

UCLA

UCLA Electronic Theses and Dissertations

Title

Numerical Simulations of High-Speed Flows Over Complex Geometries

Permalink

<https://escholarship.org/uc/item/5xs4p59w>

Author

Greene, Patrick Timothy

Publication Date

2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Numerical Simulations of High-Speed Flows Over Complex Geometries

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Aerospace Engineering

by

Patrick Timothy Greene

2014

© Copyright by
Patrick Timothy Greene
2014

ABSTRACT OF THE DISSERTATION

Numerical Simulations of High-Speed Flows Over Complex Geometries

by

Patrick Timothy Greene

Doctor of Philosophy in Aerospace Engineering

University of California, Los Angeles, 2014

Professor John Kim, Co-chair

Professor Jeff D. Eldredge, Co-chair

The effects of surface roughness on the stability of hypersonic flow are of great importance to hypersonic vehicles. Surface roughness can greatly alter boundary-layer flow and cause transition to turbulence to occur much earlier compared to a smooth wall, which will result in a significant increase of wall heating and skin friction drag. The work presented in this dissertation was motivated by a desire to study the effects of isolated roughness elements on the stability of hypersonic boundary layers. A new code was developed which can perform high-order direct numerical simulations of high-speed flows over arbitrary geometries. A fifth-order hybrid weighted essentially non-oscillatory scheme was implemented to capture any steep gradients in the flow created by the geometries. The simulations are carried out on Cartesian grids with the geometries imposed by a third-order cut-cell method. A multi-zone refinement method is also implemented to provide extra resolution at locations with expected complex physics. The combination results in a globally fourth-order scheme.

Results for two-dimensional and three-dimensional test cases show good agreement with previous results and will be presented. Results confirming the code's high order of convergence will also be shown. Two-dimensional simulations of flow over complex geometries will be presented to demonstrate the code's capabilities. Results for Mach 6 flow over a three-dimensional cylindrical roughness element will also be presented. The results will show that

the code is a promising tool for the study of hypersonic roughness-induced transition.

The dissertation of Patrick Timothy Greene is approved.

Joseph Teran

Xiaolin Zhong

Jeff D. Eldredge, Committee Co-chair

John Kim, Committee Co-chair

University of California, Los Angeles

2014

*To my parents . . .
for all their love and support.*

TABLE OF CONTENTS

1	Introduction	1
1.1	Hypersonic Boundary-Layer Transition	1
1.2	Effects of Roughness on Hypersonic Transition	3
1.3	Cartesian Grid Methods	6
1.3.1	Immersed Boundary Method	7
1.3.2	Sharp-Interface Method	8
1.4	Adaptive Mesh Refinement	9
1.5	Goals of the Present Study	10
2	Governing Equations	13
2.1	Cartesian Coordinate System	13
2.2	Curvilinear Coordinate System	14
3	Numerical Method	16
3.1	Navier-Stokes Solver	16
3.1.1	Inviscid Terms	17
3.1.2	Viscous Terms	25
3.1.3	Time Advancement	25
3.2	Cut-Cell Method	26
3.2.1	Point Classification	26
3.2.2	Dropped Points	27
3.2.3	Regular Points	29
3.2.4	Irregular Points	29

3.2.5	Boundary Points	31
3.3	Multi-Zone Refinement Method	34
3.3.1	Advance Coarse Grid	37
3.3.2	Change in State Variable with Time on Coarse Grid	37
3.3.3	Refinement Grid Ghost Point Values	37
3.3.4	Refinement Grid Sub-Step Ghost Point Values	39
3.3.5	Advance Refinement Grid	40
3.3.6	Update Coarse Grid Ghost Points	41
3.3.7	Non-Conservation of Multi-Zone Refinement Method	42
4	Order of Convergence	46
4.1	Cut-Cell Method	46
4.2	Multi-Zone Refinement Method	47
5	Cut-Cell and Immersed Boundary Method Comparison	54
5.1	Immersed Boundary Method	54
5.2	Simulation Setup	56
5.3	Results	58
6	Code Validation	62
6.1	Subsonic Flow Over a Cylinder	62
6.1.1	Simulation Setup	62
6.1.2	Results	63
6.2	Hyperbolic Tangent Roughness	65
6.2.1	Simulation Setup	66
6.2.2	Results	68

