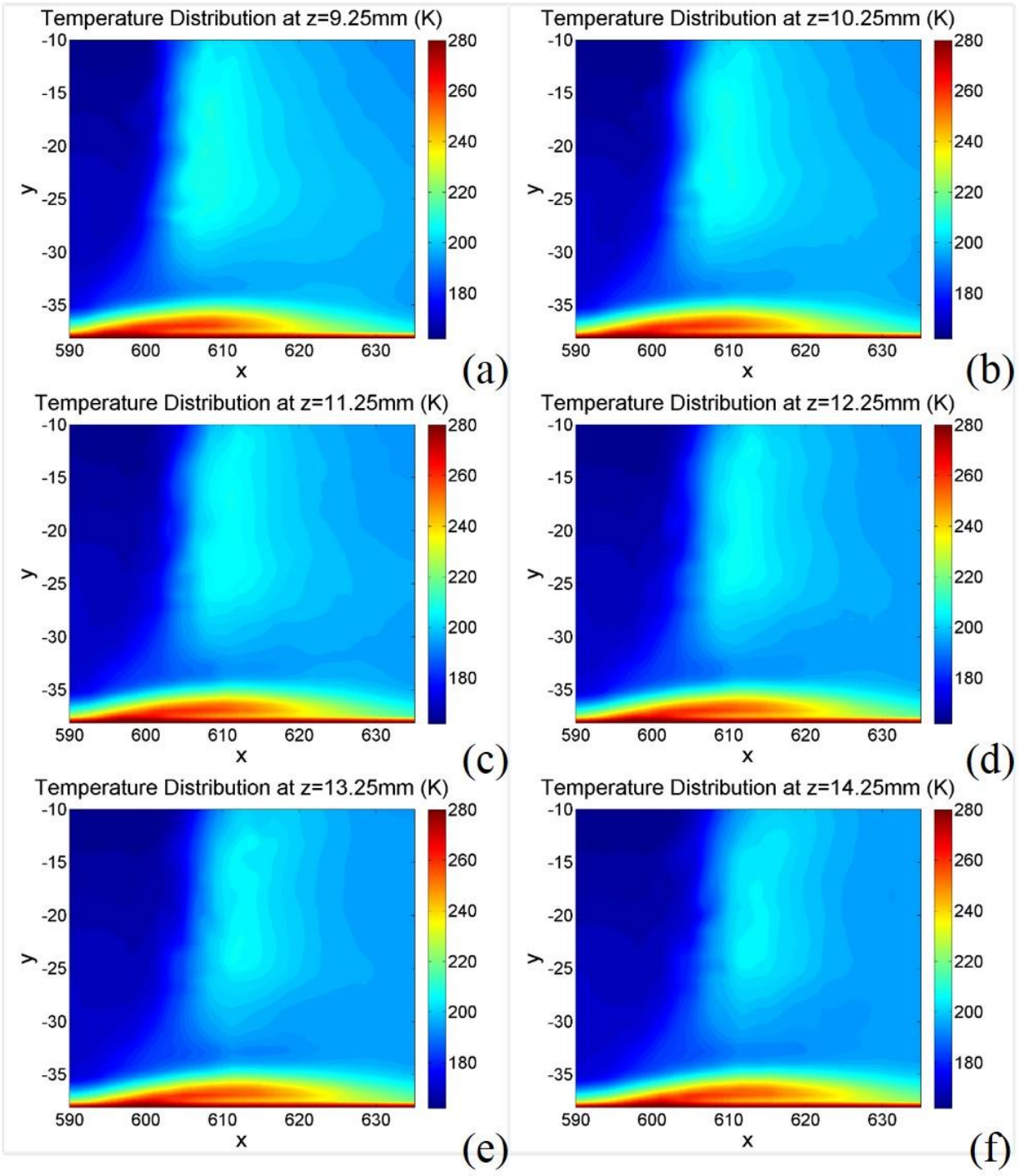


Figure 6.15: Temperature Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

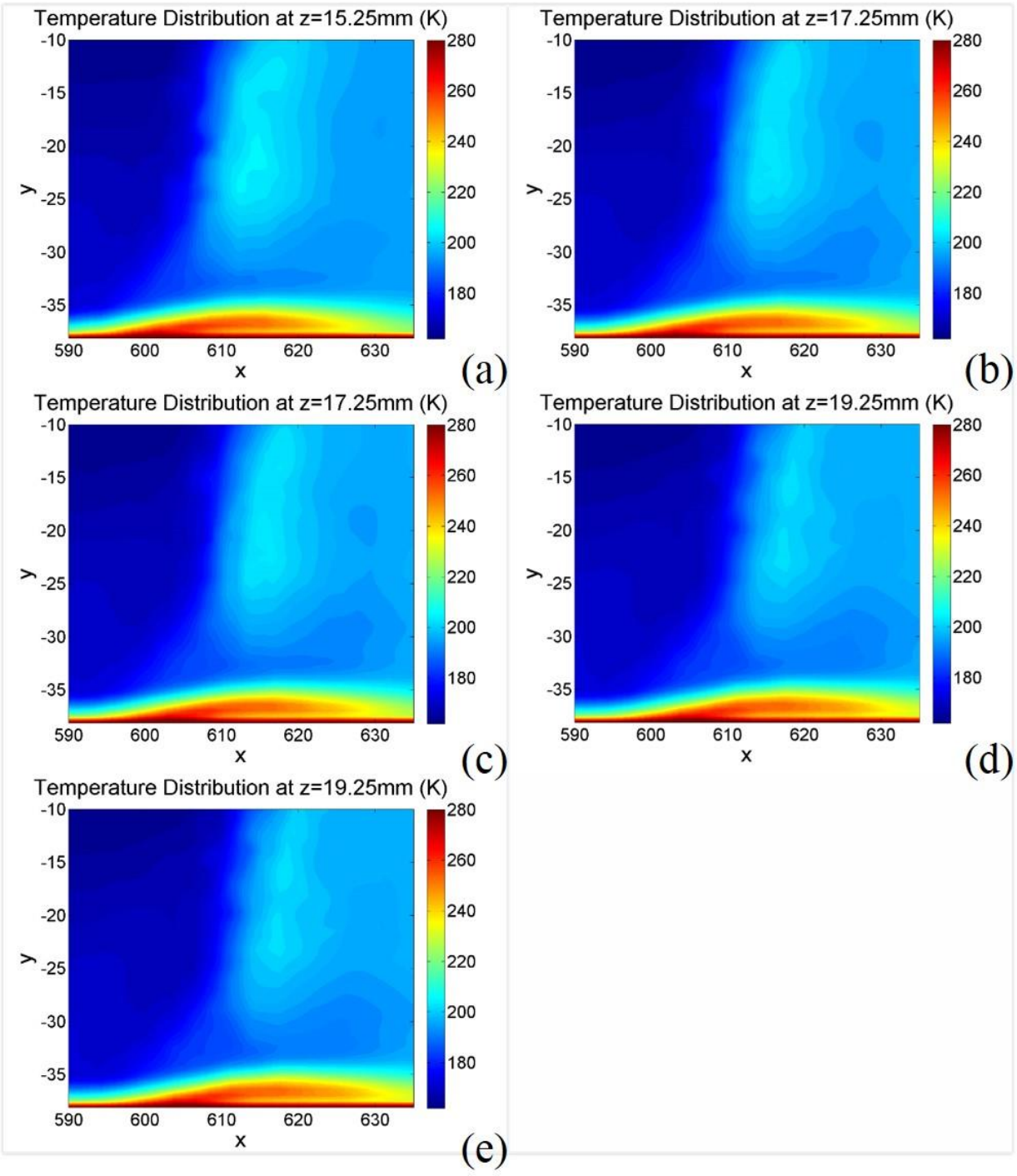
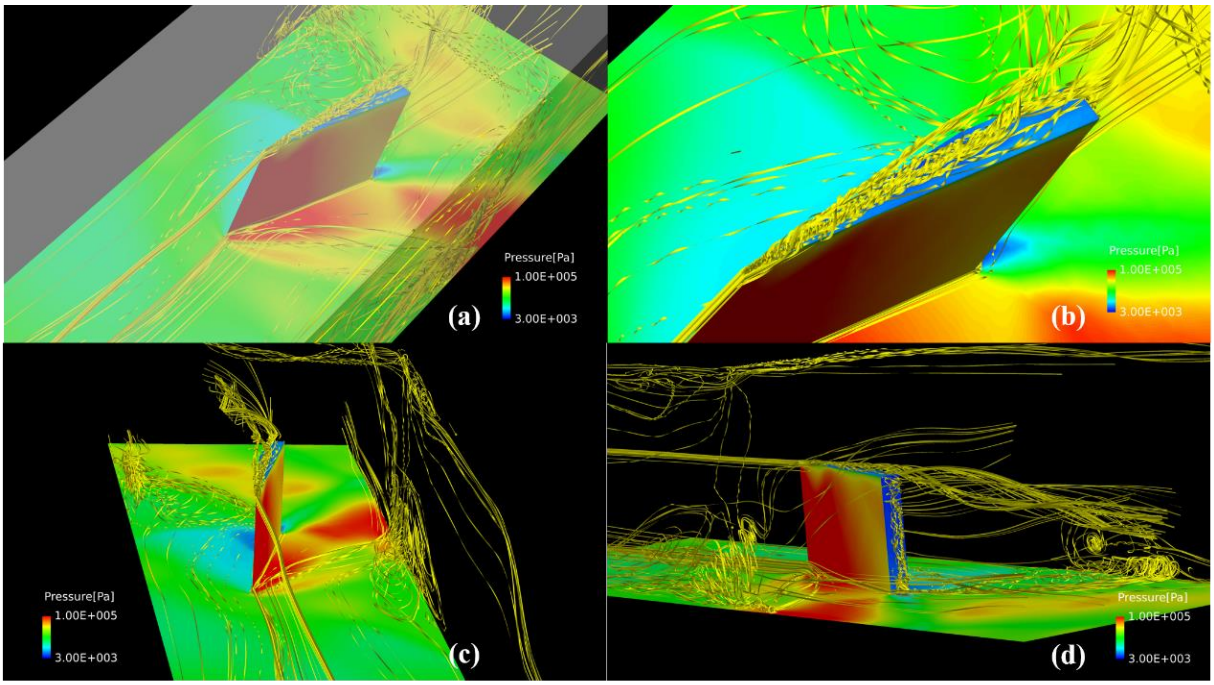


Figure 6.16: Temperature Distribution in the SWBLI of the In-Tunnel Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

6.2 In-Tunnel Case Using Blended SST $k-\varepsilon, k-\omega$ Turbulence Model



(a) The turbulence streamlines in the test section; (b) The tailing vortex on the top surface of the fin;
(c) The front view of streamlines; (d)The back view of streamlines;

Figure 6.17: Supersonic Flow Streamlines in the Test Section Using Blended $k-\varepsilon, k-\omega$ Model

The streamlines of the supersonic flow in the test section have been shown in Fig. 6.17. Similar to the numerical simulation using the Wilcox $k-\omega$ model, the tailing vortexes form at the leading edge and on the top surface of the fin. The vortexes occurring and being constrained in the region near the rear surface of the fin incessantly rotate dragging the model. Because of the oblique shock wave, the stream lines are forced to turn toward the right sidewall of the test section. Since the shock wave impinges and affects the boundary layer of the right sidewall, the flow near the right sidewall of the tunnel rises to the top surface of the tunnel and vortexes are generated and intermingled together. In addition, the blended $k-\varepsilon, k-\omega$ model provides a better description of the turbulence in the tunnel. The vortexes upstream of the shock wave and the tailing vortexes

formed at the intersection of the fin rear surface and the bottom mounting plat are illuminated.

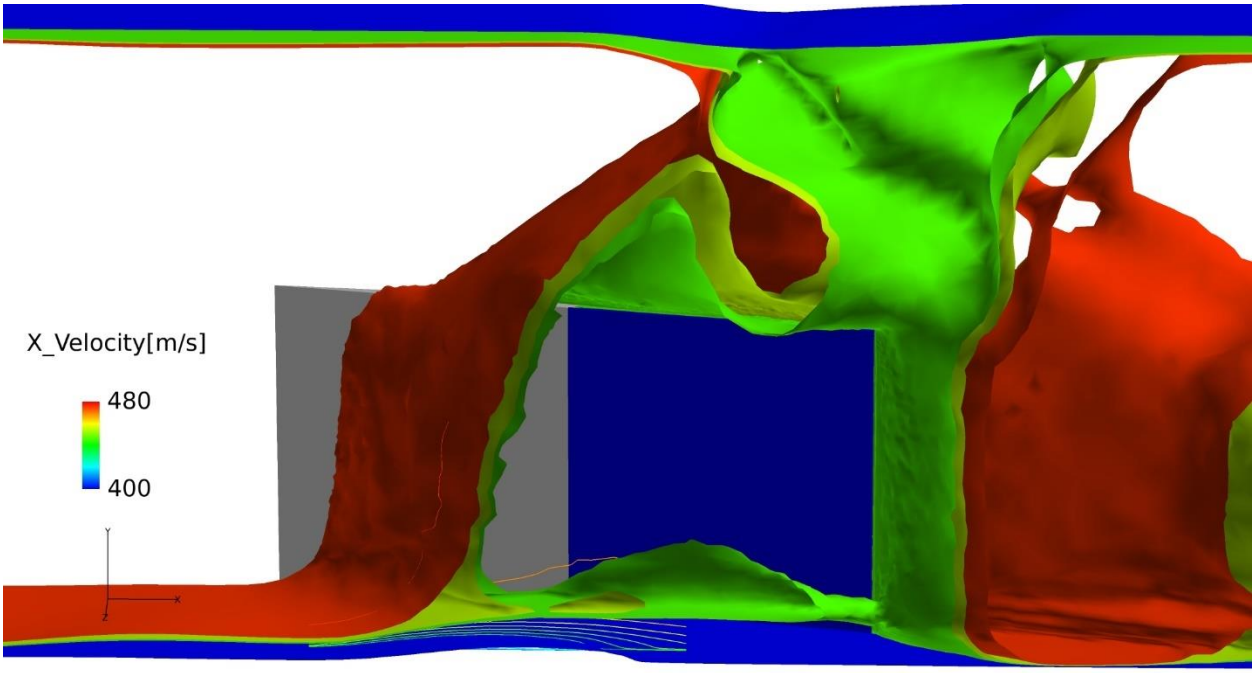


Figure 6.18: The Structure of the SWBLI Generated by the Fin in the Test Section Using Blended $k-\varepsilon, k-\omega$ Model (Iso-Surfaces of X-Velocity)

By using the blended $k-\varepsilon, k-\omega$ model, the 3D structure of the SWBLI generated by the fin in the test section is demonstrated by means of the velocity iso-surfaces in Fig. 6.18. For the same reason introduced in the previous section, a smaller value of the velocity is selected to represent the oblique shock wave. The λ -structure is shown in Fig. 6.18 by three color iso-surfaces where the main oblique shock wave and separation shock wave, the rear shock wave, and the slipline region are presented by the red, yellow, and green iso-surface, respectively. The three iso-surfaces separate at the triple point, and the portion under the separation shock wave and the rear shock wave is the separation region. With the shock wave propagation away from the fin toward the right sidewall of the tunnel, the size of the λ -structure increases. In the extension process of the

slipline region, the iso-surface of the same velocity is not always continuous. As Fig. 6.18 shows, the slipline region is cut off by the lower velocity portion. The main shock wave above the fin toward the ceiling of the tunnel turns downward and generates a high velocity region above the model.

Distributions of the different flow characteristics are provided by the slice cut figures in the z-direction in order to show the growth process of the SWBLI in the in-tunnel case using the blended $k-\varepsilon, k-\omega$ model. The distribution of the x-direction component of the velocity, the total pressure, the density, and the temperature are presented in Fig 6.19, Fig. 6.20, Fig. 6.21, Fig. 6.22, Fig. 6.23, Fig. 6.24, Fig. 6.25, Fig. 6.26, Fig. 6.27, Fig. 6.28, Fig. 6.29, and Fig. 6.30, respectively. The x-velocity distribution provides a good illustration of the development of the SWBLI structure, which demonstrate the behavior shown in Fig 6.19, Fig. 6.20, and Fig. 6.21. The downstream of the main shock wave is a low-velocity region (green and yellow region), which is terminated by the rear shock wave and slip surfaces generated by the SWBLI. The separation region increases and the slipline region extends backward with the increase of the z location. The thickness of the boundary behind the shock wave increases significantly and uses a longer distance compared with Wilcox $k-\omega$ model to change back to the thickness upstream of the shock wave. The low-velocity region behind the main shock wave and the subsonic region at the bottom in this case are larger than that in the in-tunnel case simulation using the Wilcox $k-\omega$ model. Moreover, the velocity gradient is also clearer than that in the simulation utilizing the Wilcox $k-\omega$ model.

The total pressure distribution presented in Fig. 6.22, Fig. 6.23, and Fig. 6.24 shows a high-

pressure region behind the main shock wave. It is obvious that the separation bubble is located at the bottom of the separation region. The pressure "jet", which is more intense than that in the simulation using the Wilcox $k-\omega$ model, also happens from the triple point to the bottom along the separation shock wave. With the development of the separation region in z-axis, the size separation bubble is enlarged, and the high-pressure region is shrunk.

The density distribution shown in Fig. 6.25, Fig. 6.26, and Fig. 6.27 indicates a high-density region behind the main shock wave. An extreme low-density region, which is related to the extension of the slipline region, is located at the bottom and increases along the x-direction.

The temperature distribution presented in Fig. 6.28, Fig. 6.29, and Fig. 6.30 shows a high-temperature region behind the main shock wave. Additionally, an extreme high-temperature region occurs in the subsonic region near the bottom mounting plate which illustrates the thermal loads on the aircraft's surface caused by the SWBLI. The extreme high-temperature region increases, but the high-temperature region behind the main shock wave shrinks with the development of the main shock wave in the z-direction. The separation of temperature exists in the extreme high-temperature region because of a relatively low-temperature layer (the yellow line in Fig. 6.28, Fig. 6.29, and Fig. 6.30).

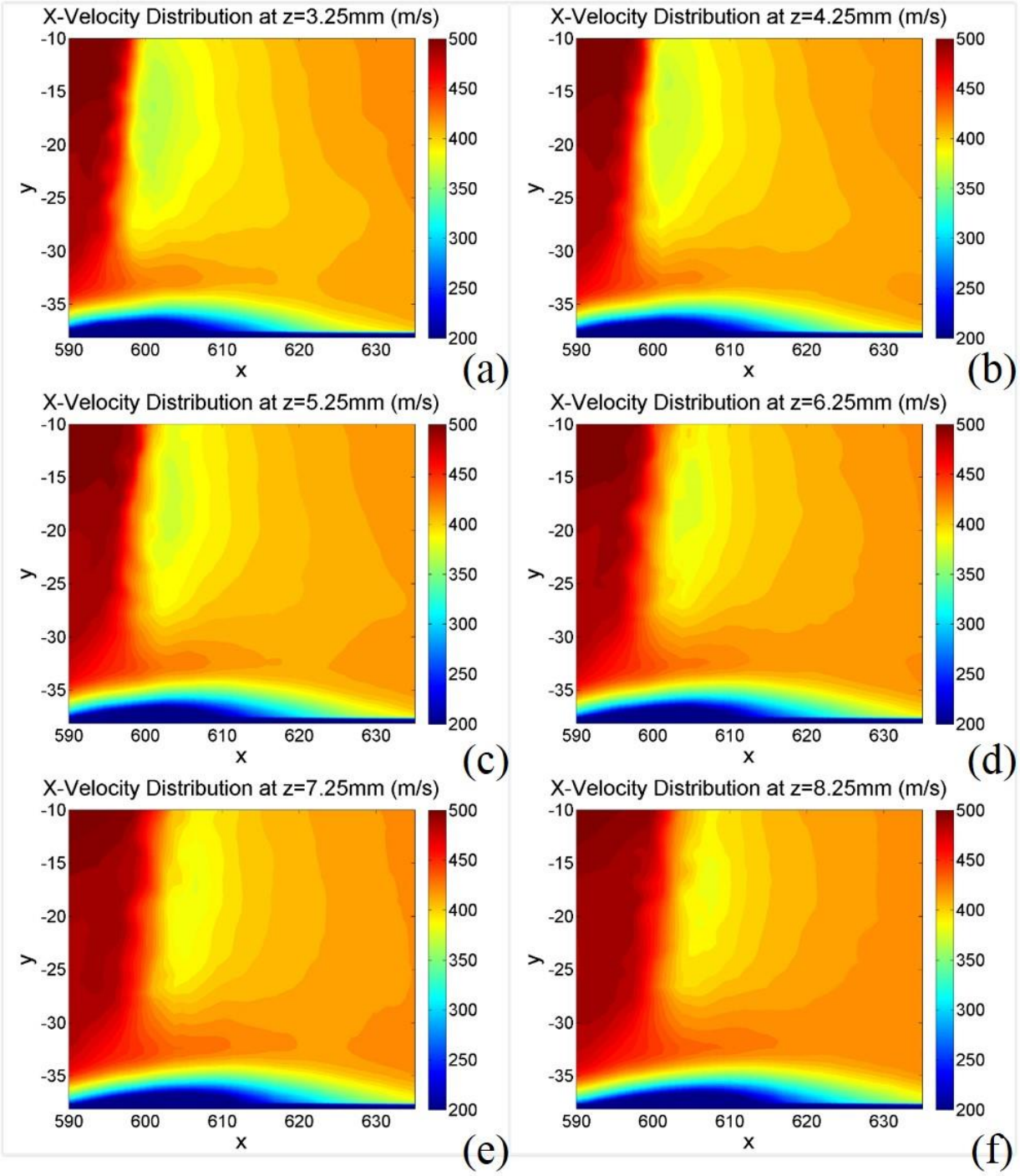


Figure 6.19: X-Velocity Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

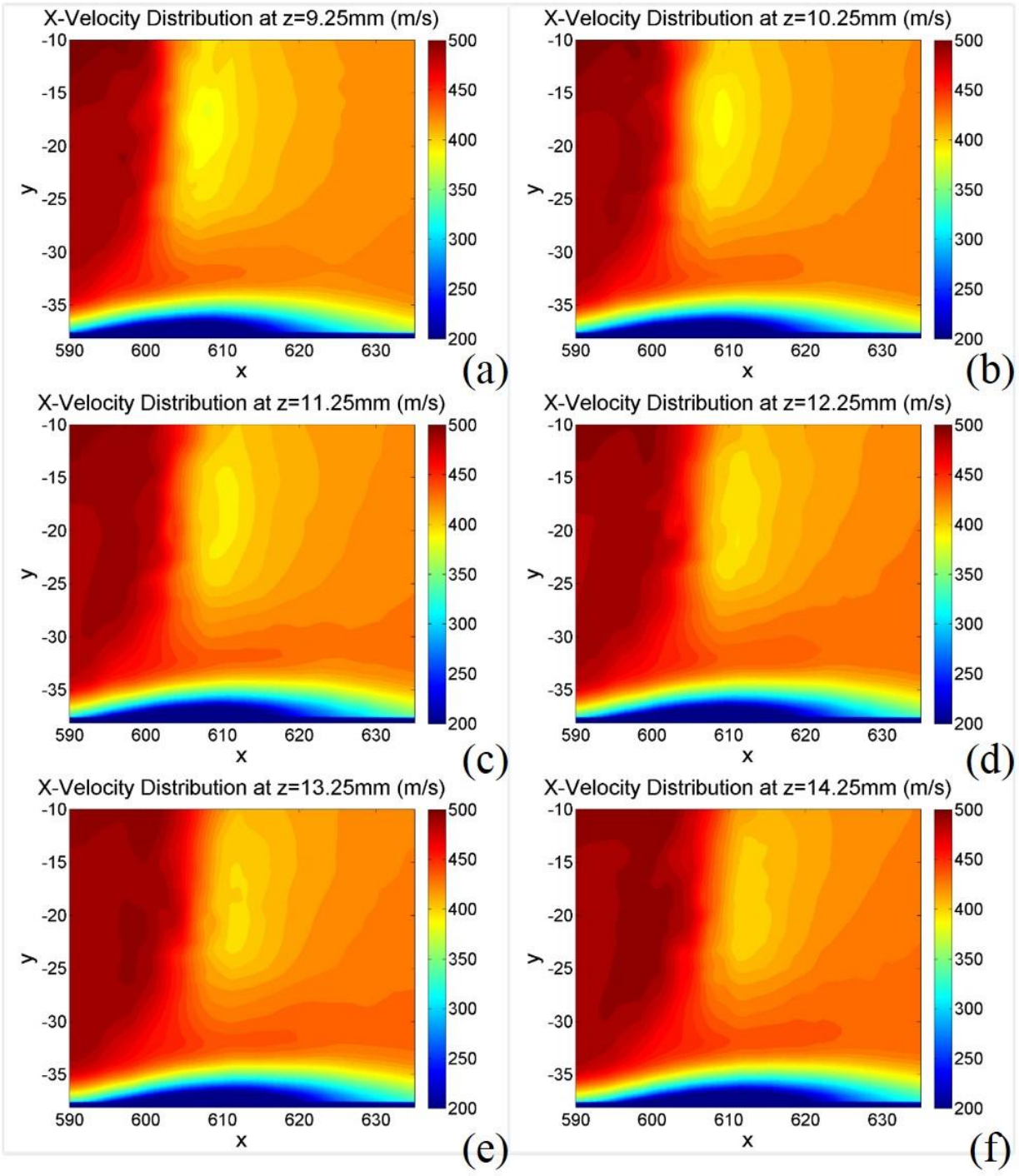


Figure 6.20: X-Velocity Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

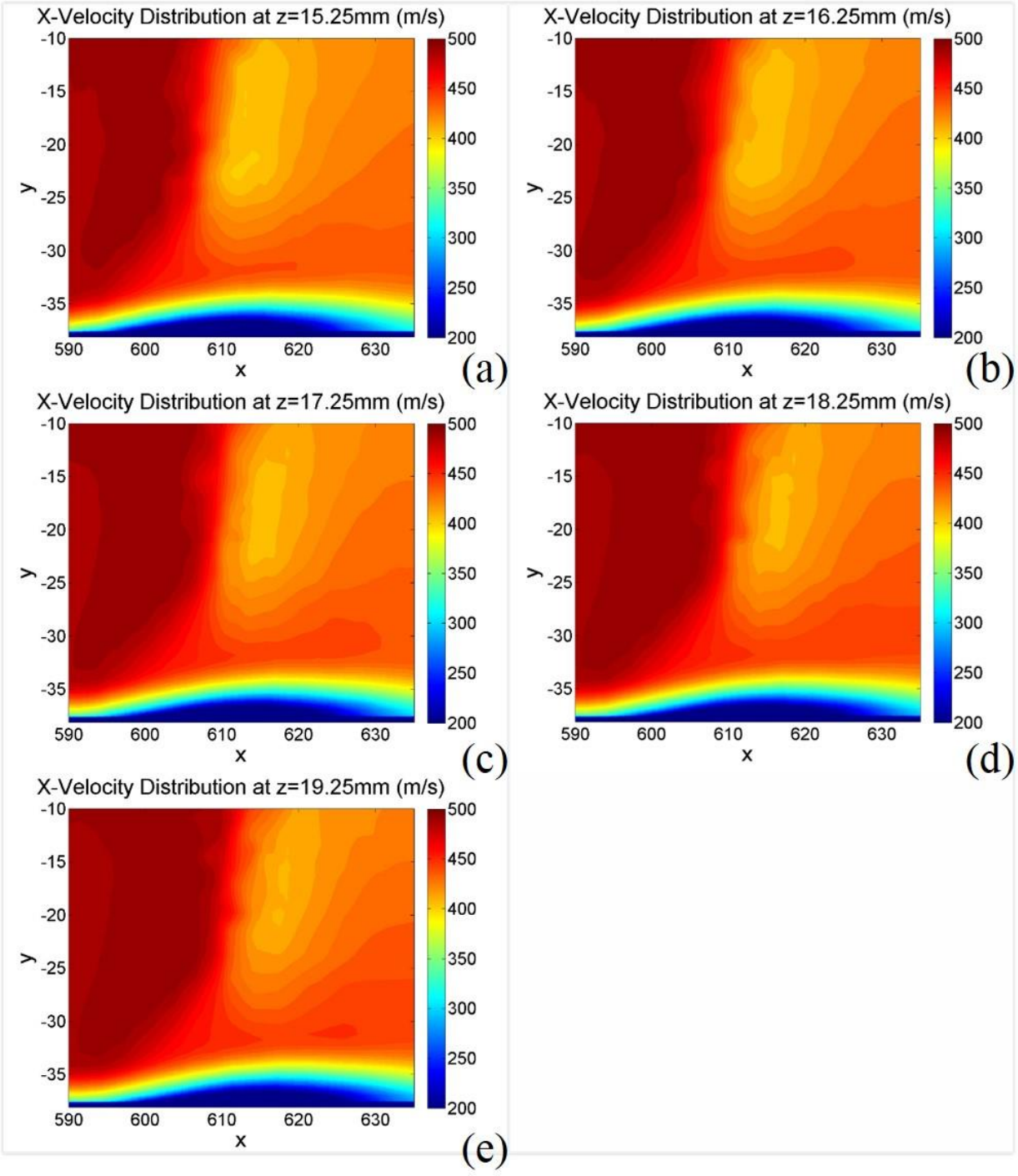


Figure 6.21: X-Velocity Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

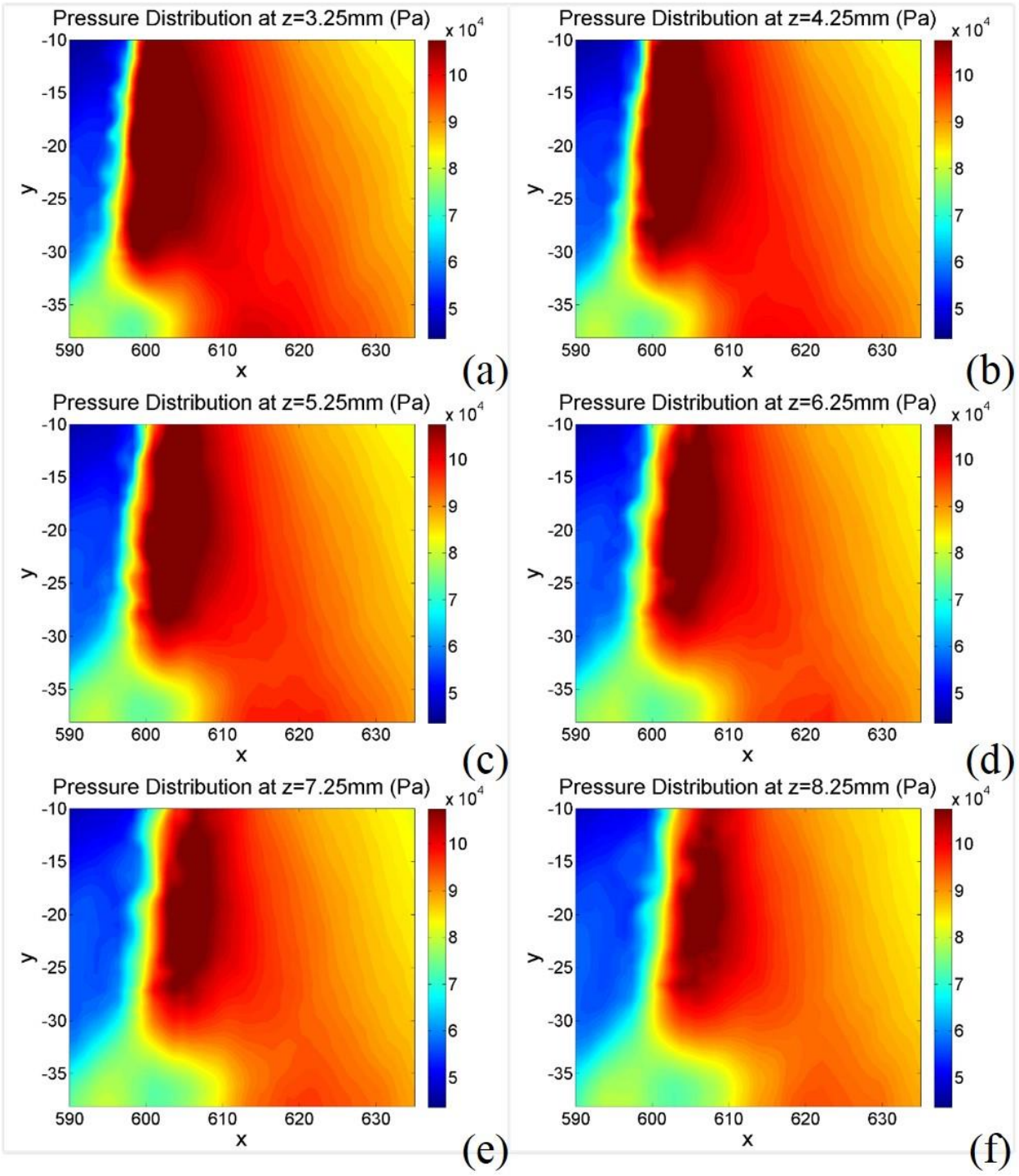


Figure 6.22: Pressure Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

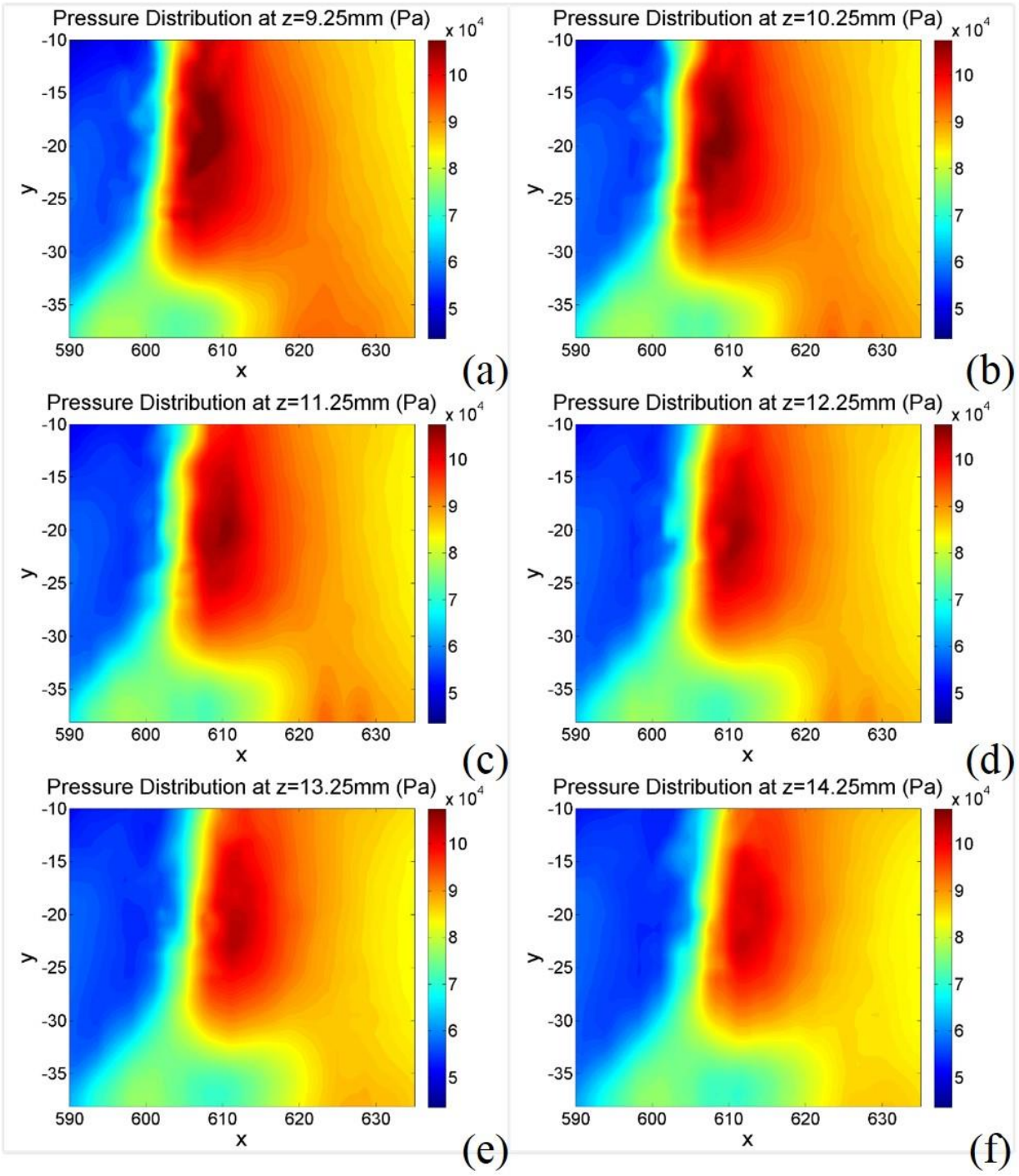


Figure 6.23: Pressure Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

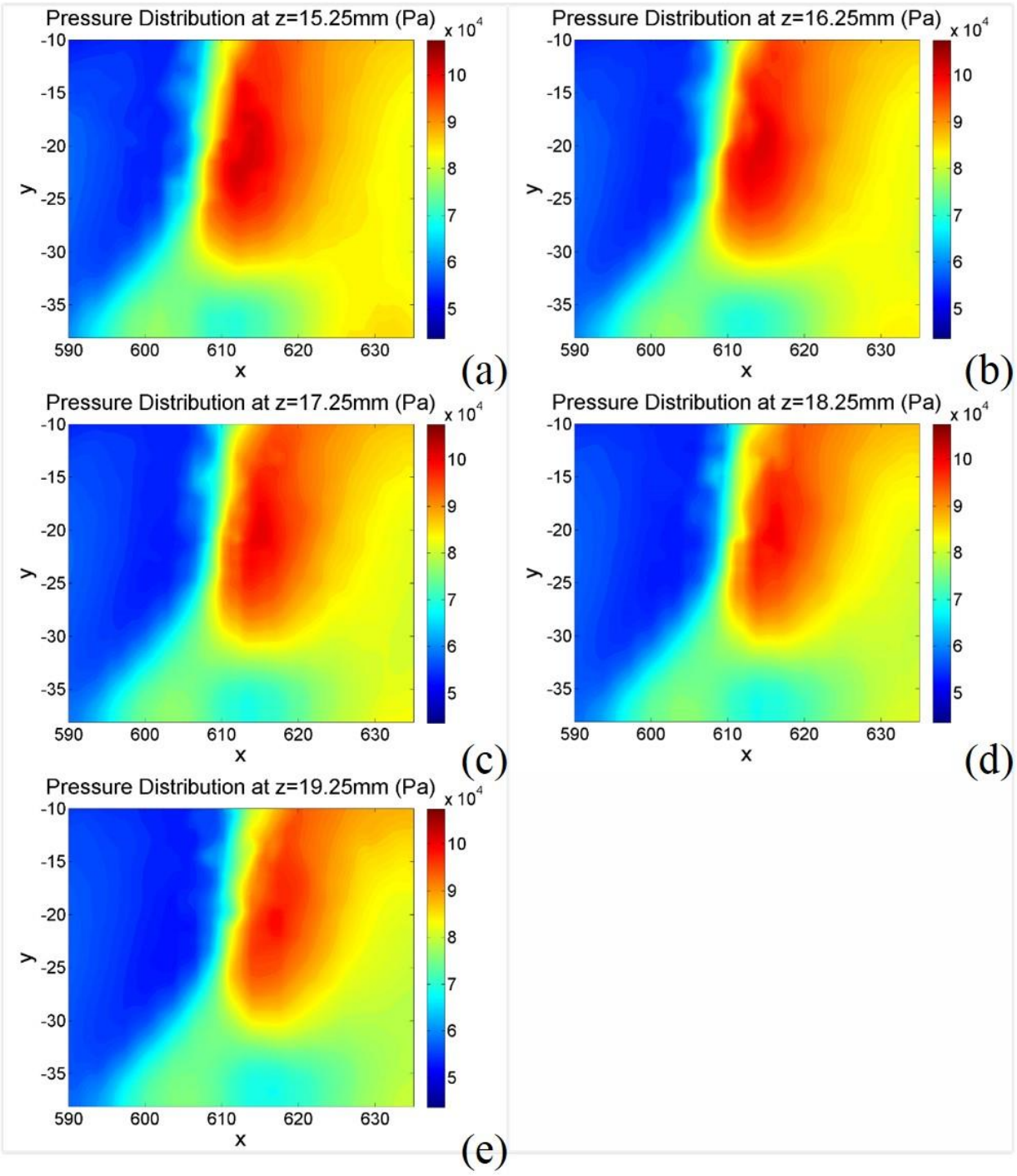


Figure 6.24: Pressure Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

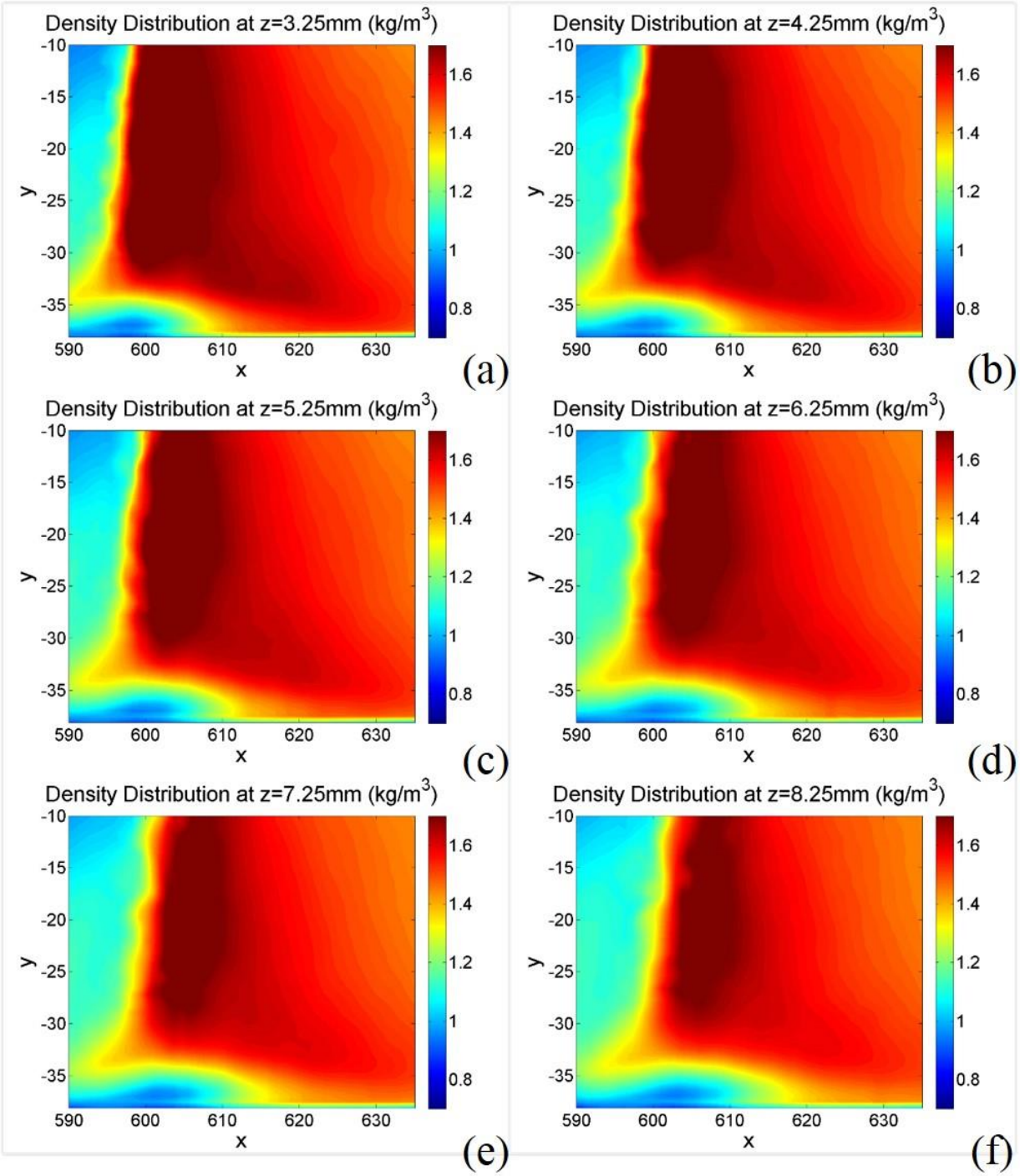


Figure 6.25: Density Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

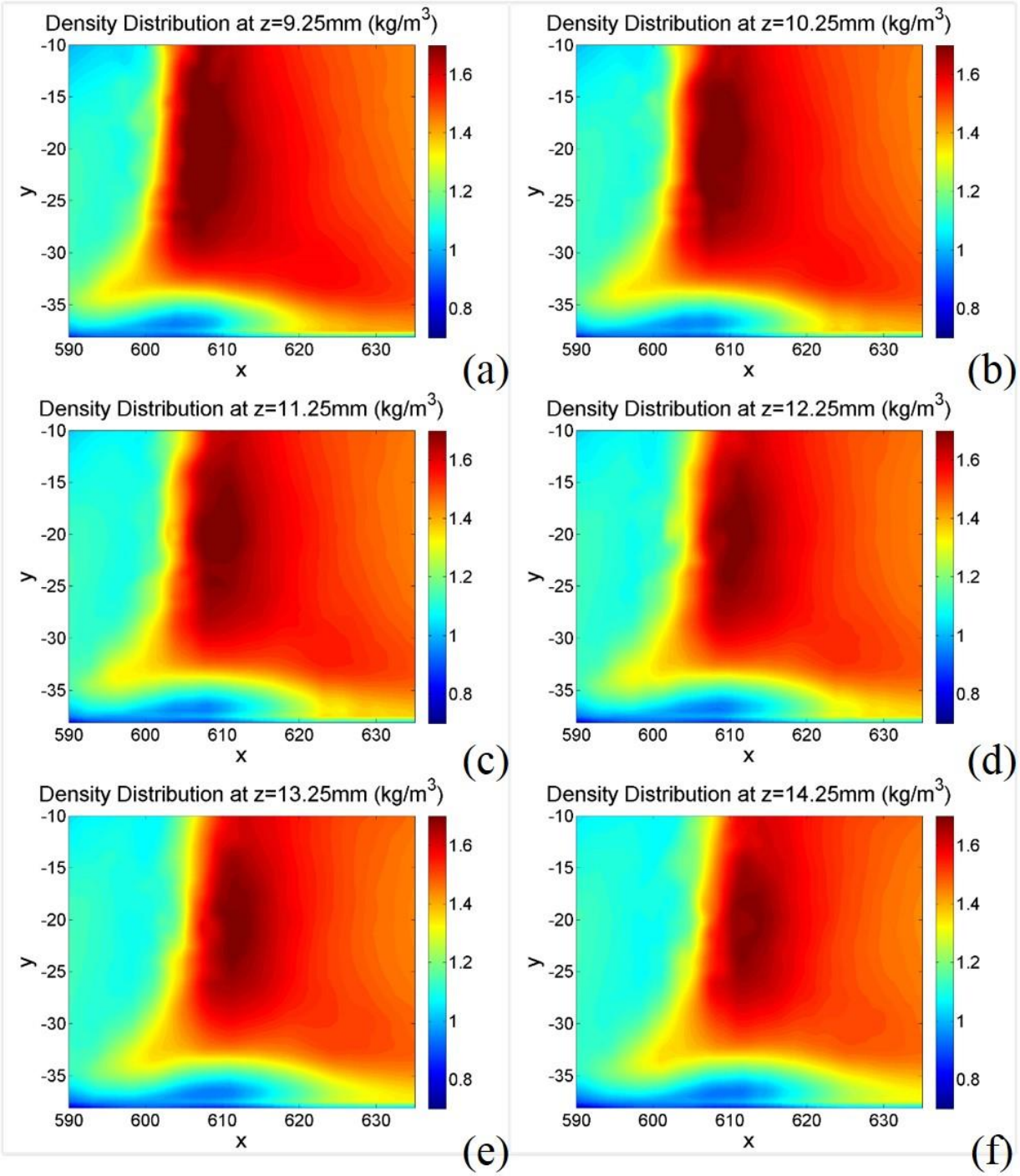


Figure 6.26: Density Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

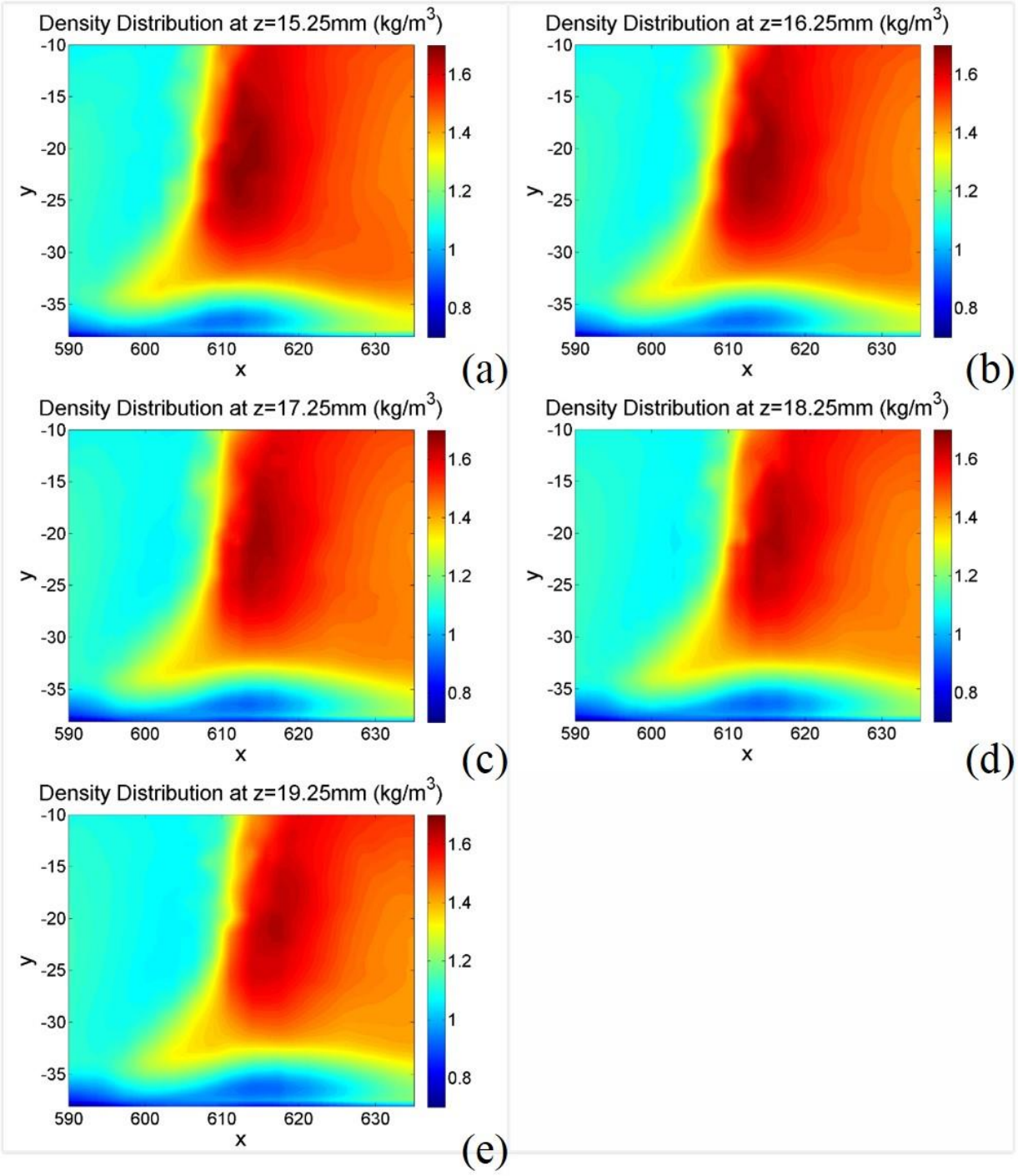


Figure 6.27: Density Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

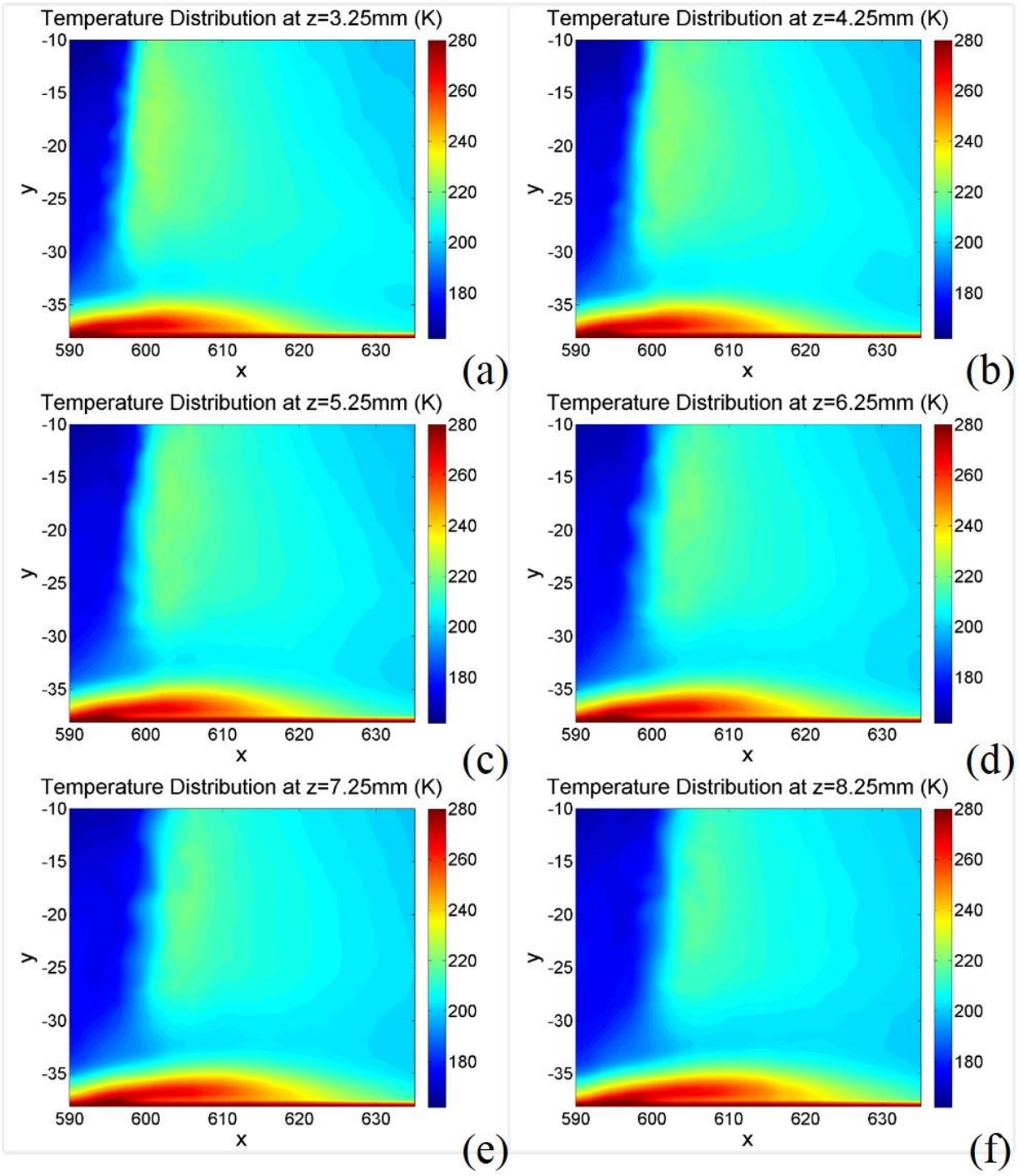


Figure 6.28: Temperature Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

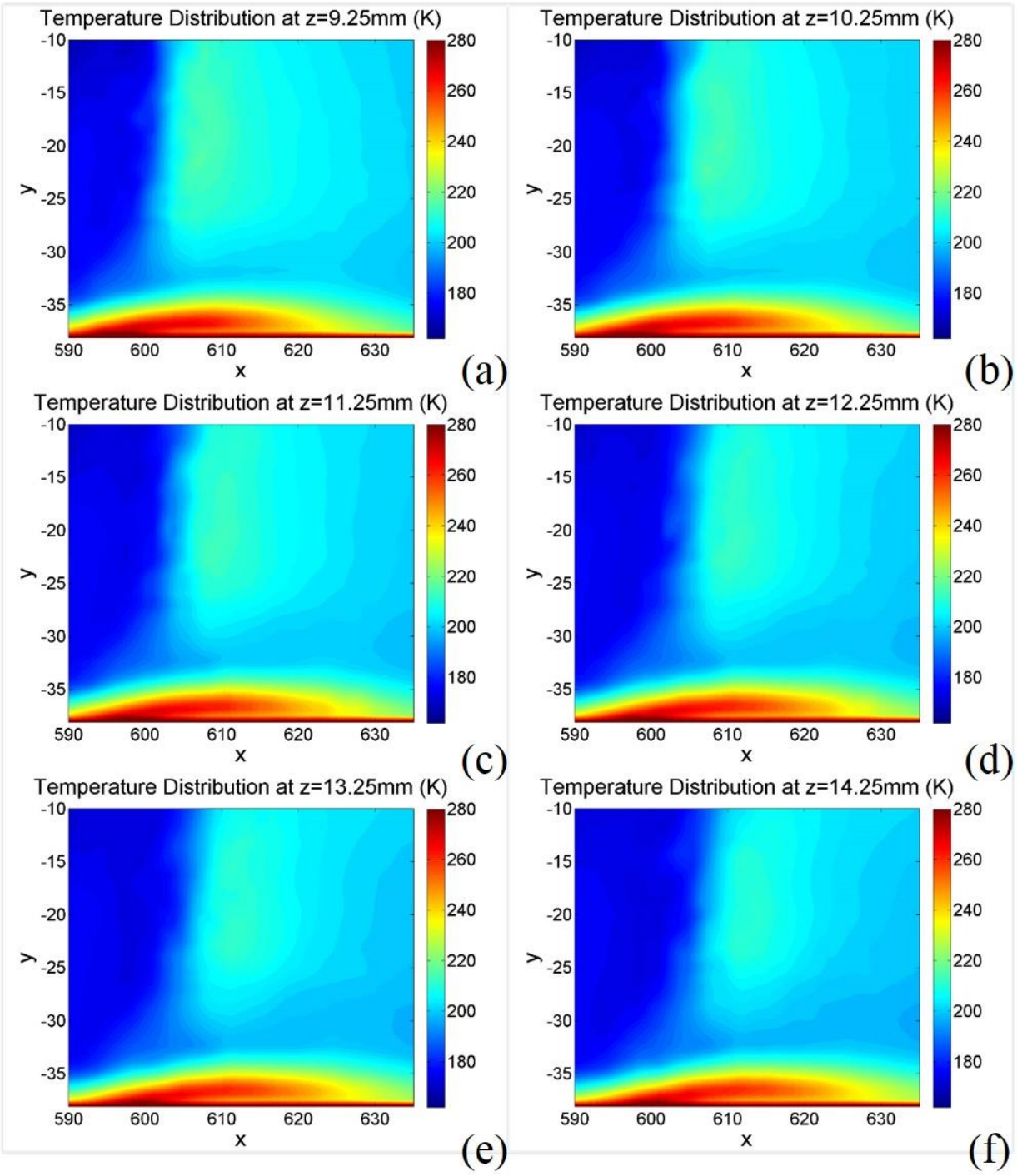


Figure 6.29: Temperature Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

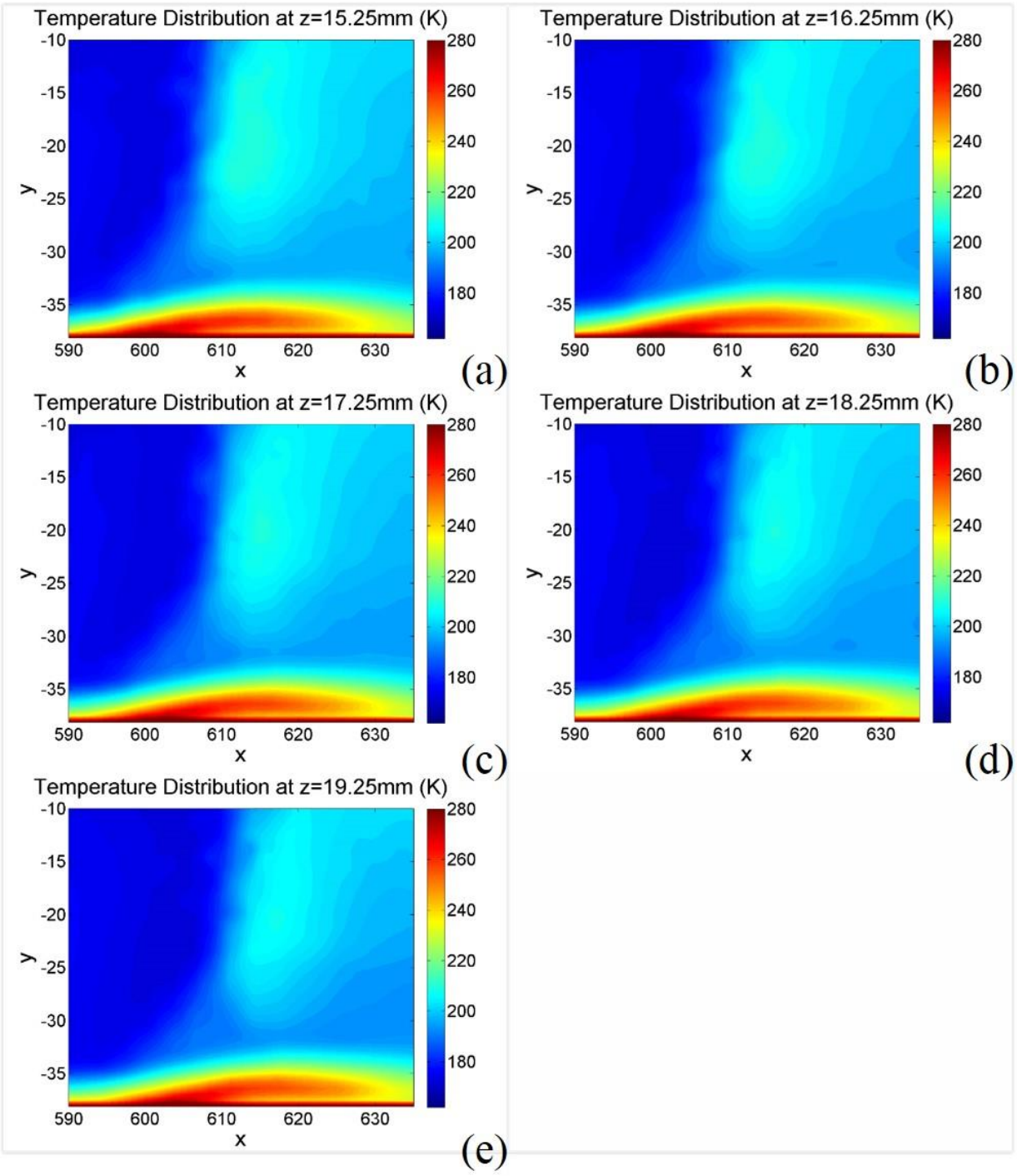
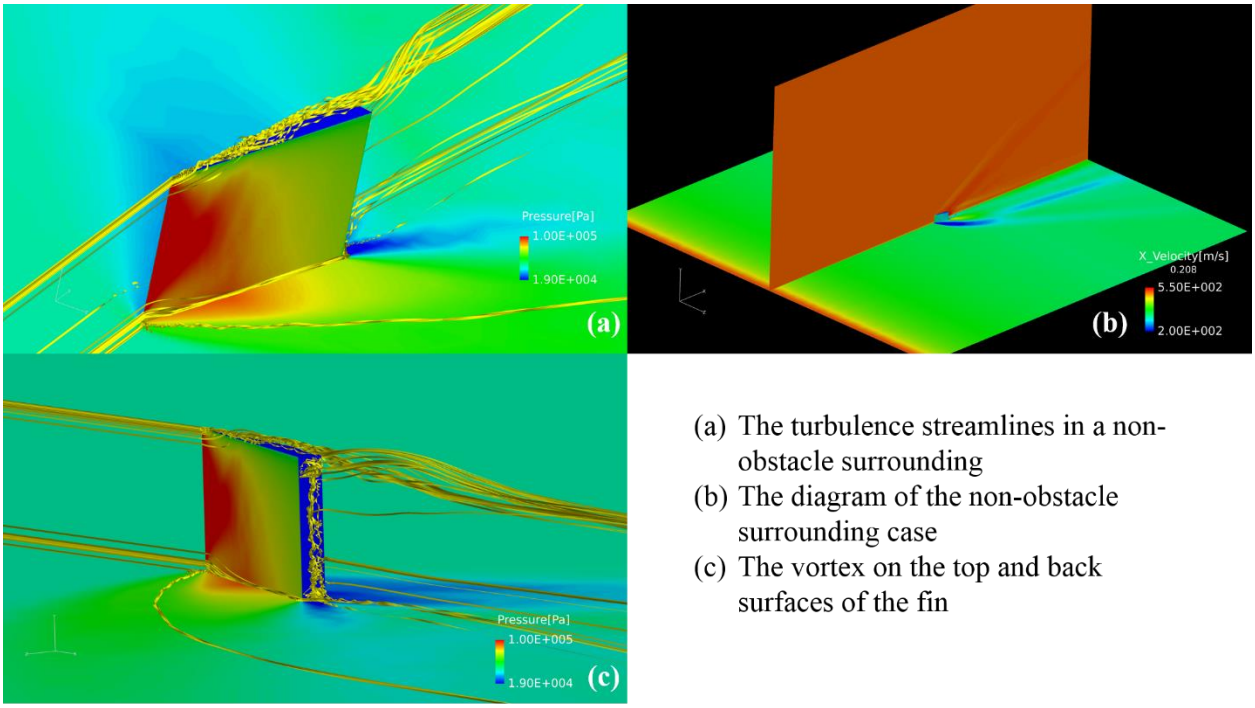


Figure 6.30: Temperature Distribution in the SWBLI of the In-Tunnel Case Simulation Using Blended $k-\varepsilon, k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

6.3 Non-Obstacle Surrounding Case Using Wilcox $k-\omega$ Turbulence Model



(a) The turbulence streamlines in a non-obstacle surrounding
(b) The diagram of the non-obstacle surrounding case
(c) The vortex on the top and back surfaces of the fin

Figure 6.31: Supersonic Flow Streamlines in the Non-Obstacle Surrounding Using the Wilcox $k-\omega$ Model

The farfield (non-obstacle surrounding) case has the different simulation environment and the boundary conditions setup. The free stream velocity is artificially set to $514\ m/s$ in order to make the flow reach Mach number 2. For this reason, as shown in Fig. 6.31 (b), the velocity at the bottom of the inlet is not zero, however, the velocity suddenly decreases to transonic and subsonic because of the viscous effects. Therefore, it is considered that the viscous boundary layers are formed correctly at the beginning of the numerical simulation, which is shown in the Fig. 6.31 (b). From the vertical cutting plan in Fig. 6.31 (b), the free stream velocity is approximately Mach 2, and the main shock wave propagates to the far field without the obstructive effects of the sidewalls of the tunnel. The oblique shock is generated by the fin, and the separation line is curved to the