7	Complex Geometries	72
7.1	Supersonic Flow Over the Ubuntu Logo	72
7.1.1	Simulation Setup	73
7.1.2	Results	75
7.2	Subsonic Flow Over the UCLA Logo	78
7.2.1	Simulation Setup	80
7.2.2	Results	82
8	Boeing/AFOSR Mach 6 Quiet Tunnel	86
8.1	Simulation Setup	86
8.2	Results	90
8.3	Geometry Change	91
9	Isolated Roughness Element	93
9.1	Gaussian Roughness	94
9.1.1	Simulation Setup	95
9.1.2	Results	99
9.2	Cylindrical Roughness	104
9.2.1	Simulation Setup	104
9.2.2	Results	105
10	Conclusion	109
10.1	Future Work	111
A	Appendix	113
A.1	Two-Dimensional Characteristic Transformation	113

A.2 Roe Average	114
References	115

LIST OF FIGURES

1.1	Paths to Transition in Boundary Layers	2
1.2	Streamlines for Mach 6 Flow Over a Gaussian Roughness	3
1.3	Length Scales for Mach 6 Flow Over a Cylindrical Roughness Element	10
3.1	Fifth-Order WENO Stencils	21
3.2	Cut-Cell Point Classification for ξ Grid Line	27
3.3	Cut-Cell Point Classification for Two-Dimensional Computational Space	28
3.4	Viscous Derivative Stencil for Cut-Cell Method	33
3.5	Sample Multi-Zone Refinement Grid	34
3.6	Multi-Zone Refinement Grid Generation and Point Classification	36
3.7	Refinement Grid Ghost Point Interpolation Stencil	38
3.8	Two-Dimensional Refinement Grid Ghost Point Interpolation Stencil	39
3.9	Sample Uniformly Space Grid	42
3.10	Sample Uniformly Space Grid with Refinement Zone	44
4.1	Sample Grid for Circular Couette Flow	47
4.2	Mach Contour for Circular Couette Flow	48
4.3	Flow Variable Profiles for Circular Couette Flow	49
4.4	Circular Couette Flow Error	50
4.5	Propagating Density Wave Contour	51
4.6	Propagating Density Wave Spatial Error	52
4.7	Propagating Density Wave Temporal Error	53
5.1	Diagram for Flow Over 2-D Hyperbolic Tangent Roughness	56

5.2	Body-Fitted and Cut-Cell Grids for 2-D Hyperbolic Tangent Roughness . . .	57
5.3	Contours for Flow Over 2-D Hyperbolic Tangent Roughness	59
5.4	Velocity and Pressure Profiles for 2-D Hyperbolic Tangent Roughness	60
6.1	Grid for Subsonic Flow Over Cylinder	63
6.2	Results for Subsonic Flow Over a Cylinder	64
6.3	Strouhal Comparison for Subsonic Flow Over Cylinder	65
6.4	Grid in the Vicinity of 3-D Hyperbolic Tangent Roughness	67
6.5	Body-Fitted Grid in the Vicinity of 3-D Hyperbolic Tangent Roughness . . .	68
6.6	Contours at $z/\delta = 0$ Plane for 3-D Hyperbolic Tangent Roughness	69
6.7	Contours at $x/\delta = 0$ Plane for 3-D Hyperbolic Tangent Roughness	70
7.1	Logo for the Ubuntu Linux Distribution	72
7.2	Grid for Flow Over the Ubuntu Logo	74
7.3	Contours for Flow Over the Ubuntu Logo	76
7.4	Zoomed in Contours for Flow Over the Ubuntu Logo	77
7.5	Streamlines for Flow Over the Ubuntu Logo	78
7.6	Frequency Analysis for Flow Over the Ubuntu Logo	79
7.7	Logo for the University of California, Los Angeles	80
7.8	Grid for Flow Over the UCLA Logo	81
7.9	Contours for Flow Over the UCLA Logo	83
7.10	Zoomed in Contours for Flow Over the UCLA Logo	84
7.11	Streamlines for Flow Over the UCLA Logo	85
8.1	Diagram of Boeing/AFOSR Mach 6 Quiet Tunnel	87
8.2	Portion of Boeing/AFOSR Mach 6 Quiet Tunnel Being Simulated	87