downstream.

The streamlines of the supersonic flow around the fin have been shown in Fig. 6.31 (a) & (c). Similar to the in-tunnel case simulation using Wilcox $k-\omega$ model, the tailing vortexes form at the top surface of the fin and the intersection of the rear surface of the fin and the bottom mounting plate. The vortexes near the rear surface of the fin incessantly rotate dragging the model. Because of the oblique shock wave, stream lines are forced to turn toward the right following the separation line and then turn back toward the free stream direction. Compared with the stream lines in the in-tunnel case using Wilcox $k-\omega$ model presented in Fig. 6.3, the turbulence flow around the fin and the oblique shock wave are uncomplicated. In the Fig. 6.3, the stream line along the separation lines impinges the right surface of the tunnel and intermingles with the stream line near the tunnel boundary in x-direction arising to the top surface of the tunnel. However, in Fig. 6.31 (a) & (c), the stream line follows the separation line and turn toward x-direction without obstruction. As shown in Fig. 6.3, the tailing vortexes near the top surface of the fin flow downward due to the influence of the turbulence flow near the ceiling of the tunnel while the tailing vortexes near the top surface of the fin generated in the farfield case, as shown in Fig. 6.31 (c), flow in x-axis without changing its direction.

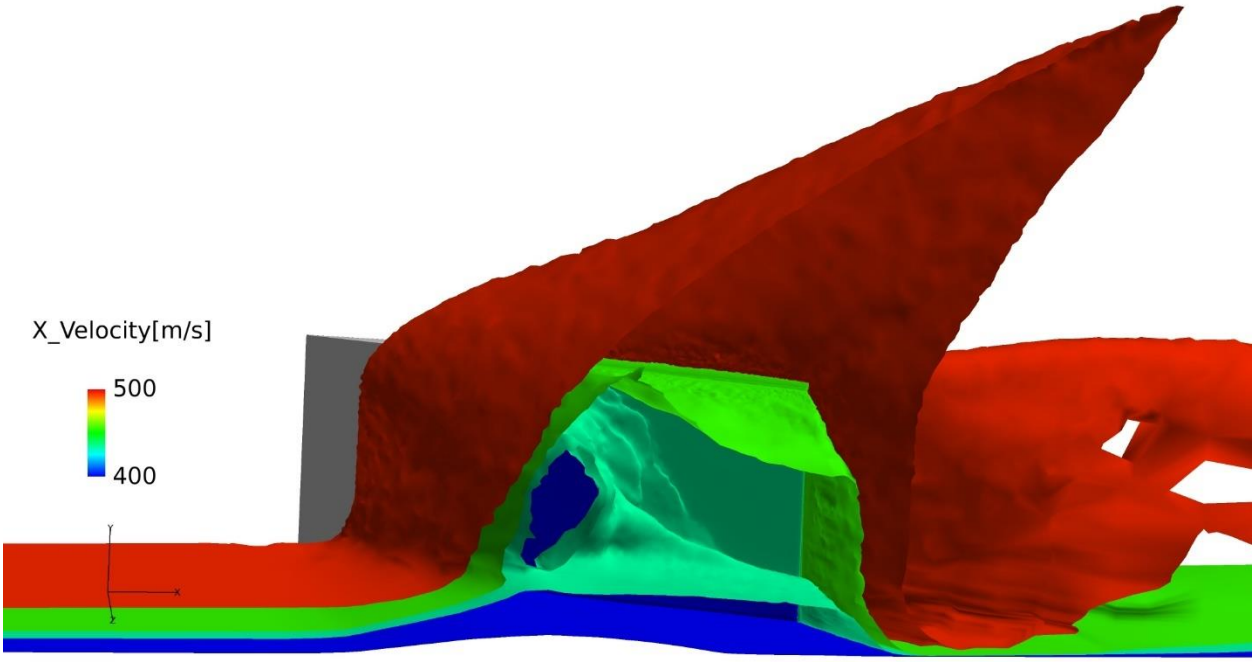


Figure 6.32: The Structure of the SWBLI Generated by the Fin in the Non-Obstacle Surrounding Using the Wilcox $k-\omega$ Model (Iso-Surfaces of X-Velocity)

By using Wilcox $k-\omega$ model, the 3D structure of the SWBLI generated by the fin in a non-obstacle surrounding condition is demonstrated by means of the velocity iso-surfaces in Fig. 6.32. The λ-structure of the SWBLI, which is not very obvious compared with in-tunnel case numerical simulations, is shown by different color iso-surfaces, where the main oblique shock wave and separation shock wave, the rear shock wave, and the slipline region are presented by the red, green and blue iso-surface, respectively. Without the tunnel, the main shock wave (red iso-surface) propagates to the far field and no reflection of shock waves exists.

Distributions of the different flow characteristics are provided by the slice cut figures in the z-direction in order to show the growth process of the SWBLI in the farfield case using Wilcox $k-\omega$ model. The distribution of the x-direction component of the velocity, the total pressure, the

density, and the temperature are presented in Fig 6.33, Fig. 6.34, Fig. 6.35, Fig 6.36, Fig. 6.37, Fig. 6.38, Fig 6.39, Fig. 6.40, Fig. 6.41, Fig 6.42, Fig. 6.43, and Fig. 6.44, respectively.

The x-velocity distribution (Fig 6.33, Fig. 6.34, and Fig. 6.35) could not show a quite obvious λ -structure of the SWBLI compared with that in the in-tunnel case. The downstream of the main shock wave is a low-velocity region. The thickness of the boundary layer increases significantly behind the shock wave and compared with the boundary layer change in the in-tunnel case (Fig. 6.5, Fig. 6.6, and Fig. 6.7), the boundary layer change on the thickness in the farfield case is much larger than that in the in-tunnel case.

The total pressure distribution presented in Fig. 6.36, Fig. 6.37, and Fig. 6.38 shows a separation of pressure and a high-pressure region exists downstream of the main shock wave. However, the separation bubble is not shown in the farfield case which is different from the in-tunnel case in Fig. 6.8, Fig. 6.9, and Fig. 6.10. Further, the high-pressure region near the bottom mounting plat behind the shock wave in the farfield case is not as obvious as the high-pressure region of the in-tunnel case using the same turbulence model.

The density distribution presented in Fig. 6.39, Fig. 6.40, and Fig. 6.41 indicates a low-density region upstream of the main shock wave while a high-density region downstream of the main shock wave. The low-density region is much larger than the low-density region in the simulation of the in-tunnel case shown in Fig. 6.11, Fig. 6.12, and Fig. 6.13.

The temperature distribution presented in Fig. 6.42, Fig. 6.43, and Fig. 6.44 shows a high-temperature region behind the main shock wave. More important, an extreme high-temperature

region occurs in the subsonic region near the bottom mounting plate, which has larger are and higher temperature than that in the simulation of in-tunnel cases using Wilcox $k-\omega$ model (Fig. 6.14, Fig. 6.15, and Fig. 6.16). This illustrates a more serious thermal load issue caused by the SWBLI compared with the in-tunnel case.