8.3	Grid for Boeing/AFOSR Mach 6 Quiet Tunnel Simulation	88
8.4	Wall Temperature Profile for Boeing/AFOSR Mach 6 Quiet Tunnel	89
8.5	Pitot Pressure Profile from the Boeing/AFOSR Mach 6 Quiet Tunnel	91
9.1	Roughness Element Comparison	95
9.2	Grid in the Vicinity of $k/\delta = 0.75$ Gaussian Roughness	97
9.3	Grid in the Vicinity of $k/\delta = 1.10$ Gaussian Roughness	98
9.4	Density Contours for Gaussian Roughness	100
9.5	Vorticity Magnitude Isosurface for Gaussian Roughness	102
9.6	Vorticity Magnitude Contours for Gaussian Roughness	103
9.7	Grids for Cylindrical Roughness Simulation	106
9.8	Isosurface of Q-Criterion for Cylindrical Roughness	107
9.9	Symmetry Plane Streamline Comparison for Cylindrical Roughness	108
9.10	Surface Streamline and Pressure Comparison for Cylindrical Roughness	108

LIST OF TABLES

3.1	Low-Dissipation Scheme Constants	20
3.2	Viscous Derivative Constants	25
3.3	Inviscid One-Sided Non-Uniform Interpolation Constants	30
3.4	Viscous One-Sided Non-Uniform Finite-Difference Constants	31
3.5	Refinement Grid Ghost Point Interpolation Constants	38
5.1	Flow Conditions for 2-D Hyperbolic Tangent Simulation	57
5.2	Constants for Immersed Boundary Method	57
7.1	Grid Constants for Flow Over the Ubuntu Logo	73
7.2	Freestream Conditions for Flow Over the Ubuntu Logo	75
7.3	Grid Constants for Flow Over the UCLA Logo	80
7.4	Freestream Conditions for Flow Over the UCLA Logo	80
9.1	Flow Conditions for Isolated Roughness Simulations	94
9.2	Roughness Reynolds Number	95
9.3	Grid Constants for Cylindrical Roughness	105

ACKNOWLEDGMENTS

I would like to express my appreciation to my co-advisers Professor John Kim, Professor Jeff D. Eldredge, and Professor Xiaolin Zhong for their guidance and support throughout my work. I would also like to thank Professor Joseph Teran for his time and input as my last committee member. In addition, I'd like to acknowledge my past and present colleagues who have been there to share both the joys and the frustrations of computational fluid dynamics: Dr. Bernie Rajamani, Clifton Mortensen, Dr. Lord Cole, Dr. Maher Lagha, Morad Alvarez, Neal Parsons, and Pilbum Kim.

I would also like to acknowledge the support from the NASA Fundamental Aeronautics Program, under cooperative agreement NNX07AC39A, monitored by Dr. Meelan Choudhari and the partial support of the AFOSR/NASA National Center for Hypersonic Research in Laminar-Turbulent Transition headed by Professor William Saric at Texas A&M University. I'd also like to thank the NASA Ames Research Center Advanced Supercomputing Division and the Extreme Science and Engineering Discovery Environment (XSEDE) for providing the computational time needed to carry out my work.

VITA

1983	Born, Glens Falls, New York, USA.
2006	B.S. in Aerospace Engineering with Mathematics minor, University of California, Los Angeles.
2006–present	Graduate Student Researcher, Department of Mechanical and Aerospace Engineering, University of California, Los Angeles.
2009–2013	Teaching Assistant, Department of Mechanical and Aerospace Engineering, University of California, Los Angeles.
2009	“A Numerical Study of Purdue’s Mach 6 Tunnel with a Roughness Element” by P.T. Greene, J.D. Eldredge, X. Zhong and J. Kim. Presented at <i>47th AIAA Aerospace Sciences Meeting</i> , AIAA Paper 2009-0174
2010	“Development and Breakdown of Görtler Vortices in High Speed Boundary Layers” by F. Li, M. Choudhari, C.-L. Chang, M. Wu and P.T. Greene. Presented at <i>48th AIAA Aerospace Sciences Meeting</i> , AIAA Paper 2010-0705
2011	“Numerical Study of Hypersonic Flow over an Isolated Roughness with a High-Order Cut-Cell Method” by P.T. Greene, J.D. Eldredge, X. Zhong and J. Kim. Presented at <i>41st AIAA Fluid Dynamics Conference</i> , AIAA Paper 2011-3249
2014	“Numerical Simulation of High-Speed Flows Over Complex Geometries with a High-Order Multi-Zone Cut-Cell Method” by P.T. Greene, J.D. Eldredge, X. Zhong and J. Kim,. Presented at <i>52nd AIAA Aerospace Sciences Meeting</i> , AIAA Paper 2014-0426