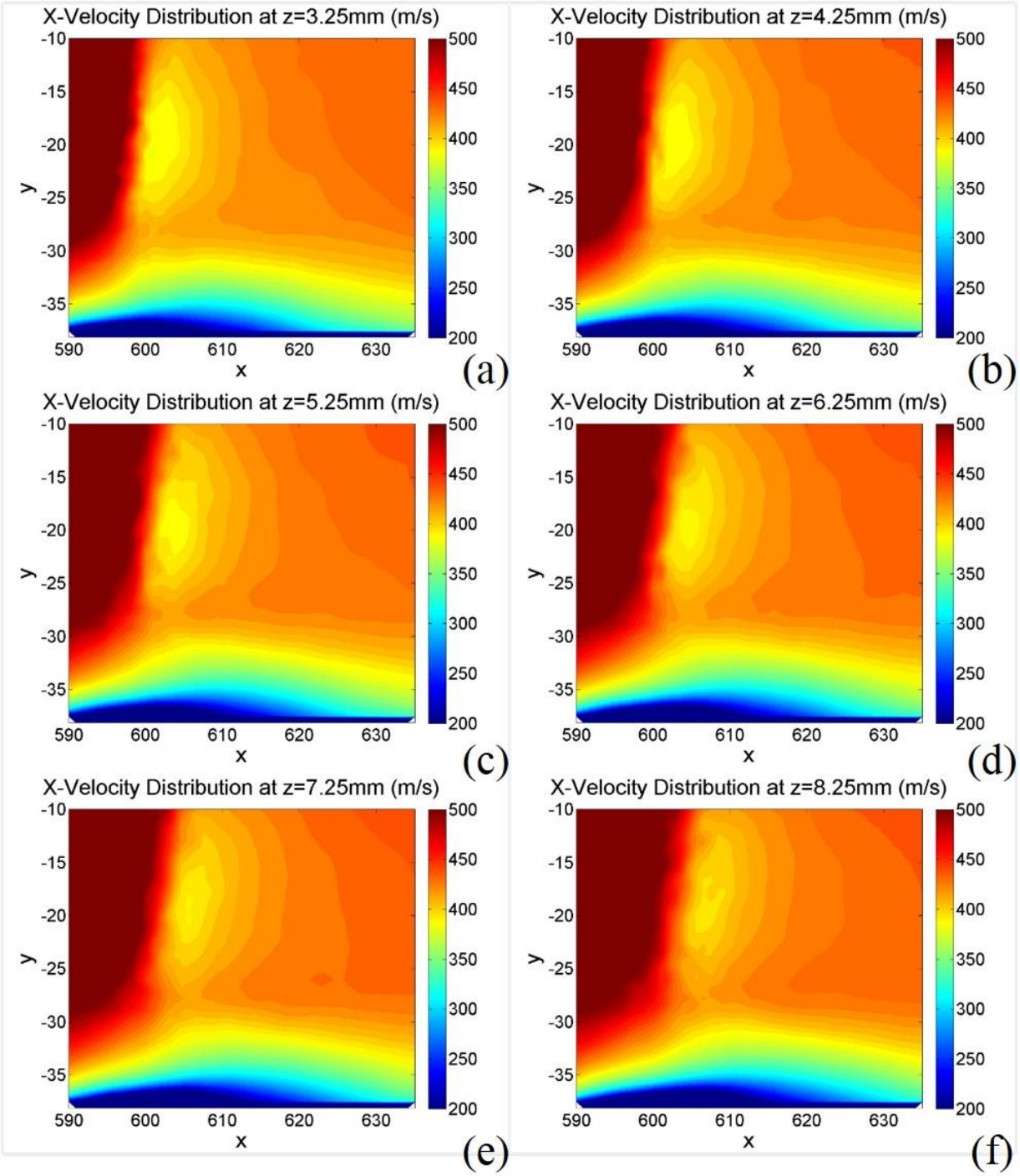


Figure 6.33: X-Velocity Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

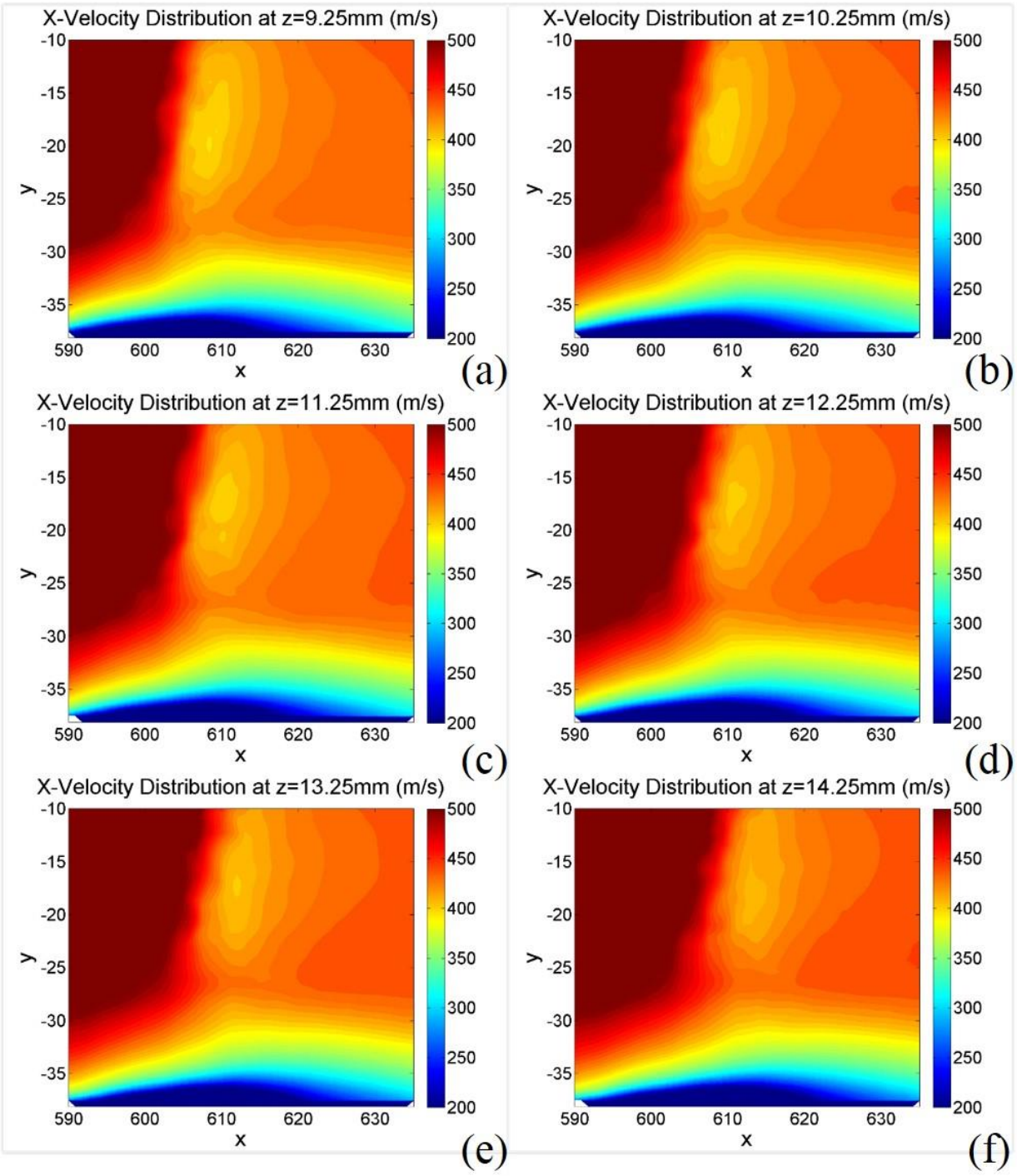


Figure 6.34: X-Velocity Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

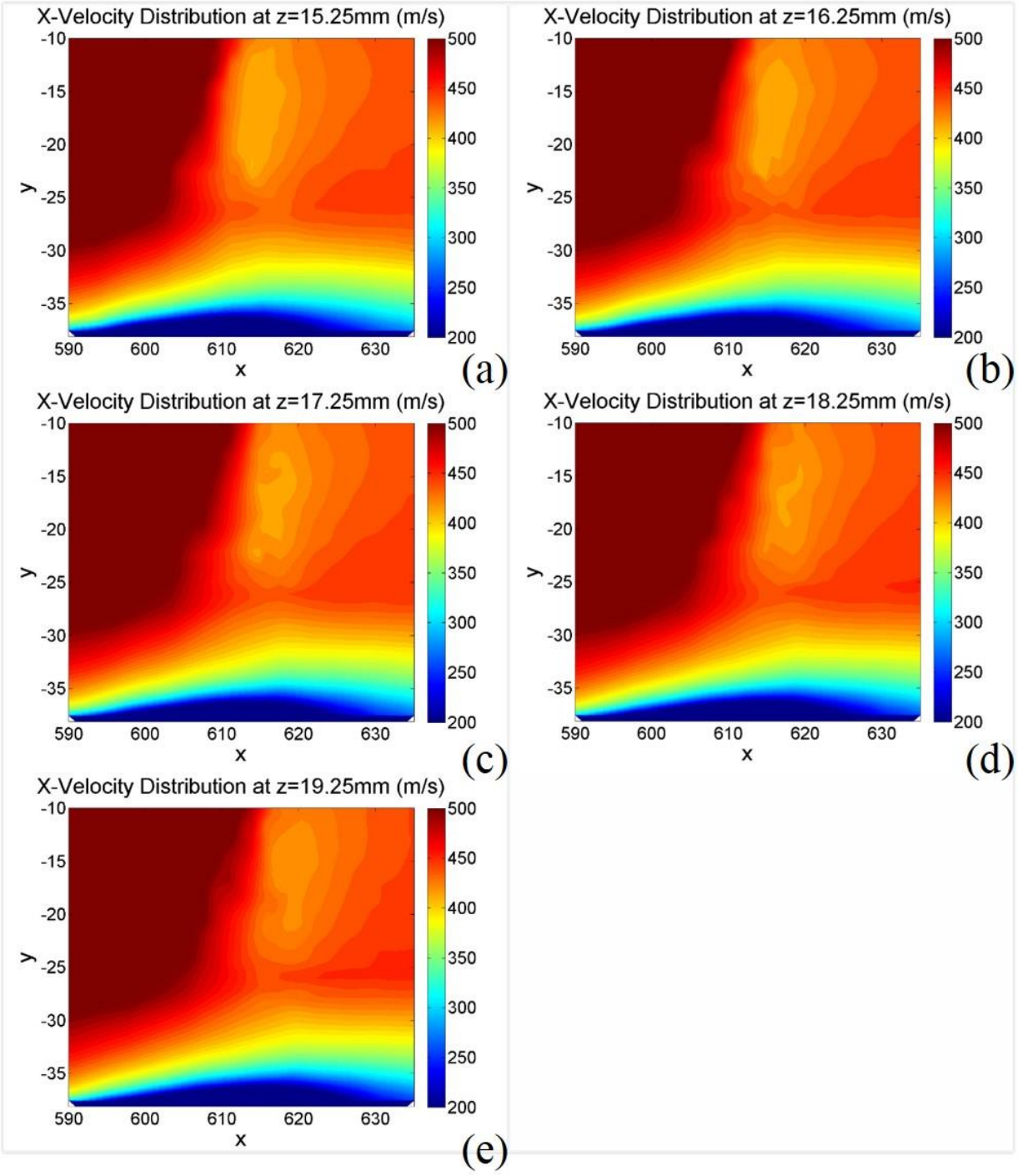


Figure 6.35: X-Velocity Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

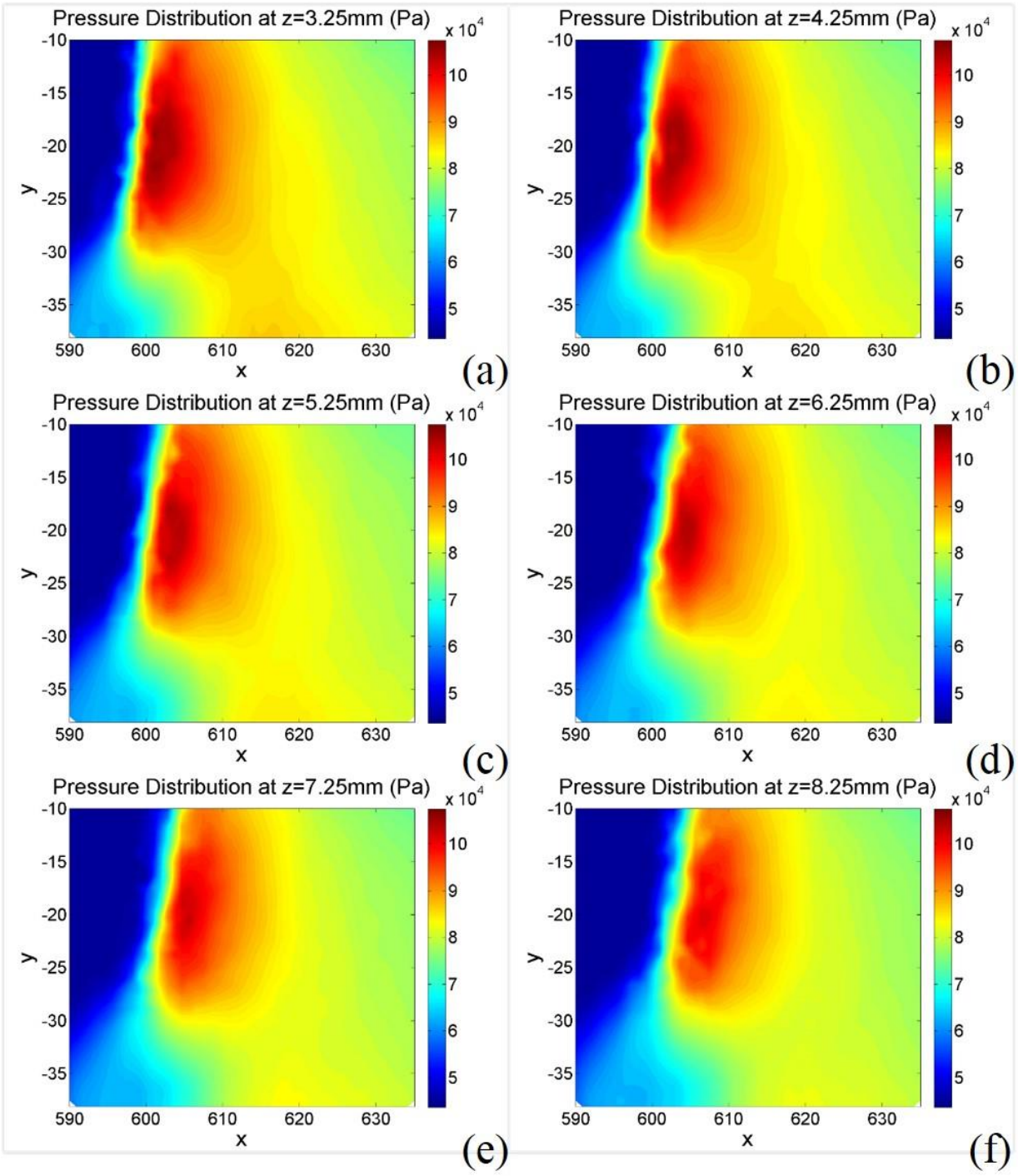


Figure 6.36: Pressure Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

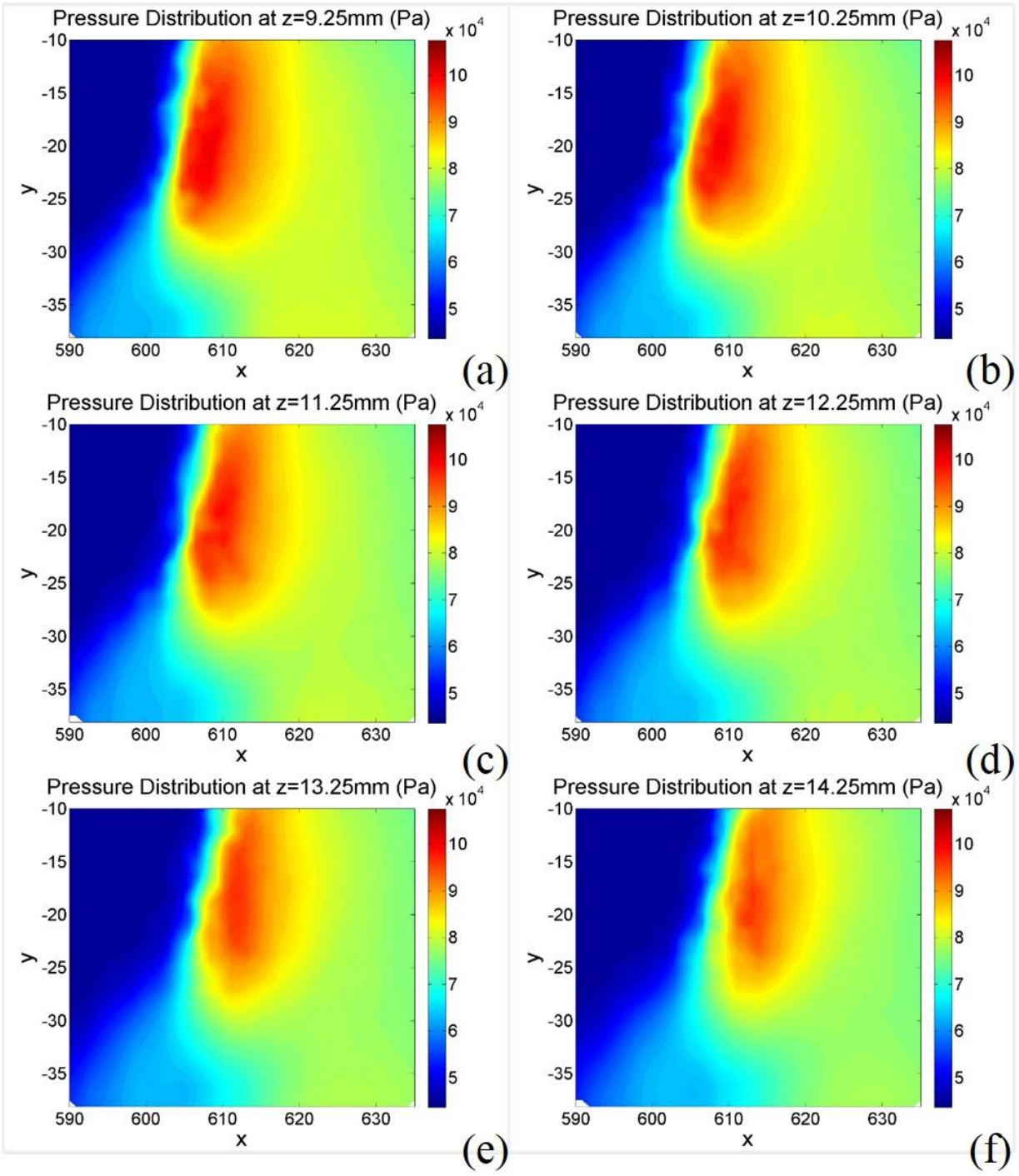


Figure 6.37: Pressure Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

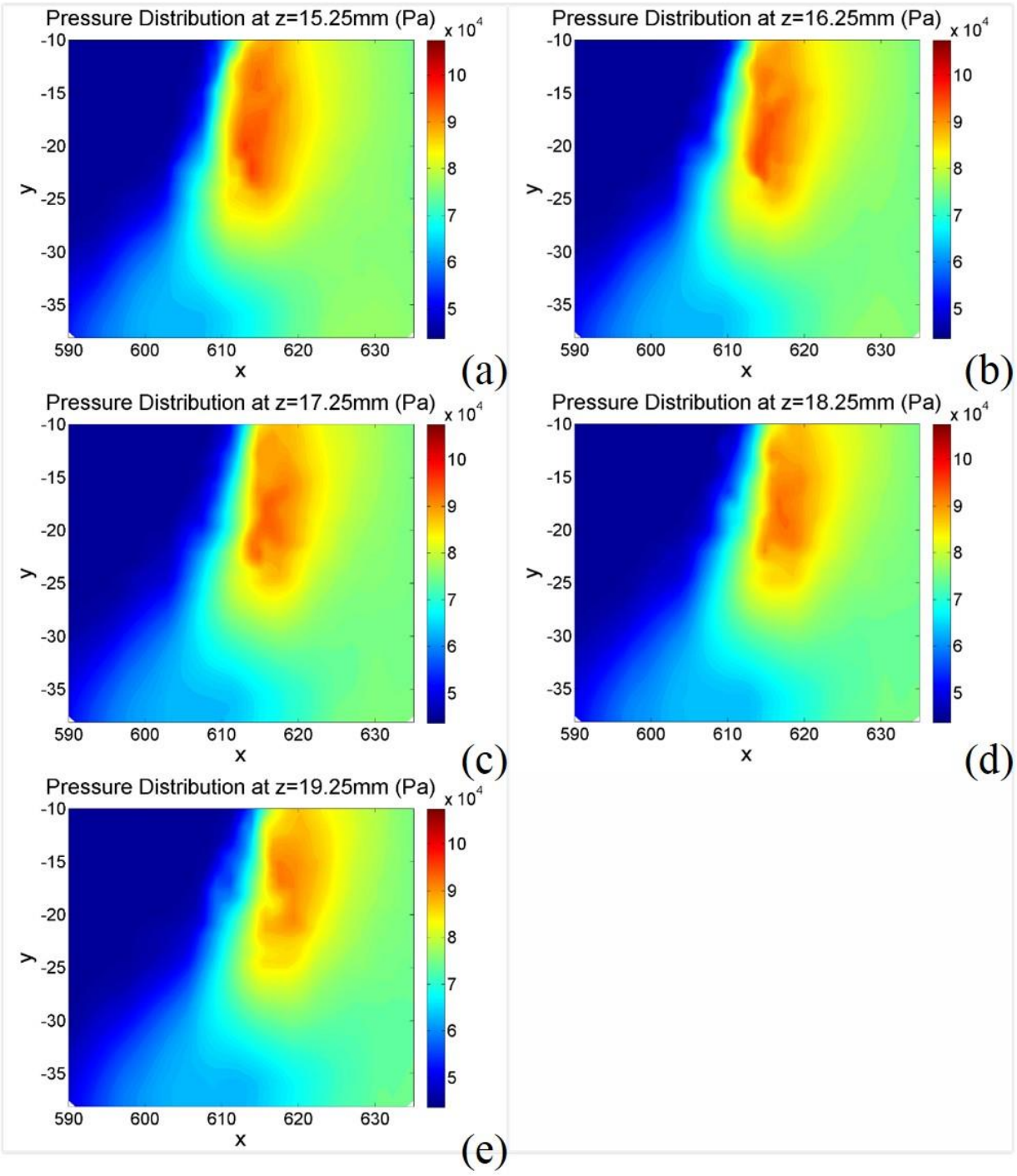


Figure 6.38: Pressure Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

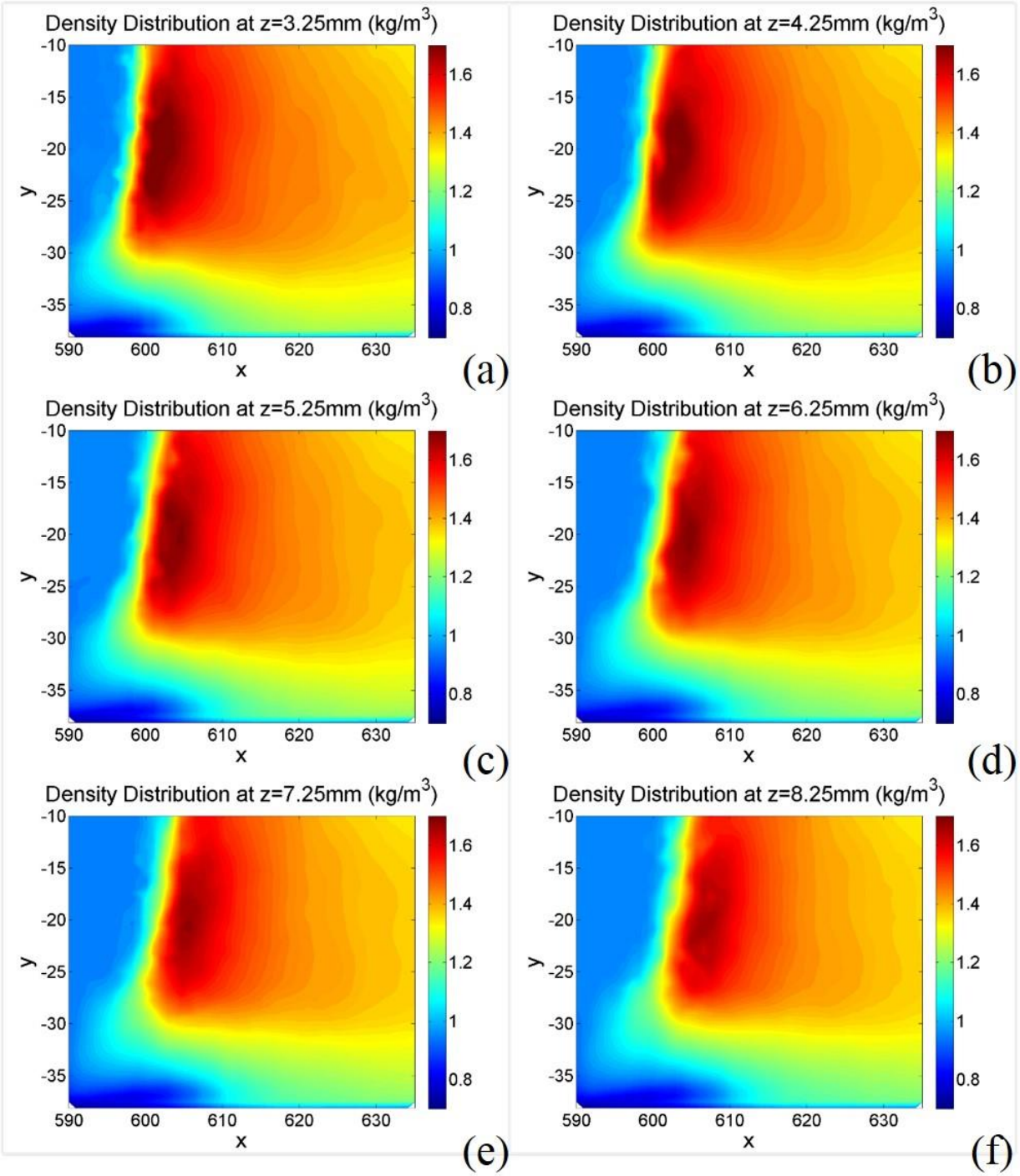


Figure 6.39: Density Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

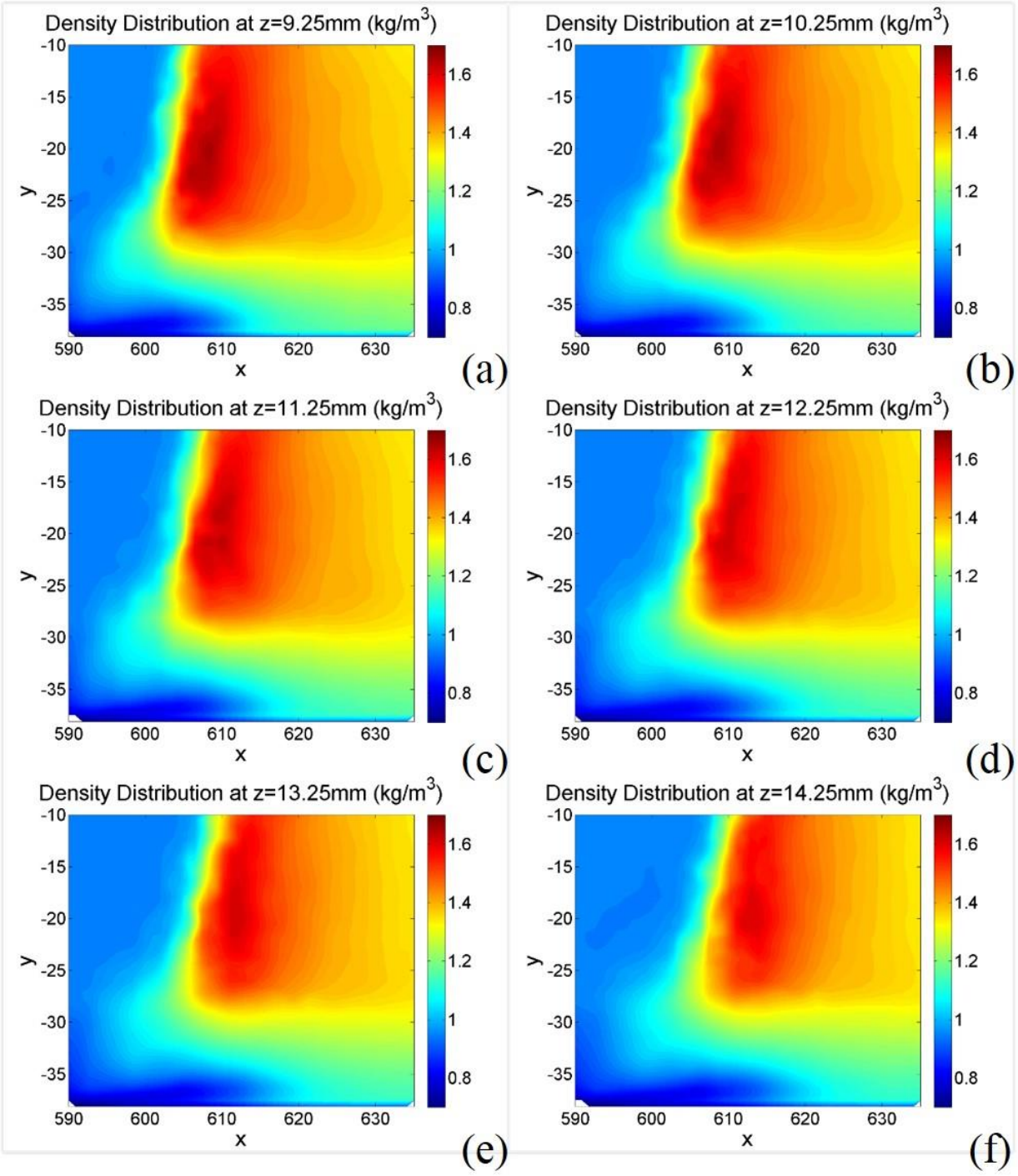


Figure 6.40: Density Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

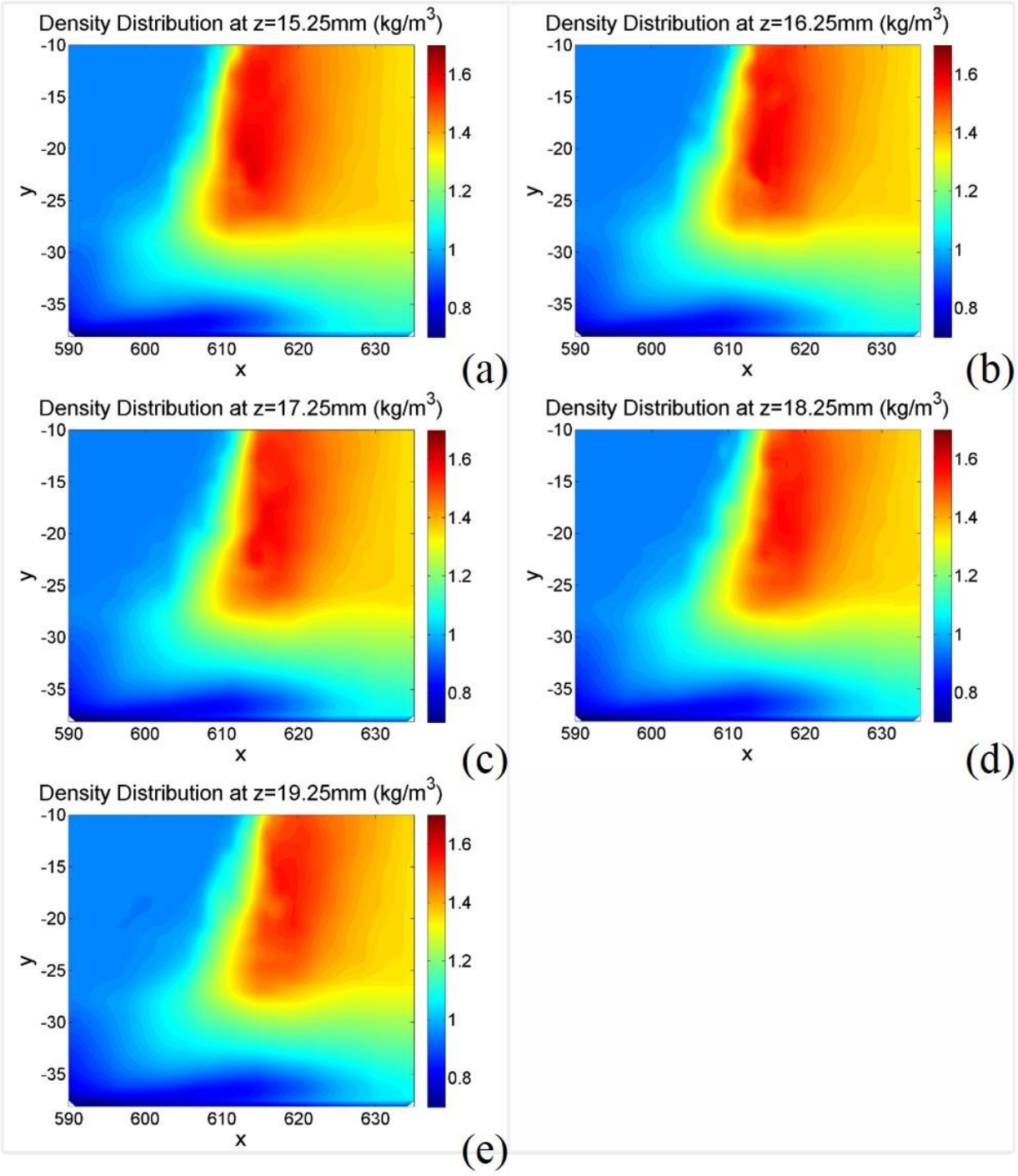


Figure 6.41: Density Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

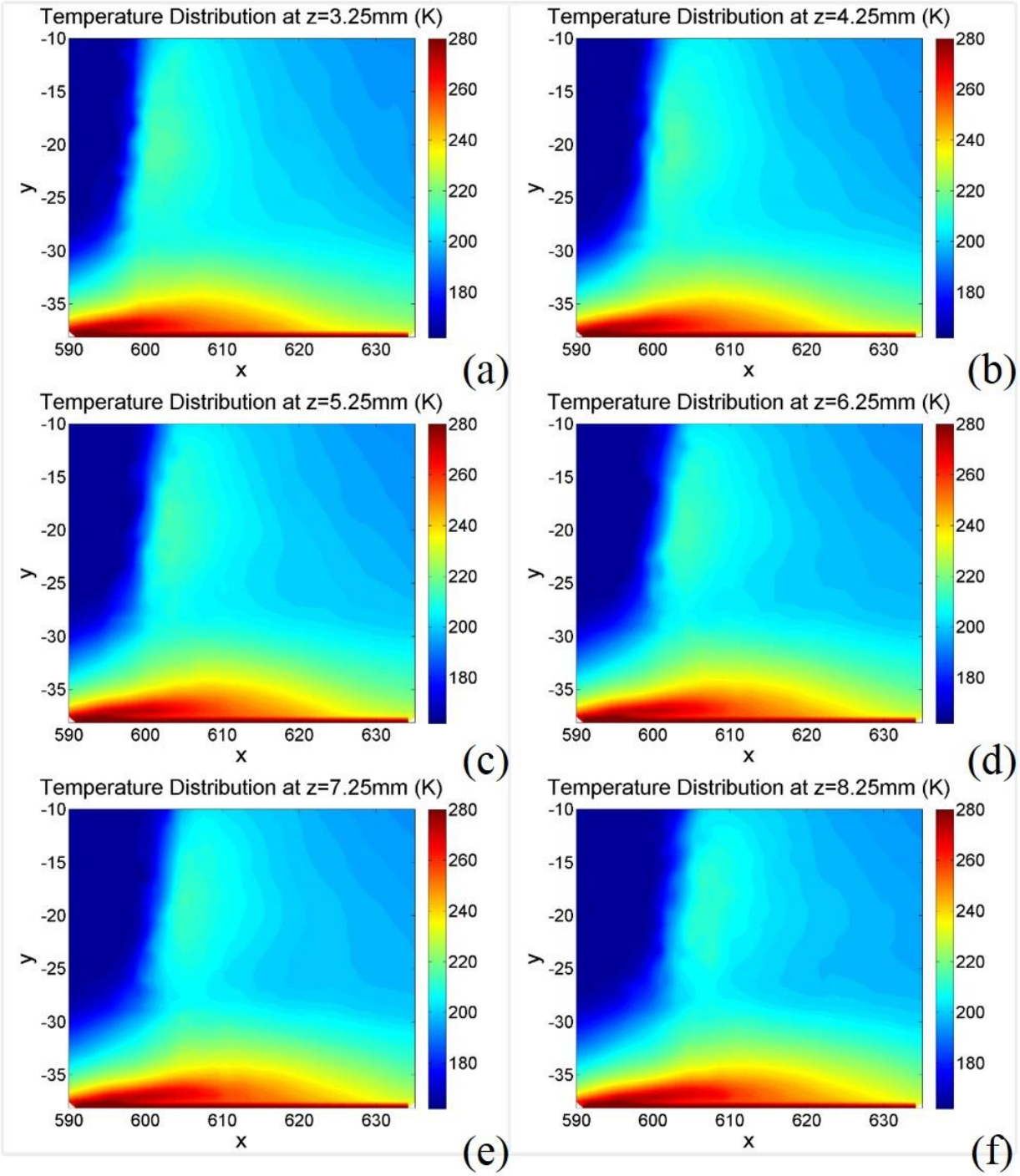


Figure 6.42: Temperature Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 3.25mm, (b) Z= 4.25mm, (c) Z= 5.25mm, (d) Z= 6.25mm, (e) Z=7.25mm, (f) Z= 8.25mm

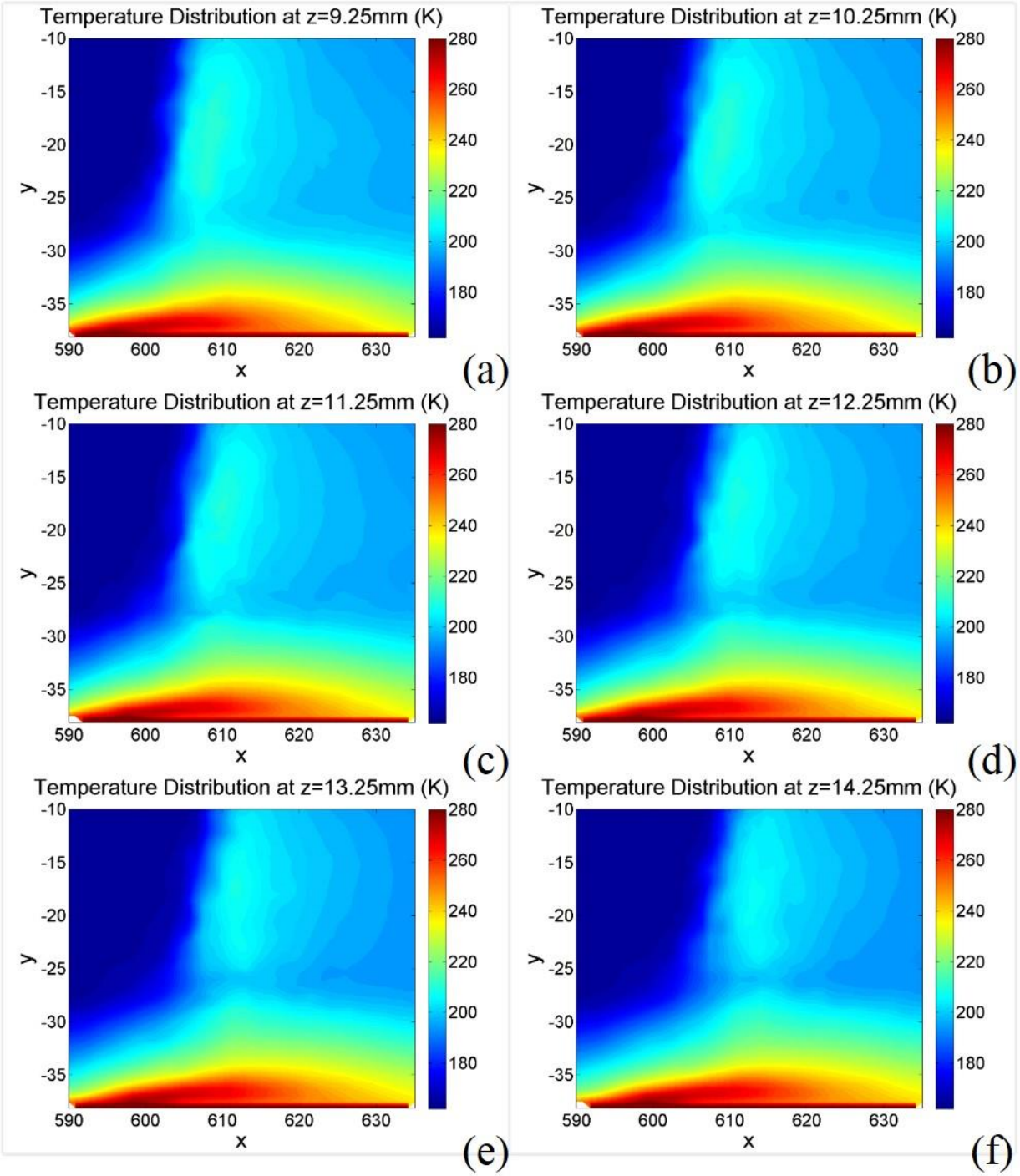


Figure 6.43: Temperature Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 9.25mm, (b) Z= 10.25mm, (c) Z= 11.25mm, (d) Z= 12.25mm, (e) Z=13.25mm, (f) Z= 14.25mm

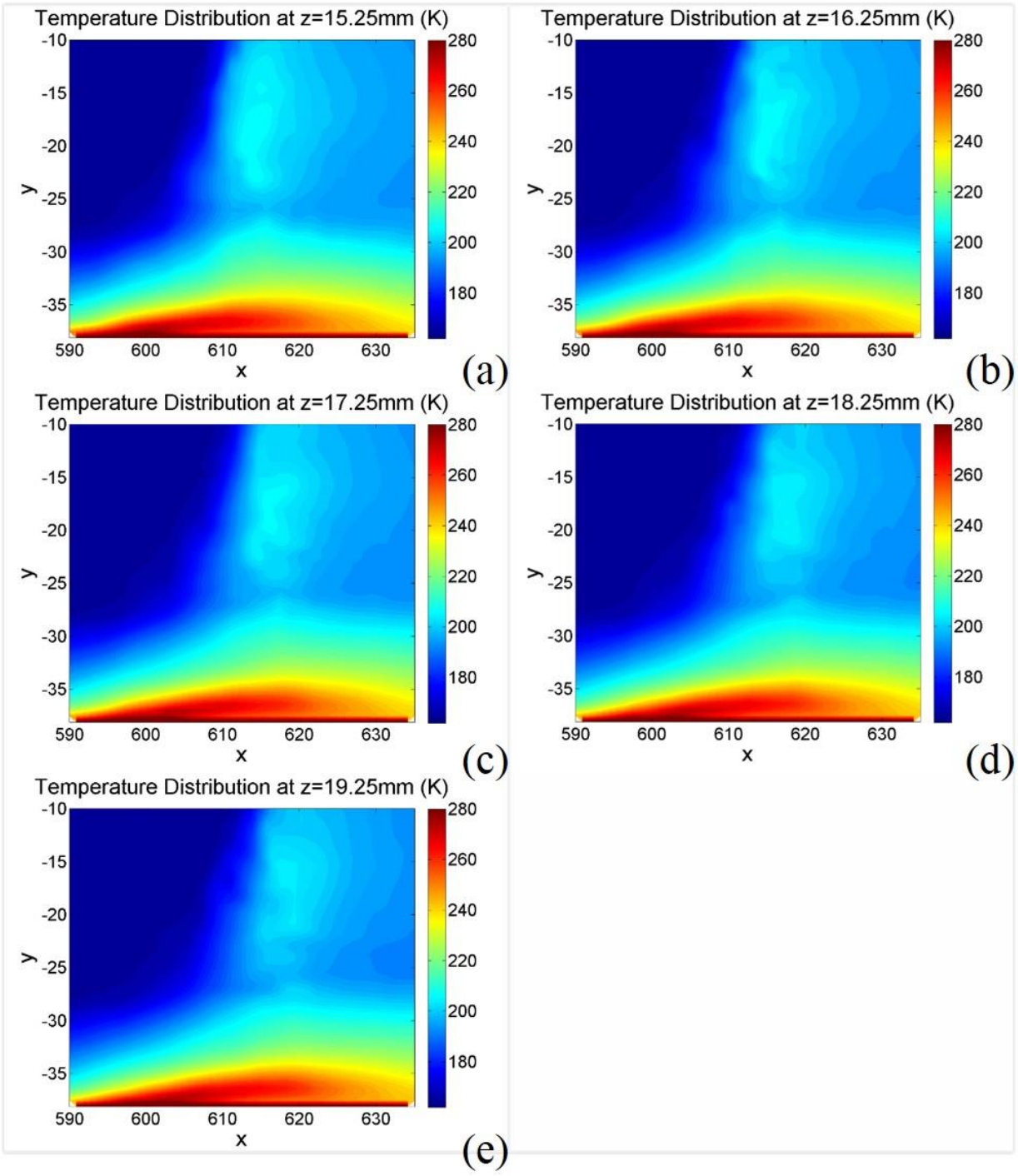
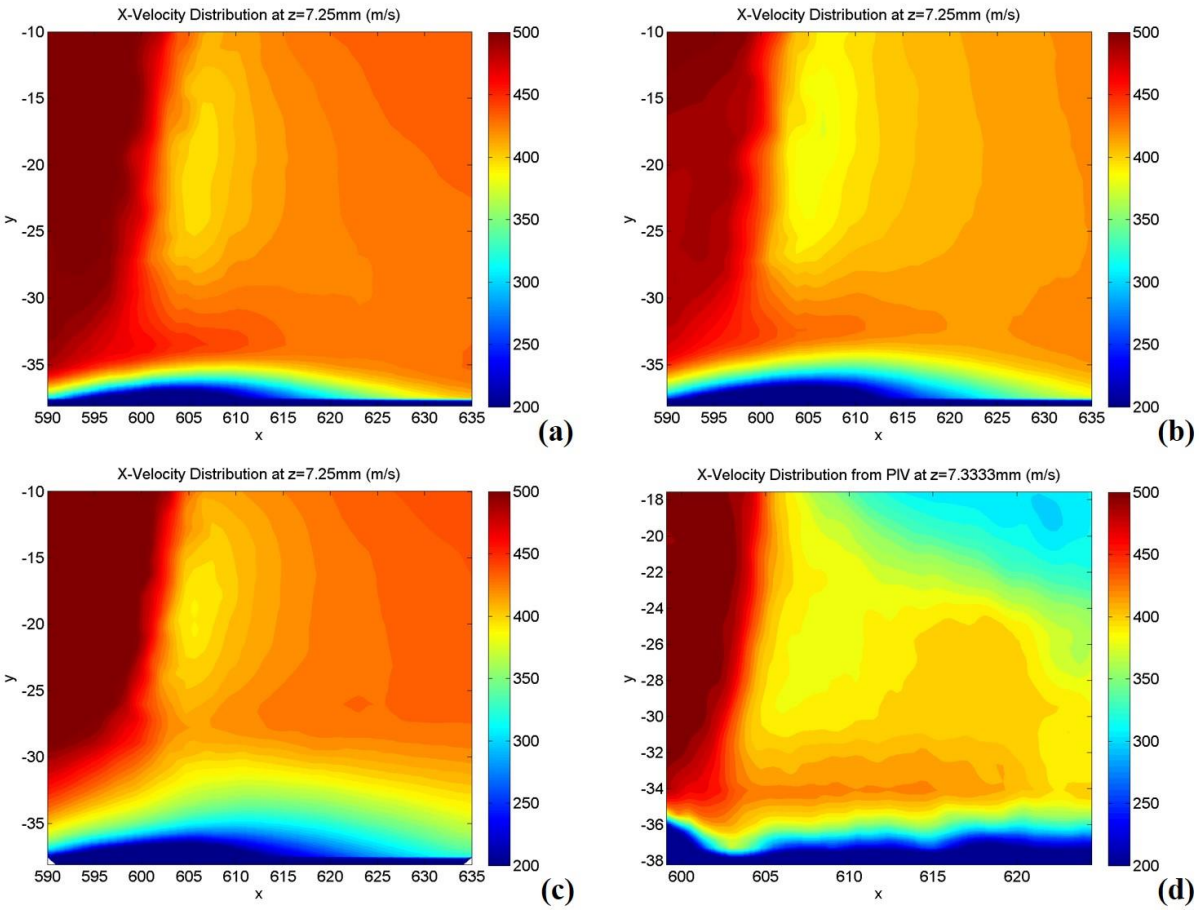


Figure 6.44: Temperature Distribution in the SWBLI of the non-obstacle surrounding Case Simulation Using Wilcox $k-\omega$ Model: (a) Z= 15.25mm, (b) Z= 16.25mm, (c) Z= 17.25mm, (d) Z= 18.25mm, (e) Z=19.25mm

6.4 Comparison with Wind Tunnel Experiments



(a) In-Tunnel Case using Wilcox k-ω Model ;
(b) In-Tunnel Case using Menter SSTModel
(c) Farfield Case using Wilcox k-ω Model;
(d) Plenoptic PIV Method

Figure 6.45: The Comparison of X-Velocity Distribution at Z = 7.25mm between Numerical Simulations and the Plenotic PIV Method

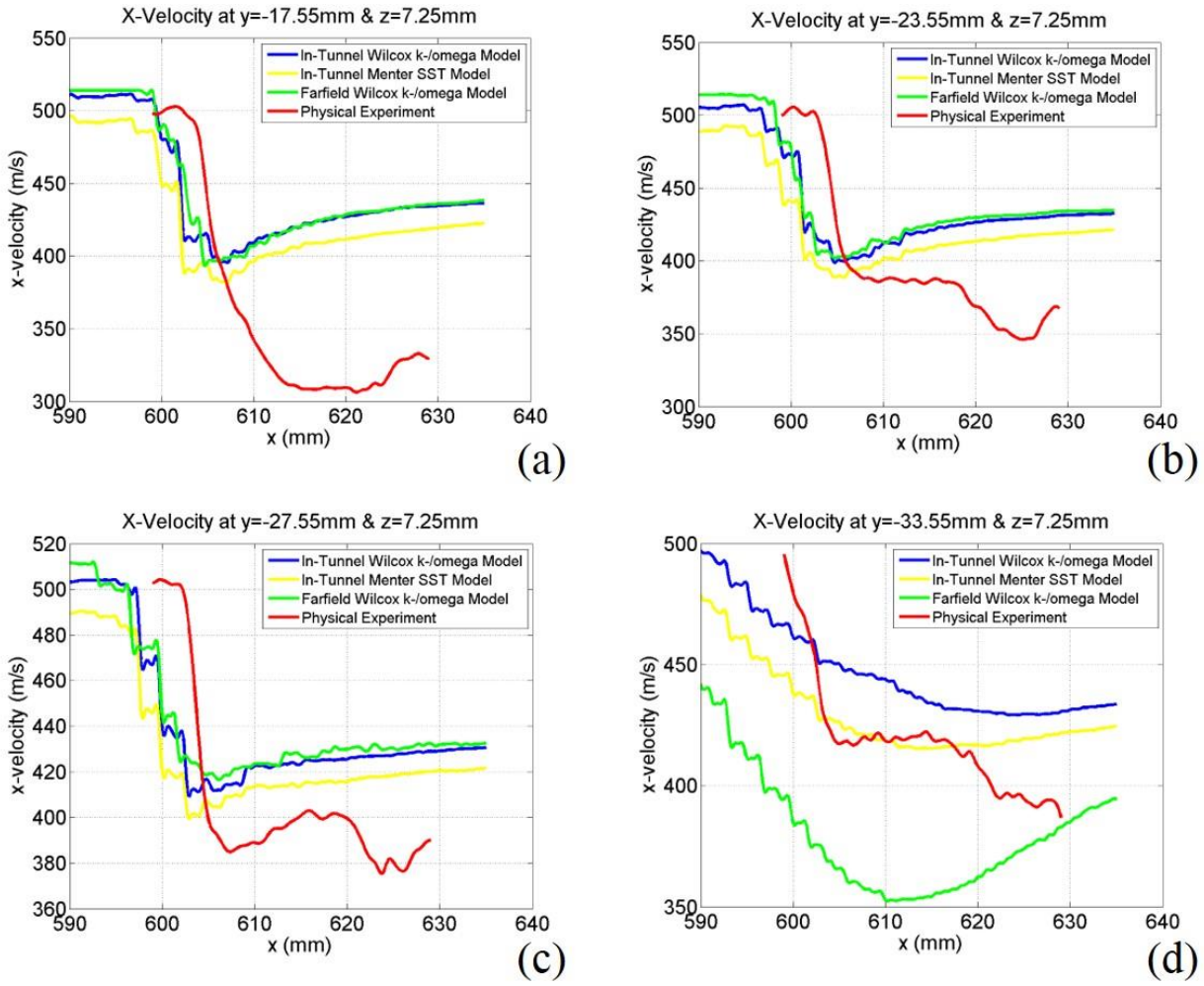
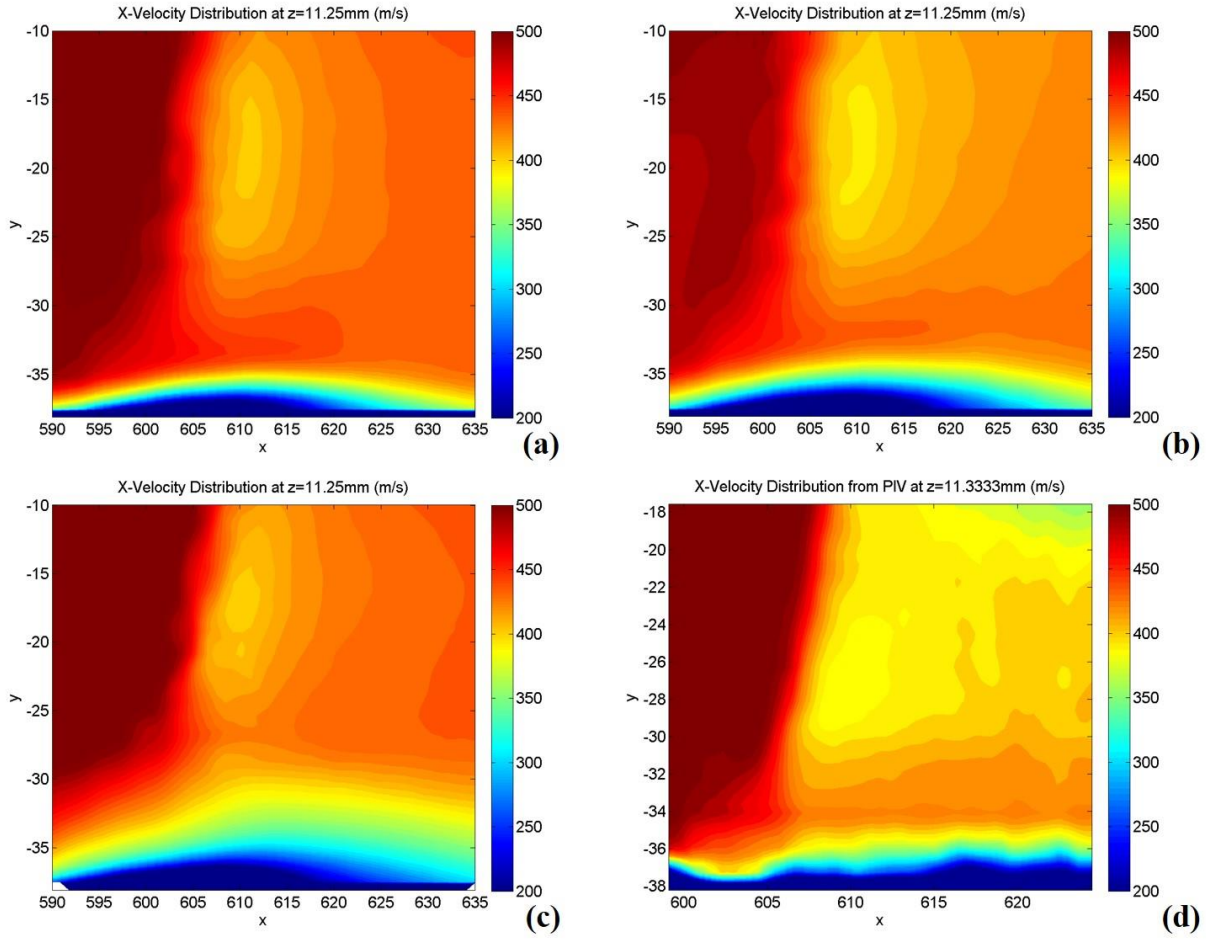


Figure 6.46: The Line Cut Comparison of X-Velocity between Numerical Simulations and the Plenotic PIV Method: (a) Y= -17.55mm & Z= 7.25mm, (b) Y= -23.55mm & Z= 7.25mm, (c) Y= -27.55mm & Z= 7.25mm, (d) Y= -33.55mm & Z= 7.25mm



(a) In-Tunnel Case using Wilcox k-ω Model ; (b) In-Tunnel Case using Menter SSTModel
(c) Farfield Case using Wilcox k-ω Model; (d) Plenoptic PIV Method

Figure 6.47: The Comparison of X-Velocity Distribution at Z = 11.25mm between Numerical Simulations and the Plenotic PIV Method

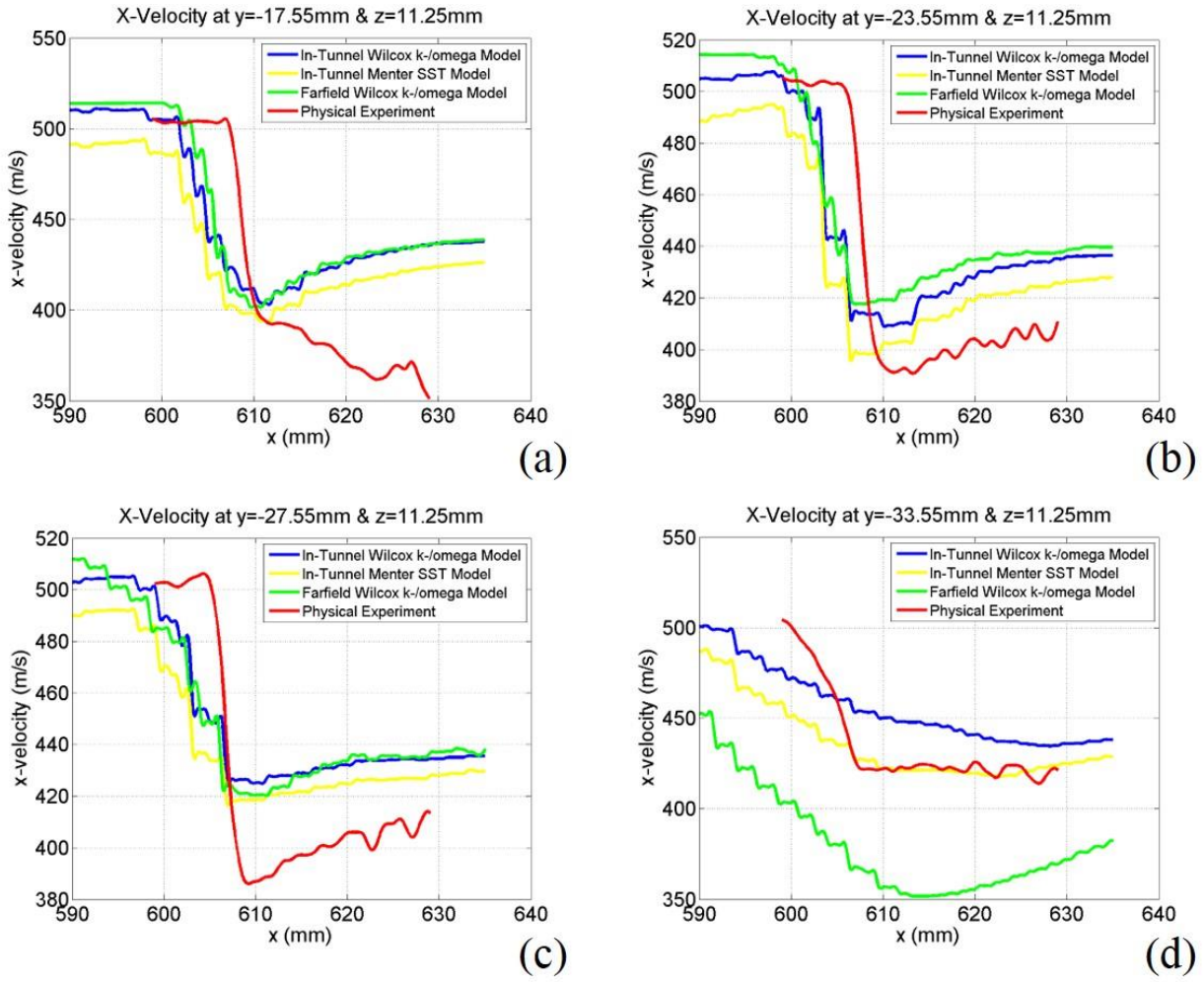
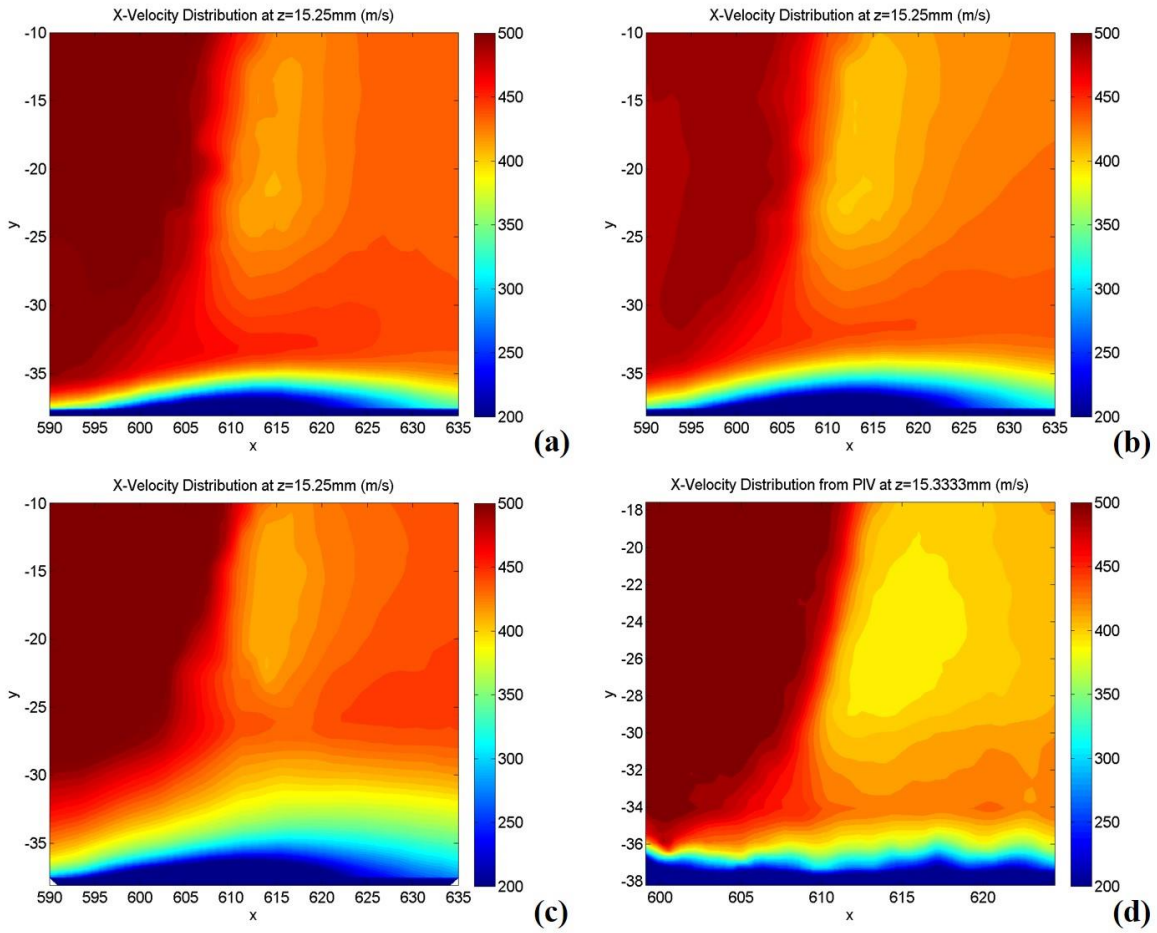


Figure 6.48: The Line Cut Comparison of X-Velocity between Numerical Simulations and the Plenotic PIV Method: (a) Y= -17.55mm & Z= 11.25mm, (b) Y= -23.55mm & Z= 11.25mm, (c) Y= -27.55mm & Z= 11.25mm, (d) Y= -33.55mm & Z= 11.25mm



(a) In-Tunnel Case using Wilcox k-ω Model ;
(b) In-Tunnel Case using Menter SSTModel
(c) Farfield Case using Wilcox k-ω Model;
(d) Plenoptic PIV Method

Figure 6.49: The Comparison of X-Velocity Distribution at Z = 15.25mm between Numerical Simulations and the Plenotic PIV Method

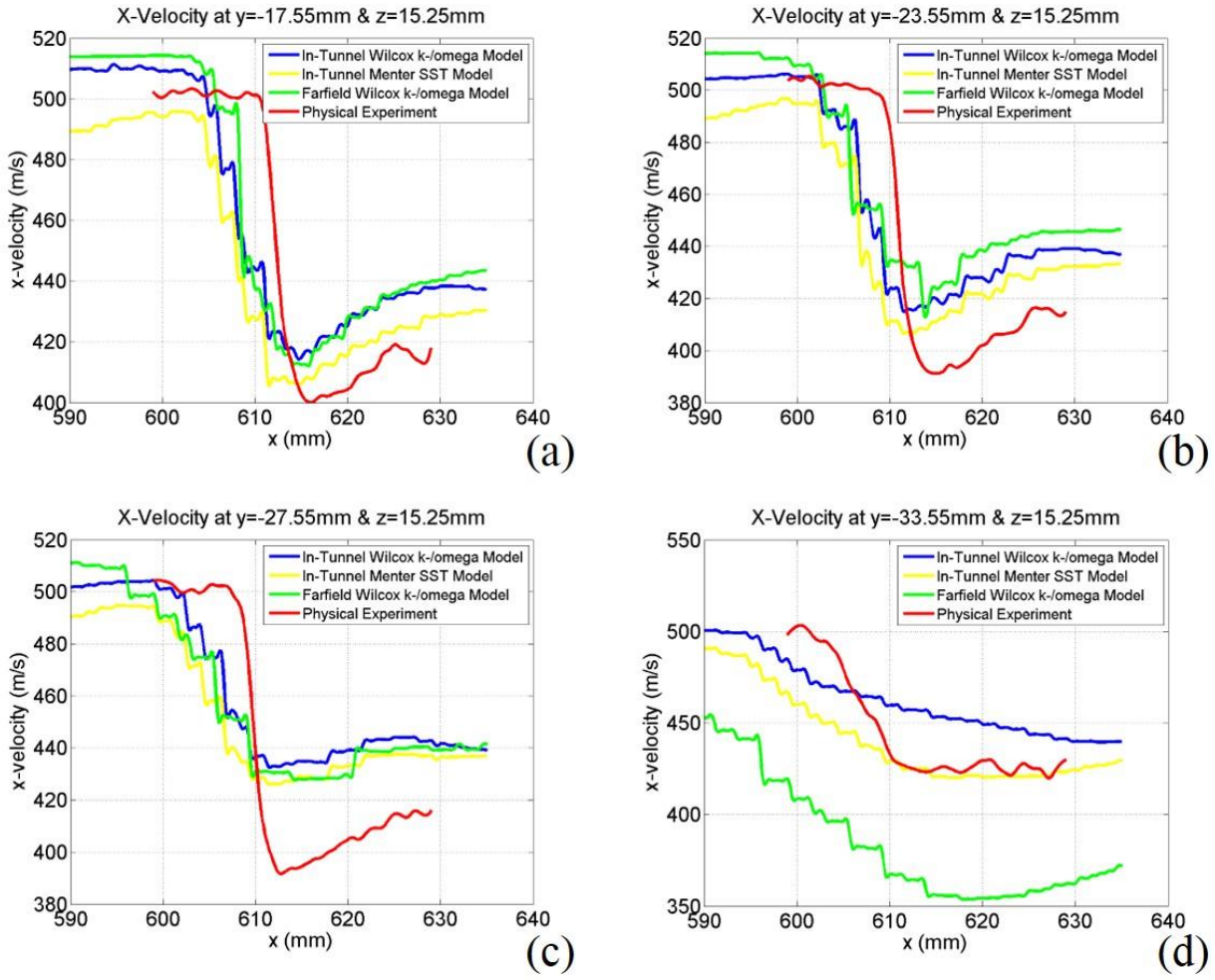


Figure 6.50: The Line Cut Comparison of X-Velocity between Numerical Simulations and the Plenotic PIV Method: (a) Y= -17.55mm & Z= 15.25mm, (b) Y= -23.55mm & Z= 15.25mm, (c) Y= -27.55mm & Z= 15.25mm, (d) Y= -33.55mm & Z= 15.25mm

Figure 6.45, Fig. 6.47, and Fig. 6.49 present the comparison of the x component of the velocity between numerical simulations and the Plenotic PIV method at three different z-axis locations: at 7.25mm, 11.25mm, and 15.25mm. Because of the structure of the data set from the physical experiment, data at the closest location is applied in the comparison. The low-velocity region behind the main shock wave from the Plenotic PIV result is larger, and also has a smaller velocity

value than that from both two numerical simulations' results. The thickness of the boundary layers from the experiments using the Plenotic PIV method has some small fluctuations but is fairly constant across the cuts. However, the thickness of the boundary layers in the two numerical simulations have an increasing portion, which is closer to the SWBLI characteristics established in Fig. 1.2. This difference might be caused by the limitation of velocity measurement at the near wall region of the Plenotic PIV method. The difficulties and inaccuracy of capturing the flow properties near the wall because of the usage of cameras have been mentioned in the previous study [64]. Compared with the numerical simulation utilizing the Wilcox $k-\omega$ model, the low-velocity region behind the main shock wave and the subsonic region at the bottom are larger in the numerical simulation utilizing the blended $k-\varepsilon, k-\omega$ model. Moreover, the velocity gradient is also clearer than that in the simulation utilizing the Wilcox $k-\omega$ model. It is obvious that the SWBLI structures are different between the farfield and in-tunnel simulations. That means that the results from the wind tunnel experiments would have some deviation to a physical atmosphere flight. Compared with the numerical simulation of the in-tunnel case, the numerical simulation results of the farfield case has a larger subsonic region near the bottom surface. This points out that the extreme high-temperature region near the sidewall in a physical flight might be larger than that in the wind tunnel experiments. Hence, the thermal load issue caused by the SWBLI on the aircraft's surface might be more serious than the experimental expectation.