CHAPTER 1

Introduction

The ability to accurately predict the location of laminar to turbulent transition in boundary layers is of great importance to the design of hypersonic vehicles. At high Mach numbers, the viscous effects in the boundary layer can cause extreme surface temperatures as kinetic energy is converted to thermal energy [1]. When the flow transitions from laminar to turbulent, this heating is further increased by turbulent mixing near the wall. In addition to the thermal effects, turbulent flow also generates larger shear stresses which increase the friction drag on the surface [2]. Without the knowledge of the transition location, designers of hypersonic vehicles must be overly cautious with their designs. Thermal protection and propulsion systems must assume turbulent flow over the entire body [3]. Being able to accurately predict the location of transition has been a goal of fluid dynamics for many years. In spite of its importance, the mechanisms leading to the transition of hypersonic boundary layers are still poorly understood [4]. Predicting the location of transition is further complicated by the presence of surface roughness. The goal of this work is to develop a set of numerical tools for studying the effects of roughness elements on the transition of hypersonic boundary-layer flow.

1.1 Hypersonic Boundary-Layer Transition

The different possible paths of transition in boundary layers are shown in Figure 1.1. The figure is adapted from Reshotko [5]. All the paths to transition begin with disturbances to the boundary layer. These disturbances can originate in the freestream or on the body. In the figure, the amplitude of the disturbances increases from left to right. The disturbances

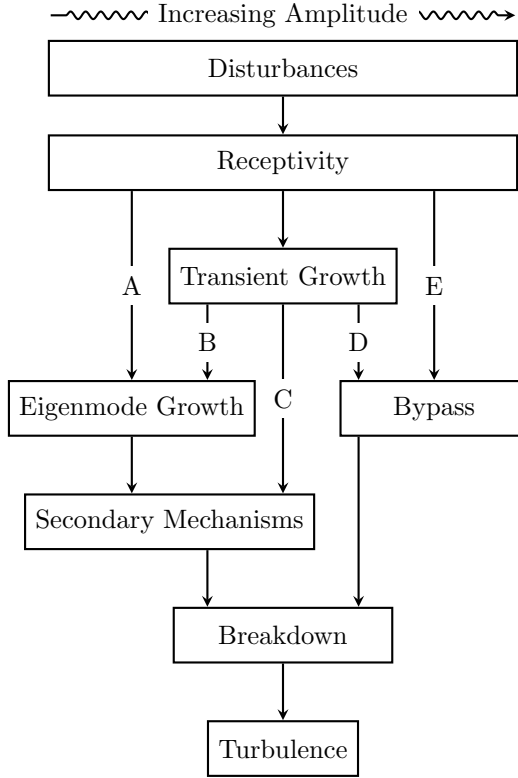


Figure 1.1: Paths to transition in boundary layers. Adapted from Reshotko [5].

enter the boundary layer through a process called receptivity. When infinitesimal amplitude disturbances enter the boundary layer, path A is followed. Transition along this path is usually due to Tollmien-Schlichting, crossflow, or Görtler mechanisms. The initial growth of the amplitude of these disturbances can be obtained as the eigensolutions of the linearized unsteady Navier-Stokes equations. As the amplitude grows, three-dimensional and nonlinear interactions occur in the form of secondary instabilities. The disturbance's growth is very rapid in this stage and breakdown to turbulence occurs quickly. The flight conditions of most supersonic and hypersonic vehicles have infinitesimal initial disturbances which has made path A an important research topic [6–9]. Transition along path A is usually predicted using linear stability theory and the empirical e^N method.