As shown in Fig. 6.46, Fig. 6.48, and Fig. 6.50, the line cuts of the x-velocity are used to better illustrate the comparison between numerical simulations and the wind tunnel experiment. Figure

6.46, Fig. 6.48, and Fig. 6.50 show the x-velocity on four y-location (y= -17.55mm, -23.55mm, -27.55mm, and -33.55mm) from Fig. 6.45, Fig. 6.47, and Fig. 6.49, respectively. The results at y= -17.55mm, -23.55mm, and -27.55mm show a good similarity on shock wave determination between numerical simulations and the physical experiment. However, a big difference occurs at -33.55mm because of the instability of the slip surface region and the difference in the λ-structure between cases. As far away from the near surface region, the simulation results become closer to the result of the wind tunnel experiment. This indicates the turbulence modeling needs to be improved in the near wall region and the boundary layer grid needs to be modified. The result in the physical experiment shows a larger low velocity region downstream of the main oblique shock wave. Moreover, the x-velocity downstream of the shock wave in the numerical simulations is larger than that in the wind tunnel experiment. The reason is that the sharp surface of the shock wave has not been captured away from the fin surface and the bottom surface of the test section. This leads to an inaccuracy on calculating the energy dissipation cross the shock wave so that the x-velocity decreases less than that in the physical experiments result through the shock wave. The grid resolution is required to improve in the further in order to capture the detail of recirculation region, and shock face.

The primary separation distance l_{sep}, which measured by the distance between two bottom legs in the λ-structure as shown in Fig.1.2, is one of the important indicators for the SWBLI structures. The primary separation distance from the result of both the numerical simulation and the physical experiments has been measured and listed in Table 6.1. The λ-structure enlarges on

the z-direction away from the fin. The primary separation distance from the in-tunnel case using the blended $k-\varepsilon, k-\omega$ model is larger than that from the in-tunnel case using Wilcox $k-\omega$ model, which represents that the result of the in-tunnel case using the blended $k-\varepsilon, k-\omega$ model has a larger λ -structure. The primary separation distance measured in the numerical simulations is bigger than that in the physical experiments. Notice that the λ -structure in the farfield case using Wilcox $k-\omega$ model is not obvious enough to measure the primary separation distance.

Cases	Z-Location		
	Z = 7.25mm	Z = 11.25mm	Z = 15.25mm
In-Tunnel Case using Wilcox k-ω Model	7.6	12.9	16.5
In-Tunnel Case using Menter SSTModel	13.5	17.1	22.3
Plenoptic PIV Method	4.6	7.7	11.5

Table 6.1: The Primary Separation Distance of the SWBLI Structure (mm)

6.5 Explication of the Additional Shock Wave in the Test Section

An unexpected oblique shock appears in the numerical simulation and is shown in Fig. 6.51. The wave detection method [70], which is based on searching the maxima of the density gradient, is applied to determine the location of the shock wave. The iso-surface for the shock wave and the criteria of the filter are expressed as

$$Iso\text{-}surface: \quad \frac{d^2\rho}{dn^2} = \nabla\left(\nabla\rho \cdot \frac{\vec{V}}{|\vec{V}|}\right) \cdot \frac{\vec{V}}{|\vec{V}|} = 0 \tag{5.1}$$

$$Filter: \quad \frac{d\rho}{dn} = \nabla\rho \cdot \frac{\vec{V}}{|\vec{V}|} > \varepsilon \tag{5.2}$$

The iso-surface corresponds to the maxima or minima of the streamwise density gradient. If the filter is larger than zero, it represents a shock wave. If the filter is smaller than zero, it represents an expansion wave. However, some smooth flow regions also satisfy the condition that the second derivatives of density equal to zero, and the first derivatives of density are greater than zero. Therefore, the perturbation value ε should be large enough to remove the smooth flow regions and small enough to preserve the weak shock wave. This wave detection method is also used in the previous analysis of simulation results.

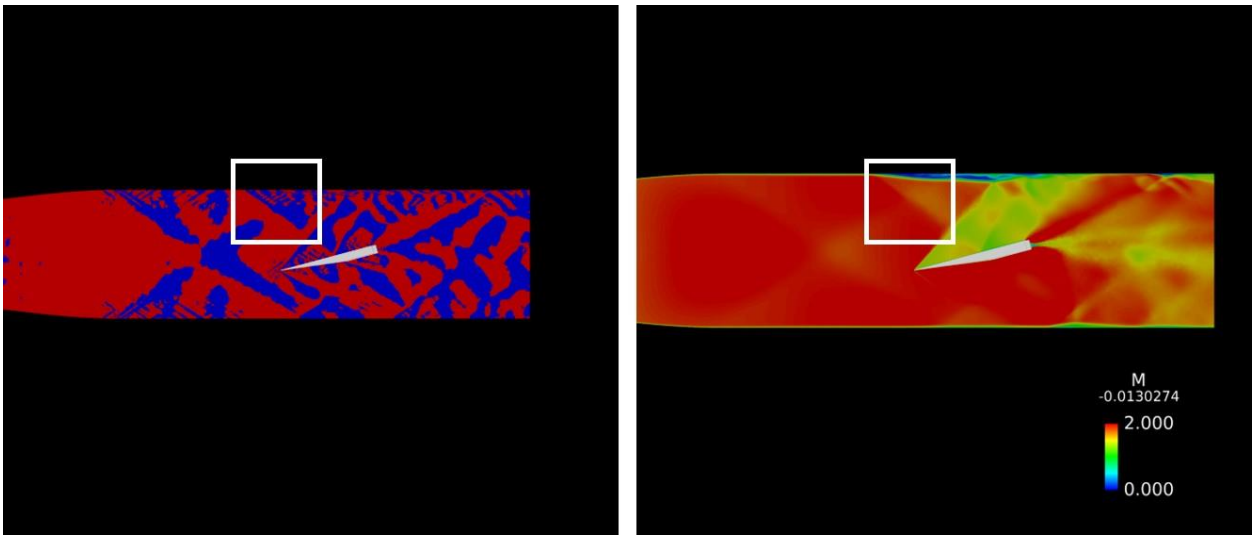


Figure 6.51: The Additional Shock Wave Created by the Change of Boundary Layer Thickness

From the result of shock wave detection (Fig. 6.51 Left), the additional shock wave is not generated by the reflection of oblique shock waves. Although the additional shock wave is weaker than the strong oblique shock generated by the fin, it is more obvious than the weak shock waves in the test section (Fig. 6.51 Right) and somewhat affects the strong oblique shock, which is the object of the research.

In the test section, the strong oblique shock wave is formed at the leading edge of the fin and

impinges the sidewall of the test section in its outward propagation. The thickness of the boundary layer on the test section's sidewall is altered due to the shock wave impingement as seen in Fig. 6.51, and the increase of the boundary layer thickness translates forward while the subsonic region near the sidewall of the tunnel enlarges. As the boundary layer thickness upstream of the main shock, it turns the flow and generates the additional weak shock wave. This phenomenon shows the complexity of the supersonic flow experiments in the wind tunnel. The existence of the additional shock waves indeed influence the result of the experiment, but it is easy to be ignored. The CFD study provides a direct way to visualize the details happening in the test section, which can be observed in the physical experiment.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

7.1.1 Shock Wave-Turbulent Boundary Layer Interaction (SWBLI)

The characteristics of the shock wave turbulence boundary layer interaction have been obtained and analyzed based on the numerical simulations. The λ-structure composed by the main shock wave, the separation shock wave, and the rear shock wave is observed in the simulation results, of which the size is increasing with the outward propagation of the shock wave. Meanwhile, the slipline region extends along the flow direction with the with the propagation of the shock wave. The separation bubble and pressure "jet" are presented in the total pressure analysis. The numerical simulation shows an extreme high-temperature region near the sidewall, which would cause the thermal load issue on the supersonic vehicle's surface. The numerical simulations also indicate additional interactions between the main shock wave and the boundary layer, which was not observed in the physical experiments due to the placement of the cameras.

7.1.2 Turbulence Model

The Wilcox $k-\omega$ model and the blended $k-\varepsilon, k-\omega$ model have been applied in the numerical simulation processes successfully. The blended $k-\varepsilon, k-\omega$ model shows a better capability on describing the behaviors of turbulence in the supersonic wind tunnel due to its advanced region switch criteria. However, the Wilcox $k-\omega$ model expresses a better stability in operating the complex vortexes in the wind tunnel. The blended $k-\varepsilon, k-\omega$ model needs to run more iterations to be convergent; the sensitivity of the control parameters make it easy to the failure. Ignoring the loss of some accuracy, the $k-\varepsilon$ term could be eliminated for the blended $k-\varepsilon, k-\omega$ model in order to obtain the improvement of the stability. The x-velocity simulation results by using two turbulence models are compared with the result from the wind tunnel experiment. The x-velocity by using the blended $k-\varepsilon, k-\omega$ model is closer to that measured in the physical experiment compared with the x-velocity by using Wilcox $k-\omega$ model. However, both models need to be improved on modeling the near wall region flow.

7.2 Future Works

The numerical simulation results capture the general characteristics of the SWBLI but the volume grid resolution need to be improved in order to capture the detail of the recirculation region, sharp surface of the oblique shock wave, and the λ-structure. The source terms should be discussed if they affect too much on the determination of the shock wave location.

When the experimental object is placed in the wind tunnel, it will produce oblique shock

waves which might influence and alter the properties of the incoming supersonic flow. Further, it is impossible to eliminate redundant oblique shock waves entirely, and the propagation and reflection of those oblique shock waves cause the supersonic flow behaviors in the wind tunnel to be more complex than that in the free stream flow. For this reason, it is significant to predict the effect of the tunnel's surface shape on the generation of shock waves, which could be applied to improve the accuracy of investigations related to supersonic and hypersonic flows.

The impingement and coalescence of oblique shock waves have been observed in the numerical simulation results. The 2D-structure and 3D-structure of shock wave-turbulent boundary layer interactions have been examined in previous research. However, the features of the interaction between two oblique shock waves and boundary layers are rarely investigated. Furthermore, the studies about the supersonic flow behaviors in the area where several different shock waves and the boundary layer interact together should be encouraged.

The Plenoptic particle image velocimetry (PIV) method utilizes seeding particles with a mean diameter of 0.3 μm to capture the characteristics of the flow field. However, in the physical experience, it is obvious that seeding particles adhere and accumulate on the surfaces of sidewalls and the model. The micron-scale particles are large enough to change the micro-shape of the surfaces and have an effect on the formation of the boundary layers. The CFD provides a good way to establish the model and simulate in the expected condition helping the Plenoptic particle image velocimetry (PIV) method overcome the limitation of cameras. In order to enhance the comprehension of shock wave-turbulent boundary layer interactions, more work on the

relationship between the boundary layer deformation and the micro-shape and micro-structure change should be developed in the future.

The supersonic flow shows its complexity when the shock wave and the boundary layer interact with each other in the near wall region. Therefore, modeling the behaviors of the near wall region flow is a big challenge for the modern turbulence closure. Even though the CFD techniques continue to advance, modern turbulence models, which play an important role in numerical simulation, need better techniques to improve the predictability of numerical methods.

Reference

[1] M.S. Holden. (1986, January). A review of aerothermal problems associated with hypersonic flight. *AIAA 24th Aerospace Sciences Meeting, AIAA Paper 86-0267.*

[2] I. Nompelis, G. Candler, M.Holden, & T. Wadhams. (2003, January). Real Gas Effects on Hypersonic Shock Wave Laminar Boundary Layer Interactions. *AIAA 41th Aerospace Sciences Meeting, AIAA Paper 2003-443*.

[3] Antonio Ferri. (1939). Experimental Results with Airfoils Tested in the High-Speed Tunnel at Guidonia.

[4] David Dolling. (2000). 50 Years of Shock Wave/Boundary Layer Interaction Research - What Next? *Fluids 2000 Conference and Exhibit.*

[5] J. Anderson. (1989). *Hypersonic and High Temperature Gas Dynamics.* American Institute of Aeronautics and Astronautics.

[6] J. Anderson, & R. Passman. (2014). *X-15: The World's Fastest Rocket Plane and the Pilots Who Ushered in the Space Age.* Zenith Press.

[7] Nishul Arora, Mohd Y. Ali, Yang Zhang, & Farrukh S. Alvi. (2015, January). Shock-Boundary Layer Interaction due to a Sharp Unswept Fin in a Mach 2 Flow. *AIAA SciTech*, pp. 1-11.

[8] V.S. Avduevskii, & K.I. Medvedev. (1976, January). Physical Properties of the Flow in the Separation Region for Three-Dimensional Interaction of a Boundary Layer with a Shock Wave. *Fluid Dynamics, 2(1),* pp. 17-22.

[9] F. Alvi, & G. Settles. (1990). Structure of Swept Shock Wave/Boundary-Layer Interactions Using conical Shadowgraphy. *AIAA Paper 1990-1644*.

[10] D.R. Chapman, D.M. Kuehn, & H.K. Larson. (1958). Investigation of Separated Flow in Supersonic and Subsonic Streams with Emphasis on the Effect of Transition. *Report 1356, NACA*.

[11] F.L Young, L.G. III Kaufman, & R.H. Korkegi. (1968, Decemeber). Experimental Investigation of Interactions between Blunt Fin Shock Waves and Adjacent Boundary Layers at Mach Number 3 and 5. *ART Technical Report ARL 68-0214*.

[12] D.M. Voitenko, A.I. Zubkov, & Y.A. Panov. (1967). Exitence of Supersonic Zones in Three-Dimensional Separation Flows. *Mekhanika Zhidkosti I Gaza, 2(1),* pp. 20-24.

[13] D. S. Dolling, & S.M. Bogdonoff. (1982, December). Blunt Fin-Induced Shock Wave/Turbulent Boundary Layer Interactions. *AIAA Journal, 20(12),* pp. 1674-1680.

[14] L. II Kaufman, R. Korkegi, & L. Morton. (1972, August). Shock Impingement Caused by Boundary Layer Separation ahead of Blunt Fins. Technical Report ARL 72-0118.

[15] O. Ozcan, & M. Holt. (1984, May). Supersonic Separated Flow Past a Cylindrical Obstacle on a Flate Plate. *AIAA Journal, 22(5),* pp. 611-617.

[16] R. Sedney, & C.W. Jr. Kitchens. (1975). The Structure of the Three-Dimensional Separated Flows in Obstacle-Boundary Layer *Interactions. AGARD-CO-168, 37,* pp. 1-15.

[17] D.M. Voitenko, A.I. Zubkov, & Y.A. Panov. (1966, January). Supersonic Gas Flow Past a Cylindrical Obstacle on a Flate Plat. *Mekhanika Zhidkosti I Gaza, 1(1),* pp. 84-88.

[18] Gary S. Settles, & David S. Dolling. (1992). Swept Shock Wave/Boundary-Layer Interactions. *Tactical missile aerodynamics: General Topics, 141,* pp. 505-574.

[19] E.A. Jr. Price, & R.L. Jr. Stallings. (1976, January). Investigation of Turbulent Separated Flows in the Vicinity of Fin-Type Protuberances at Supersonic Mach Number. *Technical Note NASA D-3804, Langley Research Center*.

[20] Patankar, Suhas V. (1980). *Numerical Heat Transfer and Fluid Flow.* Hemisphere Publishing Corporation. ISBN 0891165223.

[21] K. A. Surana, S. Allu, P. W. Tenpas, J. N. Reddy. (2007). "K-Version of Finite Element Method in Gas Dynamics: Higher-Order Global Differentiability Numerical Solutions". *International Journal for Numerical Methods in Engineering. 69(6),* pp. 1109–1157.

[22] Huebner, K.H.; Thornton, E.A.; and Byron, T.D. (1995). *The Finite Element Method for Engineers (3rd Ed.).* Wiley Interscience.

[23] Smith GD (1985). *Numerical solution of partial differential equations: finite difference methods (3rd Ed.).* Oxford University Press.

[24] Kao, K.-H., & Liou, M.-S. (1995). Advance in overset grid schemes: From Chimera to DRAGON Grids. *AIAA Journal, 33*(10), pp. 1809-1815.

[25] D. S. Nichols, K. Sreenivas, S. L. Karman, & B. Mitchell. (2007). Turbulence Modeling for Highly Separated Flows. *45th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2007-1407.*

[26] D. S. Nichols, D. G. Hyams, K. Sreenivas, B. Mitchell, L. K. Taylor, & D. L. Whitfield. (2006, January). An Unstructured Incompressible Multi-Phase Solution Algorithm. *44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2006-1290.*

[27] K.Sreenivas, D. Hyams, D. Nichols, B. Mitchell, L. Taylor, W. Briley, & D. Whitfield. (2005, January). Development of an Unstructured Parallel Flow Solver for Arbitrary Mach Numbers. *43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2005-0325.*

[28] P. Moin, & K. Mahesh. (1998). Direct Nmerical Smulation: A Tool in Turbulence Research. *Annual Review of Fluid Mechanics, 30*(1), pp. 539-578.

[29] D. K. Lilly. (1992). A Proposed Modification of the Germano Subgrid Scale Closure. *Physics of Fluids A: Fluid Dynamics, 4*, p. 633.

[30] C. Hirsch, & B. Tartinville. (2009). Reynolds-Averaged Navier-Stokes Modelling for Industrial Applications and Some Challenging Issues. *International Journal of Computational Fluid Dynamics, 23*(4), pp. 295-303.

[31] F. R. Menter. (1994). Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA journal, 32*(8), pp. 1598-1605.

[32] D. C. Wilcox. (1998). *Turbulence Modeling for CFD* (2 ed.). La Canada, California: DCW Industries Inc.

[33] W. Roger Briley, Lafe Taylor, & Dave Whitfield. (2003). High-Resolution Viscous Flow Simulations at Arbitrary Mach Number. *Journal of Computational Physics, 141,* pp. 79-105.

[34] W. Briley, L. Taylor, & D. Whitfield. (2003). High-Resolution Viscous Flow Simulations at Arbitrary Mach Number. *Journal of Computational Physics, 184,* pp. 79–105.

[35] L. K. Taylor. (1991). Unsteady Three-Dimensional Incompressible Algorithm based on Articial Compressibility. *Ph.D thesis, Mississippi State University.*

[36] D. Stephen Nichols, Kidambi Sreenivas, Steve L. Karman, & Brent Mitchell. (2007). Turbulence Modeling for Highly Separated Flows. *45th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2007-1407*.

[37] F. R. Menter, M. Kuntz, & R. Bender. (2003). A Scale-Adaptive Simulation Model for Turbulent Flow Predictions. *41st AIAA Aerospace Sciences Meeting and Exhibit*, p. 0767.

[38] P. Spalart, & et al. (1997). Comments of Feasibility of LES for Wings and on the Hybrid RANS/LES Approach. *Advances in DNS/LES, Proceedings of the First AFOSR International Conference on DNS/LES.*

[39] P. R. Spalart. (2006). A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities. *Theoretical and Computational Fluid Dynamics, 20*(3), pp. 181-195.

[40] R. H. Nichols, & C. C. Nelson. (2001). Weapons Bay Acoustic Predictions Using a Multiscale Turbulence Model. *Aircraft Store Compatibility Symposium and Workshop*

[41] F. R. Menter. (1997). Eddy Viscosity Transport Equations and Their Relation to the k-e Model. *Journal of Fluids Engineering, 119,* pp. 876-884.

[42] D.C. Wilcox. (2007, January). Formulation of the k-w Turbulence Model Revisited. *45th AIAA Aerospace Sciences Meeting and Exhibit*.

[43] F. R. Menter. (2009). Review of the Shear-Stress Transport Turbulence Model Experience from an Industrial Perspective. *International Journal of Computational Fluid Dynamics, 23*(4), pp. 305-316.

[44] M. Strelets. (2001, January). Detached Eddy Simulation of Massively Separated Flows. *39th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 01-0879.*

[45] D. Stephen Nichols. (2011). Accounting for Shocks in the kekw Turbulence Model. *41th AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2011-3573*.

[46] P. L. Roe. (1986). Characteristic-Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics, 18*, pp. 337–365.

[47] A. Harten, P. D. Lax, & B. van Leer. (1983). On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review, 25*, pp. 35–61.

[48] E. F. Toro. (2009). *Riemann Solvers and Numerical Methods for Fluid Dynamics* (3 ed.). Springer-Verlag Berlin Heidelberg. doi:10.1007/b79761

[49] E. F. Toro, M. Spruce, & W. Speares. (1994). Restoration of the Contact Surface in the HLL-Riemann Solver. *Shock Waves 4(1),* pp. 25-34.

[50] E. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics* (*3rd Ed*.). New York: Springer Dordecht Heidelberg, London.

[51] P. Batten, N. Clarke, C. Lambert, & D. Causon. (1997, November). On The Choice of Wavespeeds for the HLLC Riemann Solver. *SIAM J. Sci. Comput, 18(6),* pp. 1553–1570.

[52] A. Harten, B. Engquist, S. Osher, & S. R. Chakravarthy. (1987). Uniformly High Order Accurate Essentially Non-Oscillatory Schemes. *Journal of Computational Physics, 72*(2), pp. 231-303.

[53] X. D. Liu, S. Osher, & T. Chan. (1994). Weighted Essentially Non-oscillatory Schemes. *Journal of Computational Physics, 115(1),* pp. 200-228.