If the disturbance has the proper initial condition, it will experience transient growth. Transient growth is a result of the coupling between oblique Orr-Sommerfeld and Squire modes, which leads to algebraic growth followed by exponential decay. Transient growth can

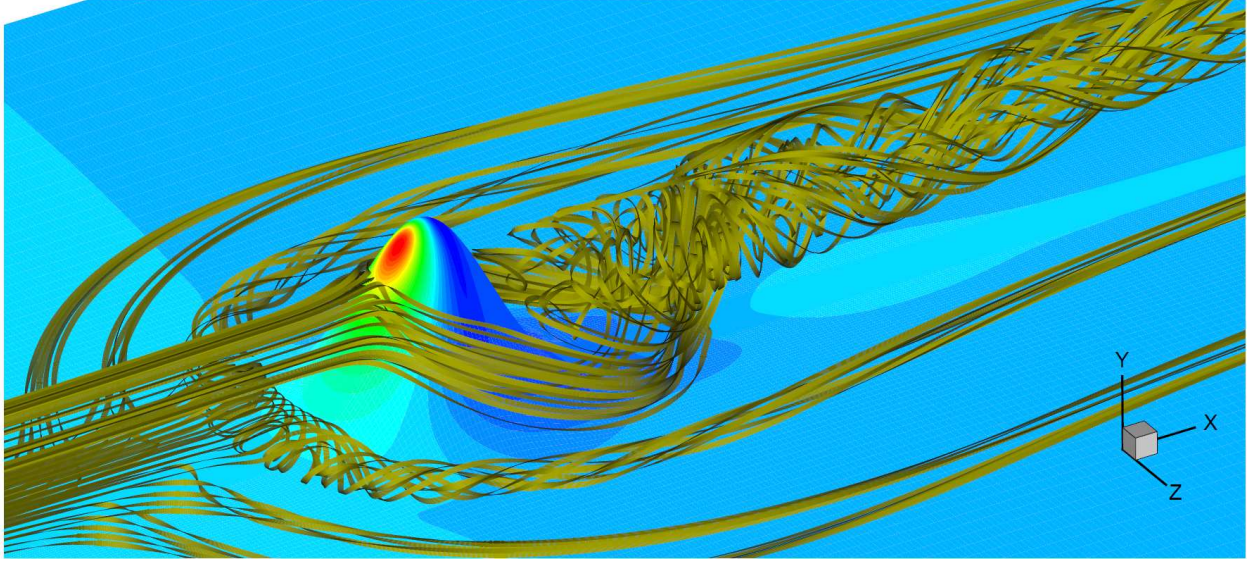


Figure 1.2: Streamlines for Mach 6 flow over an isolated Gaussian roughness. The pressure contour on roughness surface is also shown. The roughness element height is 73% of the boundary-layer thickness.

lead to turbulence along paths B, C, or D. Along path B, the transient growth provides a higher amplitude to the disturbance before undergoing eigenmode growth. Along path C, the transient growth is large enough to excite the secondary instabilities directly. The final path for transient growth leads to bypass transition. In bypass transition, the disturbances are so strong that the flow appears to be turbulent already. It is also possible to go straight to bypass transition from receptivity along path E if the initial disturbances are large enough.

1.2 Effects of Roughness on Hypersonic Transition

Roughness elements can be classified as either distributed or isolated. The work done here will focus on the effects of isolated roughness elements. Figure 1.2 shows the streamlines of Mach 6 flow over a Gaussian shaped roughness element whose height is 73% of the undisturbed boundary-layer thickness. The results come from a numerical simulation done as part of this work. The figure demonstrates many of the key features of hypersonic flow over isolated roughness elements. The flow clearly generates a pair of large counter-rotating streamwise vortices in the wake of the roughness. There is also a group of horse-shoe vortices

that start upstream of the roughness element. The horse-shoe vortices and the vortices in the roughness wake persist for very long distances downstream of the roughness. In this case, they are still visible at the domain outlet 25 boundary-layer thicknesses downstream. In addition to the vortices, roughness elements also generate a separation region upstream of the roughness which can create a shock. If the roughness is sufficiently tall, another shock may be generated by the roughness itself.

The effects of roughness elements have been documented for some time experimentally. Schneider [4] has recently written a review of experiments studying the effects of roughness on hypersonic boundary-layer transition. Roughness elements generally cause the location of transition to move forward compared to flow without a roughness element. It is possible to have a small enough roughness element such that there is no effect on the transition location. Such a roughness is said to be below its critical height. As the roughness height is increased past this critical height, the location of transition will continue to move closer to the roughness element. Eventually transition will occur at its closest location behind the roughness. This location may still be a finite distance downstream of the roughness. Further increases of the roughness height will no longer cause the transition location to move forward. The height at which this occurs is called the effective height. Although a great deal of research has been done to study the effects of roughness on hypersonic transition, most of the data was for correlations to be used in the design of hypersonic vehicles. Very little is actually known about the exact physics behind hypersonic boundary-layer transition with roughness elements.

In the recent past, a great deal of work went into exploring transient growth as an important role in roughness-induced transition. Tumin and Reshotko [10] presented a spatial theory for transient growth in boundary-layer flow. They showed that the most significant transient growth is associated with steady streamwise vortices. As shown in Figure 1.2, roughness elements generate a horse-shoe vortex and a pair of vortices in their wake. This is exactly the kind of disturbance that Tumin and Reshotko showed would lead to optimal transient growth. White et al. [11–13] have performed a series of experiments involving

subsonic flow over an array of cylindrical roughness elements and have identified transient growth in the wake of the array. Fischer and Choudhari [14] and Choudhari and Fischer [15] performed numerical simulations of White’s experiments. They also confirmed the presence of transient growth. Reshotko and Tumin [16] used their spatial transient growth theory to produce a model for roughness-induced transition. Their model reproduced the trends of Reda and the passive nosetip technology (PANT) data [17, 18].

Recent work has shifted from transient growth as the mechanism for transition in hypersonic flows with roughness elements, to the possible existence of an absolute instability being generated by the roughness. For taller roughness elements in subsonic flow, the wake region is dominated by unsteady hairpin vortices [19]. Ergin and White [20] conducted experiments of unsteady low-speed flow over isolated roughness elements. They found the locations of maximum fluctuation intensity to correspond to locations of inflection points in the streamwise velocity. This suggests that the unsteadiness seen in subsonic flow may have been caused by a Kelvin-Helmholtz type instability. The unsteady vortex shedding seen in subsonic flow is not present in supersonic flow [21]. This difference may be due to the stabilizing effect of high Mach numbers on shear layers [22]. Although the unsteadiness is no longer present, a similar instability may still remain.

Danehy et al. [23] visualized Mach 4.2 flow over an isolated 2 mm radius hemispherical roughness element using Nitric oxide planar laser-induced fluorescence (PLIF). They showed that breakdown to turbulence begins in the downstream legs of the horse-shoe vortex that forms upstream of the roughness. Chang et al. [24] recently performed numerical simulations that support these results. They found strong unsteadiness in Mach 4.16 flow over isolated cylindrical roughness elements with heights of 1.3 and 2.5 times the boundary-layer thickness with the unsteadiness first visible in the legs of the horse-shoe vortices. Spectral analysis and the computed evolution of disturbance root-mean-square amplitude suggest that the source for the unsteadiness is a possible absolute instability located in the separation region formed upstream of the roughness element. However, these vortices may not be the only possible source for an absolute instability. Choudhari et al. [25] numerically simulated Mach 3.5 flow

over diamond shaped isolated roughness elements with a range of heights. They showed that the large vortices found in the wake of the roughness can sustain strong high-frequency convective instabilities that can lead to early transition.

Bartkowicz et al [26]. have simulated flow inside the Boeing/AFOSR Mach 6 quiet wind tunnel at Purdue University with a cylindrical roughness element and reproduced an instability which was measured experimentally [27, 28]. They suggest that the instability is created by an unsteady high-momentum jet created near the top of the upstream side of roughness, which flows down into the vortex system located near the wall. Duan et al [29]. have also simulated flow under the conditions from the Purdue wind tunnel. They measured a different instability frequency than the experiments but this may have been due to discrepancies between the simulated geometry and the actual wind tunnel.