[54] C. W. Shu. (2009). High Order Weighted Essentially Non-oscillatory Schemes for Convection Dominated Problems. *SIAM Review, 51*(1), pp. 82-126.

[55] D. G. Hyams. (2000, May). An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies. *PhD thesis, Mississippi State University*, pp. 23-25.

[56] J. Blazek. (2001). *Computational Fluid Dynamics: Principles and Applications*. Elsevier Science Ltd.

[57] B. van Leer. (1997). Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Godunov's Method. *Journal of Computational Physics, 135*, pp. 29–248.

[58] C. Berthon. (2006). Robustness of MUSCL Schemes for 2d Unstructured Meshes. *Journal of Computational Physics, 218*, pp. 495–509.

[59] M. Berger, & M. J. Aftosmis. (2005). Analysis of Slope Limiters on Irregular Grids. *43rd AIAA Aerospace Sciences Meeting and Exhibit.*

[60] R. Courant, K. Friedrichs, & H. Lewy. (1967). On the Partial Differential Equations of Mathematical Physics. *IBM Journal of Research and Development, 11*(2), pp. 215-224.

[61] M. B. Alkislar. (2001). Flowfield Measurement in a Screeching Rectangular Jet. *Phd dissertation, Florida State University.*

[62] J. T. Bolton, B. Thurow, N. Arora, & F. S. Alvi. (2016). Volumetric Measurement of a Shock Wave-Turbulent Boundary Layer Interaction Using Plenoptic Particle Image Velocimetry. *32nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, pp. 1-13.

[63] J. Klemkowsky, & B. Thurow. (2016, January). 3D Visualization of Density Gradients Using a Plenoptic Camera and Background. *54th AIAA Aerospace Sciences Meeting, AIAA Paper 2016-1047.*

[64] C. Jones, C. Clifford, J. T. Bolton, B. Thurow, L. Mears, & F. S. Alvi. (2017). Preliminary Plenoptic PIV Results for Volumetric Measurements of Shock Wave-Boundary Layer Interactions. *33rd AIAA Aerodynamic Measurement Technology and Ground Testing Conference, AIAA Paper 2017-4065.*

[65] T. W. Fahringer, & B. S. Thurow. (2015, January). Comparing Volumetric Reconstruction Algorithms for Plenoptic-PIV. *53rd AIAA Aerospace Sciences Meeting*, pp. 1-10

[66] A. Pope, & K. Goin. (1965). *High-Speed Wind Tunnel Testing.* New York, NY, USA: John Wiley and Sons.

[67] E. Reshotko, W. S. Saric, & H. M. Nagib. (1997). Flow Quality Issues for Large Wind Tunnels.

[68] J.E. John, & T.G. Keith. (2006). *Gas Dynamics.* (*3rd Ed*.) Upper Saddle River, NJ: Pearson Prentice Hall.

[69] Frank M. White. (2017). *Fluid Mechanics (4th Ed.).* New York: McGraw-Hill.

[70] Pagendarm HG, & Seitz B. (1993). An Algorithm for Detection and Visualization of Discontinuities in Scientific Data Fields Applied to Flow Data with Shock Waves. In P. Palamidese, *Scientific Visualization: Advanced Software Techniques*.

[71] R.M. Beam and R.F. Warming. (1978) An implicit factored scheme for the compressible Navier-Stokes equations. *AIAA Journal, 16*(4), pp.393-402

[72] Lafayette K. Taylor, J.A. Busby, M.Y. Jiang, A. Arabshahi, K. Sreenivas, and D.L. Whitfield. (1992) Time accurate incompressible Navier-Stokes simulation of the flapping foil experiment. *Sixth International Conference on Numerical Ship Hydrodynamics*.

[73] Balsara, D.S., & Shu, C.W. (2000) Monotonicity Preserving Weighted Essentially Non-Oscillatory Schemes with Increasingly High Order of Accuracy. *Journal of Computational Physics, 160,* pp. 405-452.

[74] D. Nichols, D. Hyams, K. Sreenivas, B. Mitchell, L. Taylor, & D. Whitfield. (2006). An Unstructured Incompressible Multi-Phase Solution Algorithm. *44th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings 2006-1290*

Appendix A

The MATLAB Program for Converging-Diverging Nozzle Design

Flow Characteristic:

```
G = 1.4;

Me = 2.1;

n = 25;

grids = 1;
```

Initialize datapoint matrices:

```
Km = zeros(n,n);    % K- vlaues (Constant along right running characteristic lines)

Kp = zeros(n,n);    % K- vlaues (Constant along left running characteristic lines)

Theta = zeros(n,n); % Flow angles relative to the horizontal

Mu = zeros(n,n);    % Mach angles

M = zeros(n,n);     % Mach Numbers

x = zeros(n,n);     % x-coordinates

y = zeros(n,n);     % y-coordinates
```

Find NuMax (maximum expansion angle):

```
[~, ThetaMax, ~] = PMF(G,Me,0,0);

NuMax = ThetaMax/2;
```

Define some flow parameters of originating characteristic lines:

```
dT = NuMax/n;

ThetaArc(:,1) = (0:dT:NuMax);

NuArc = ThetaArc;

KmArc = ThetaArc + NuArc;

[~, ~, MuArc(:,1)] = PMF(G,0,NuArc(:,1),0);
```

Coordinates of wall along curve from throat

```
y0 = 1; % Define throat half-height

ThroatCurveRadius = 3*y0; % Radius of curvature just downstream of the throat

[xarc, yarc] = Arc(ThroatCurveRadius,ThetaArc); % Finds x- and y-coordinates for given theta-values

%xarc = ThroatCurveRadius.*tand(ThetaArc)./sqrt(1 + tand(ThetaArc).^2);

%yarc = ThroatCurveRadius - sqrt(ThroatCurveRadius.^2 - x.^2);

yarc(:,1) = yarc(:,1) + y0; % Defines offset due to arc being above horizontal
```

Fill in missing datapoint info along first C+ line:

```
Km(:,1) = KmArc(2:length(KmArc),1);

Theta(:,1) = ThetaArc(2:length(KmArc),1);

Nu(:,1) = Theta(:,1);

Kp(:,1) = Theta(:,1)-Nu(:,1);

M(1,1) = 1;

Nu(1,1) = 0;

Mu(1,1) = 90;

y(1,1) = 0;

x(1,1) = xarc(2,1) + (y(1,1) - yarc(2,1))/tand((ThetaArc(2,1) - MuArc(2,1) - MuArc(2,1))/2);

% Finds the information at interior nodes along first C+ line

for i=2:n
```

```
        [M(i,1), Nu(i,1), Mu(i,1)] = PMF(G,0,Nu(i,1),0);
        s1 = tand((ThetaArc(i+1,1) - MuArc(i+1,1) + Theta(i,1) - Mu(i,1))/2);
        s2 = tand((Theta(i-1,1) + Mu(i-1,1) + Theta(i,1) + Mu(i,1))/2);
        x(i,1) = ((y(i-1,1)-x(i-1,1)*s2)-(yarc(i+1,1)-xarc(i+1,1)*s1))/(s1-s2);
        y(i,1) = y(i-1,1) + (x(i,1)-x(i-1,1))*s2;
end
```

Find flow properties at remaining interior nodes:

```
for j=2:n;
    for i=1:n+1-j;
        Km(i,j) = Km(i+1,j-1);
        if i==1;
            Theta(i,j) = 0;
            Kp(i,j) = -Km(i,j);
            Nu(i,j) = Km(i,j);
            [M(i,j), Nu(i,j), Mu(i,j)] = PMF(G,0,Nu(i,j),0);
            s1 = tand((Theta(i+1,j-1)-Mu(i+1,j-1)+Theta(i,j)-Mu(i,j))/2);
            x(i,j) = x(i+1,j-1) - y(i+1,j-1)/s1;
            y(i,j) = 0;
        else
            Kp(i,j) = Kp(i-1,j);
            Theta(i,j) = (Km(i,j)+Kp(i,j))/2;
            Nu(i,j) = (Km(i,j)-Kp(i,j))/2;
            [M(i,j), Nu(i,j), Mu(i,j)] = PMF(G,0,Nu(i,j),0);
            s1 = tand((Theta(i+1,j-1)-Mu(i+1,j-1)+Theta(i,j)-Mu(i,j))/2);
            s2 = tand((Theta(i-1,j)+Mu(i-1,j)+Theta(i,j)+Mu(i,j))/2);
```

```
            x(i,j) = ((y(i-1,j)-x(i-1,j)*s2)-(y(i+1,j-1)-x(i+1,j-1)*s1))/(s1-s2);
            y(i,j) = y(i-1,j) + (x(i,j)-x(i-1,j))*s2;
        end
    end
end
```

Find wall node information:

```
xwall = zeros(2*n,1);
ywall = xwall;
ThetaWall = ywall;
xwall(1:n,1) = xarc(2:length(xarc),1);
ywall(1:n,1) = yarc(2:length(xarc),1);
ThetaWall(1:n,1) = ThetaArc(2:length(xarc),1);
for i=1:n-1
    ThetaWall(n+i,1) = ThetaWall(n-i,1);
end
for i=1:n

    s1 = tand((ThetaWall(n+i-1,1) + ThetaWall(n+i,1))/2);
    s2 = tand(Theta(n+1-i,i)+Mu(n+1-i,i));
    xwall(n+i,1) = ((y(n+1-i,i)-x(n+1-i,i)*s2)-(ywall(n+i-1,1)-xwall(n+i-1,1)*s1))/(s1-s2);
    ywall(n+i,1) = ywall(n+i-1,1) + (xwall(n+i,1)-xwall(n+i-1,1))*s1;

end
assignin('caller','Coords',[xwall ywall])
```

Plot MOC Result:

```
figure (1)
if grids == 1
    plot(xwall,ywall,'r-','LineWidth',3)
    axis equal
    axis([0 ceil(xwall(length(xwall),1)) 0 ceil(ywall(length(ywall),1))])
    hold on
    plot(xarc,yarc,'r-','LineWidth',3)
    hold on
    plot(xarc,yarc)
    for i=1:n-1
        plot(x(1:n+1-i,i),y(1:n+1-i,i),'b-')
    end
    for i=1:n
        plot([xarc(i,1) x(i,1)],[yarc(i,1) y(i,1)],'b-')
        plot([x(n+1-i,i) xwall(i+n,1)],[y(n+1-i,i) ywall(i+n,1)],'b-')
    end
    for c=1:n
        for r=2:n+1-c
            plot([x(c,r) x(c+1,r-1)],[y(c,r) y(c+1,r-1)],'b-')
        end
    end
    hx1 = xlabel('Length [x/y0]');
    hy1 = ylabel('Height [y/y0]');
    set(hx1,'Fontsize',15);
    set(hy1,'Fontsize',15)
    ht1 = title(['Supersonic Portion Contour Using MOC ']);
```

```
    set(ht1,'Fontsize',15);
    set(gca,'FontSize',15);
    grid on
    print(gcf,'-dpng',strcat('C:\Users\Jiayue\Desktop\Supersonic.png'));
end
```

Plot the Contour of CD Nozzle:

```
Diverging = ywall;
X = zeros(1150,1);
Y = zeros(1150,1);
x1 = [0:0.008:8];
x2 = [0:0.02:2];
len = xwall(50)
x3 = xwall./len.*6;
height_exit = Diverging(50);
Diverging = Diverging./height_exit.*1.5;
%Diverging = Diverging./height_exit.*2.0;
height_throat = Diverging(1);
y1 = 5-5.*(10.*(x1./8).^3-15.*(x1./8).^4+6.*(x1./8).^5);
y1 = y1+height_throat;
for i = 1:1:1150
    if i <= 1000
        X(i) = x1(i)+2;
        Y(i) = y1(i);
    else if i>1000 && i<=1100
            X(i) = x2(i-1000)+10;
```

```
            Y(i) = height_throat;


        else
            X(i) = x3(i-1100)+12;
            Y(i) = Diverging(i-1100);
        end
    end
end
figure(2)
plot(X,Y,'-','LineWidth',2);
hold on
plot(X,-Y,'-','LineWidth',2);
hold on
plot([X(1150),X(1150)],[-Y(1150),Y(1150)],'-','LineWidth',2);
hold on
% Plot Reservoir
plot([0,2],[Y(1),Y(1)],'-','LineWidth',2)
hold on
plot([0,2],[-Y(1),-Y(1)],'-','LineWidth',2)
hold on
plot([0,0],[-Y(1),Y(1)],'-','LineWidth',2)
%
hx2 = xlabel('Length');
hy2 = ylabel('Height');
set(hx2,'Fontsize',15);
set(hy2,'Fontsize',15)
ht2 = title(['The Contour of Converging-Diverging Nozzle']);
```

```
set(ht2,'Fontsize',15);

set(gca,'FontSize',15);

set(gca,'xtick',[0:2:20]);

set(gca,'ytick',[-8:2:8]);

grid on

print(gcf,'-dpng',strcat('C:\Users\Jiayue\Desktop\CDnozzle.png'));

assignin('caller','Contour',[X Y])
```

Appendix B

The Python Script for the Data Extraction

```
#!/usr/bin/env python
################################################################################
# supersonic wind tunnel data extraction
################################################################################
#import math,numpy,sys
#import simcenter
import numpy

def UX_get_nsensor(uxp,sdb,grid):
    print "PYTHON: changing nsensor."
    grid['nsensor'][0] = 3

    return 0

def UX_get_sensor_points(uxp,sdb,grid):
    print "PYTHON: getting sensor points."
    nsensor = grid['nsensor']
    sxyz = grid['xyz']

    if nsensor == 0:
        return 0

    sxyz[1,:] = [x, y, z]
    sxyz[2,:] = [x, y, z]
    sxyz[3,:] = [x, y, z]

    # scale by reference length
    scale = uxp['REFERENCE-LENGTH'][0]
    print "PYTHON: scaling by reference length " + str(scale)
#     sxyz = sxyz / scale
#     grid['xyz'] = sxyz
    sxyz /= scale
```

```
    return 0

def UX_get_sensor_box(uxp,sdb,grid):
    print "PYTHON: getting sensor box."
    nsensor = grid['nsensor']
    sbox = grid['box']

    if nsensor == 0:
        return 0

    for i in xrange(1,nsensor+1):
        sbox[i,0] = 5.0
        sbox[i,1] = 5.0
        sbox[i,2] = 5.0

    return 0

# unit testing.    Here, we can run the module directly and double check the numbers.
if __name__ == "__main__":
   ndim = 3
   nrelease = 1
   nsensor = 2
   grid = {}
   uxp = {}
   sdb = {}
   grid["nrelease"] = 0
   grid["ndim"] = ndim
   grid["xyz"] = numpy.zeros((nsensor+1,ndim))
   grid["box"] = numpy.zeros((nsensor+1,ndim))
   grid["nsensor"] = numpy.zeros(1)
   uxp['REFERENCE-LENGTH'] = numpy.zeros(1)
   uxp['REFERENCE-LENGTH'][0] = 2.0
   UX_get_nsensor(uxp,sdb,grid)
   UX_get_sensor_points(uxp,sdb,grid)
   print grid['xyz']
```

Appendix C

The MATLAB Program for Data Analysis

Interpolation processing and generating images of flow characteristics:

```
load('wilcox.mat');    %data input
%load('sst.mat');
%load('sst.mat');
S = size(Data);
    p1 = 1;
    num = 1;
for i = 1:1:S-1
    if Data(i,3) ~= Data(i+1,3)
        z_location = Data(i,3)*1000;    %in mm
        p2 = i;
        A = Data(p1:p2,:);
        A(:,1:3) =  A(:,1:3).*1000; % in mm
        x_min = min(A(:,1));
        x_max = max(A(:,1));
        y_min = min(A(:,2));
        y_max = max(A(:,2));
        figure (1)  %  X-Velocity Distribution

[X,Y,V]=griddata(A(:,1),A(:,2),A(:,4),linspace(x_min,x_max,50)',linspace(y_min,y_max,50),'cubic')
```

```
;
        contourf(X,Y,V,80);
        shading flat
        hx = xlabel('x');
        hy = ylabel('y');
        set(hx,'Fontsize',25);
        set(hy,'Fontsize',25);
        colormap(jet(128))
        colorbar
        caxis([200,500])
        ht = title(['X-Velocity Distribution at z=' num2str(z_location),'mm (m/s)']);
        set(ht,'Fontsize',25);
        set(gca,'FontSize',20);
        print(gcf,'-
dpng',strcat('C:\Users\Jiayue\Desktop\Pic_from_Matlab\Vx','_at_z=',num2str(z_location),'.png'));
        figure (2)  %  Density Distribution

[X,Y,rho]=griddata(A(:,1),A(:,2),A(:,7),linspace(x_min,x_max,50)',linspace(y_min,y_max,50),'cubic
');
        contourf(X,Y,rho,80);
        shading flat
        hx = xlabel('x');
        hy = ylabel('y');
        set(hx,'Fontsize',25);
        set(hy,'Fontsize',25);
        caxis([0.7,1.7])
        colormap(jet(128))
```

```
        colorbar
        ht = title(['Density Distribution at z=' num2str(z_location),'mm (kg/m^3)']);
        set(ht,'Fontsize',25);
        set(gca,'FontSize',20);
        print(gcf,'-
dpng',strcat('C:\Users\Jiayue\Desktop\Pic_from_Matlab\rho','_at_z=',num2str(z_location),'.png'));
        figure (3)  %  Pressure Distribution

[X,Y,P]=griddata(A(:,1),A(:,2),A(:,8),linspace(x_min,x_max,50)',linspace(y_min,y_max,50),'cubic')
;
        contourf(X,Y,P,80);
        shading flat
        hx = xlabel('x');
        hy = ylabel('y');
        set(hx,'Fontsize',25);
        set(hy,'Fontsize',25);
        caxis([43657,107536])
        colormap(jet(128))
        colorbar
        ht = title(['Pressure Distribution at z=' num2str(z_location),'mm (Pa)']);
        set(ht,'Fontsize',25);
        set(gca,'FontSize',20);
        print(gcf,'-
dpng',strcat('C:\Users\Jiayue\Desktop\Pic_from_Matlab\Pressure','_at_z=',num2str(z_location),'.pn
g'));
        figure (4)  %  Temperature Distribution
```

```
[X,Y,T]=griddata(A(:,1),A(:,2),A(:,9),linspace(x_min,x_max,50)',linspace(y_min,y_max,50),'cubic')
;
        contourf(X,Y,T,80);
        shading flat
        hx = xlabel('x');
        hy = ylabel('y');
        set(hx,'Fontsize',25);
        set(hy,'Fontsize',25);
        caxis([162,280])
        colormap(jet(128))
        colorbar
        ht = title(['Temperature Distribution at z=' num2str(z_location),'mm (K)']);
        set(ht,'Fontsize',25);
        set(gca,'FontSize',20);
        print(gcf,'-
dpng',strcat('C:\Users\Jiayue\Desktop\Pic_from_Matlab\Temperature','_at_z=',num2str(z_location),'
.png'));
        num = num +1
        p1 = p2 +1;
    end
end
```

Generation of line cuts for simulation data:

```
clear all;
clc
%load('wilcox.mat');    % data input
```

```
%load('SST.mat');
load('far.mat');
%load('PIVdata.mat');
z_location = 7.25;     % determine z location
%z_location = 11.25;
%z_location = 15.25;
S = size(Data);
p1 = 1;
num = 1;
Data(:,1:3) =  Data(:,1:3).*1000;      % in mm
X = linspace(590,635,100);
Y = zeros(4,100);
V = zeros(4,100);
yc = [-17.55; -23.55; -27.55; -33.55];     % y location of four lines
for i = 1:1:S(1)-1
    if Data(i,3) ~= Data(i+1,3) && abs(Data(i+1,3)-z_location)<0.001
        p1 = i+1;
    end
    if Data(i,3) ~= Data(i+1,3) && abs(Data(i,3)-z_location)<0.001
        p2 = i;
        A = Data(p1:p2,:);
        for j = 1:1:4
            y_temp = yc(j);
            for k = 1:1:100
                clear n
                clear m
                clear c1
```

```
                clear c2
                c1 = (A(:,1)-X(k)).^2+(A(:,2)-y_temp).^2;
                n = find(c1==min(c1));
                Sn = size(n,1);
                for h = 1:1:Sn
                    c2(h,1) = A(n(h),1)-X(k);
                end
                m = find(c2==min(c2));
                Y(j,k) = A(n(m(1)),2);
                V(j,k) = A(n(m(1)),4);
            end
        end
    end
end
X1 = 590:0.1:635;
for i = 1:1:4
    V1(i,:) = interp1(X,V(i,:),X1,'Spline');
end
```

Generation of line cuts for physical experiment data:

```
clear
clc
load('PIVdata.mat');
z_location = 7.25;     % determine z location
%z_location = 11.25;
%z_location = 15.25;
```

```
X_velocity=double(uAvg);
for i=1:1:73
    for j=1:1:46
        for k=1:1:73
            if mask(i,j,k)==0
                X_velocity(i,j,k) = 0;
            end
        end
    end
end
x = x(14:60,5:36,:);
y = y(14:60,5:36,:);
z = z(14:60,5:36,:);
X_velocity = X_velocity(14:60,5:36,:);
num = 1;
z_col = 0;
y_col = 0;
X = x(:,1,1);
yc = [-17.55; -23.55; -27.55; -33.55];     % y location of four lines
for i = 1:1:73
    if abs(z(1,1,i)-z_location)<0.1
        z_col = i;
    end
end
for i = 1:1:4
    for j = 1:1:32
        if abs(y(1,j,1)-yc(i))<0.001
```

```
            y_col = j;
        end
    end
    V(:,i) = X_velocity(:,y_col,z_col);
end
X2 = 599:0.1:629;
for i = 1:1:4
    V2(i,:) = interp1(X,V(:,i),X2,'Spline');
end
```

Image plot:

```
z_location = [7.25,11.25,15.25];
yc = [-17.55; -23.55; -27.55; -33.55];
n = 0;
for i = 1:1:3
    n = n+1;
    for j = 1:1:4
        figure (n)
        load(strcat('wilcox_z_location_',num2str(z_location(i)),'.mat'));
        plot(X1,V1(j,:),'b-','LineWidth',3);
        hold on
        load(strcat('SST_z_location_',num2str(z_location(i)),'.mat'));
        plot(X1,V1(j,:),'y-','LineWidth',3);
        hold on
        load(strcat('far_z_location_',num2str(z_location(i)),'.mat'));
        plot(X1,V1(j,:),'g-','LineWidth',3);
```

```
        hold on
        load(strcat('PIV_z_location_',num2str(z_location(i)),'.mat'));
        plot(X2,V2(j,:),'r-','LineWidth',3);
        hx = xlabel('x (mm)');
        hy = ylabel('x-velocity (m/s)');
        set(hx,'Fontsize',15);
        set(hy,'Fontsize',15);
        ht = title(['X-Velocity at y=',num2str(yc(j)),'mm & z=',num2str(z_location(i)),'mm']);
        set(ht,'Fontsize',15);
        set(gca,'FontSize',15);
        h = legend('In-Tunnel Wilcox k-/omega Model','In-Tunnel Menter SST Model','Farfield Wilcox
k-/omega Model','Physical Experiment');
        set(h,'Fontsize',15);
        grid on
        print(gcf,'-dpng',strcat('C:\Users\Jiayue\Desktop\pic_comparison\X-Velocity             at
y=',num2str(yc(j)),'mm & z=',num2str(z_location(i)),'.png'));
        n = n+1;
    end
end
```

Vita

Jiayue Hu was born in Beijing, China, on August 27, 1993. After graduating from the Affiliated High School of Peking University, he studied in the Beijing Institute of Technology, where he completed his Bachelor's degree in Engineering Mechanics in 2015. Following this, he started graduate school in the Computational Fluid Dynamics Laboratory at Auburn University, where he is finishing school with a M.S. degree in Aerospace Engineering. Jiayue plans to continue his research in Aerospace Engineering and pursue a Ph.D.