The goal of this work was to develop a code which can be used to study the effects of isolated roughness elements on the stability of hypersonic boundary layers. The code performs direct numerical simulations (DNS) of the governing equations for compressible fluid flow. There are two major difficulties when simulating high-speed flow over roughness elements. The first is modeling the roughness geometry. The second is resolving the large variety of length scales present in the problem.

1.3 Cartesian Grid Methods

The conventional approach to imposing geometries in DNS is to use a body-fitted grid [30]. In body-fitted grids, at least one of the edges of the computational domain will conform to the geometry. This makes the application of boundary conditions relatively simple. If the governing equations are discretized using a finite-difference method, then a structured grid would need to be used. If a finite-volume or finite-element method is used, then either a structured or unstructured grid is permitted. In either case, the generation of a body-fitted grid for complex geometries can be a very difficult and time-consuming process.

As an alternative to body-fitted grids, an assortment of Cartesian grid methods have

been developed. The commonality between all the methods is that they can impose complex geometries on simple Cartesian grids. Most Cartesian grid methods fall into one of two categories: immersed boundary methods or sharp-interface methods.

1.3.1 Immersed Boundary Method

The immersed boundary method was originally developed by Peskin [31, 32] in the 1970's. Peskin used the immersed boundary method to study blood flow in a beating heart. The blood flow was simulated on a Cartesian grid while the elastic walls were discretized on a Lagrangian grid. The effect of the elastic wall was imposed on the fluid through a source term added to the equations governing the fluid flow. The source term would have an approximate Dirac δ function associated with it which would limit its area of influence to near the walls. The use of a forcing term added to the governing equations is the defining characteristic of immersed boundary methods.

Goldstein et al. [33] extended the immersed boundary method for flows with immersed rigid walls. Goldstein et al. designed the forcing term to have a magnitude and direction opposing the local flow. This would drive the fluid to zero velocity at the immersed boundary. This method, and others like it, are usually referred to as continuous forcing methods. Mohd-Yusof [34] derived an alternative to this called the discrete forcing method. Mohd-Yusof changed the forcing term so that it would impose zero velocity at the boundary for all time. A number of other improvements and modifications have been made to the immersed boundary method since it was originally developed [35–42].

The primary advantage of the immersed boundary method is the ease of its implementation. Since the body is imposed through a source term, the method can be easily incorporated into an existing code without having to alter the underlining solver. The numerical method used in the solver is simply applied everywhere, including inside any immersed bodies and across their surfaces. This has led to the immersed boundary method being used to simulate a large variety of flows [31, 43–52]. Although used successfully for many problems,

the immersed boundary method has its disadvantages. Most applications of the immersed boundary method have been at low to moderate Reynolds numbers. At high Reynolds numbers, the spreading effect of the approximate delta function has a more detrimental effect on the flow since the local solution has greater importance [30]. The approximate Dirac δ function used in the method causes the interface of the body to be smeared. In addition, the method is only locally first order, which may not have the required accuracy for some applications.

1.3.2 Sharp-Interface Method

While the immersed boundary method uses a forcing term added to the governing equation, sharp-interface methods will alter the discretization of the governing equation near the body. The alteration allows boundary conditions to be applied just at the edge of the body. The methods generate precise locations for the body instead of smearing the interface as the immersed boundary method does. The two most common sharp-interface approaches are ghost fluid methods and cut-cell methods.

The ghost fluid method was developed by Fedkiw et al. [53, 54]. The method begins by creating a layer of ghost fluid (which can be cells or nodes depending on the discretization method) inside the immersed body. The values at the ghost points are extrapolated from the real fluid point in such a way that the boundary conditions are imposed at the interface exactly. The ghost fluid method is still being improved [55–59]. Similarly to the immersed boundary method, the interior discretization scheme is used across the body boundary, but now the values on the inside of the body are based on the physical fluid values.

The primary motivation of the cut-cell method was to ensure global and local conservation of mass and momentum [30]. To achieve this, the original formulation of the cut-cell method was made in a finite-volume discretization. Any cell which the edge of the body intersects is divided along the edge. The portion of the cells which are inside the body are discarded. If the center of the original cell is outside the body, then the remaining portion of the cell is

used as the new cell. If the center is inside the body, then the remaining portion is merged into a neighboring cell. Unlike the previously mentioned methods, no values from inside the immersed body are ever used in the calculations. The cut-cell method was originally developed by Clarke et al. [60] for inviscid flow. Udaykumar et al. [61–63] and Ye et al. [64] extended the method to viscous flow. Improving the cut-cell method is still an active research topic [65–69].

Compared to the immersed boundary methods, sharp-interface methods tend to be more complex to implement. Sharp-interface methods are also victim to the “small cell problem” [70]. The small cell problem occurs when a grid cell becomes much smaller than its neighbors due to it being near the body’s edge. For time dependent problems, this can severely limit the size of the permitted time step. Most sharp-interface methods have addressed this issue and provide a solution. The added complexity for sharp-interface methods is usually out weighted by the ability to precisely enforce the location of the immersed body’s edge. This can be seen in the wide variety of problems the methods have been applied to [56, 59, 64, 66, 71–75].

1.4 Adaptive Mesh Refinement

Figure 1.3 shows a contour of the streamwise velocity component for Mach 6 flow over a cylindrical roughness whose height is 75% of the undisturbed boundary layer thickness. The figure depicts the large variation in the length scales involved in problems of this nature. When the flow encounters the roughness element, a boundary layer is created along the roughness element. The thickness of this boundary layer can be more than an order of magnitude smaller than the undisturbed boundary layer. In addition, the streamwise length of the domain can be an order of magnitude larger than the boundary-layer thickness. Problems such as this can benefit greatly from the adaptive mesh refinement method.

The adaptive mesh refinement method was originally developed by Berger et al. [76–78]. The method begins by covering the entire computational domain with a coarse grid.

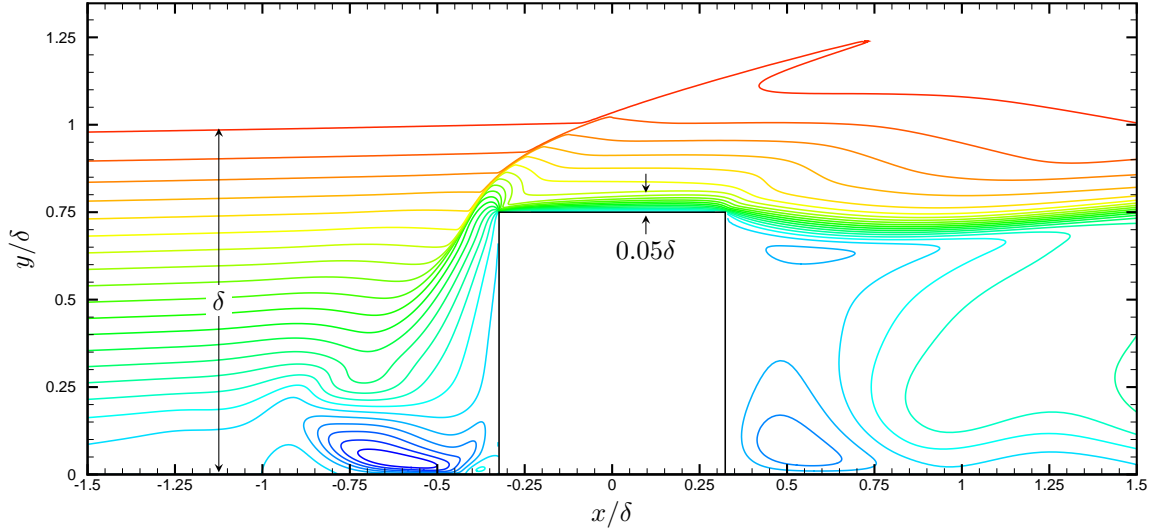


Figure 1.3: Comparison of length scales for Mach 6 flow over cylindrical roughness element. Streamwise velocity contour shown. The $x - y$ plane through the center of roughness is shown. The roughness height is 75% the undisturbed boundary layer thickness, δ .

Then smaller refinement grids are added at location where extra resolution is required. Refinement grids can be recursively nested until the desired level of accuracy is obtained. In addition to the increase in spatial resolution, the refinement grids are also refined in time. This is necessary to maintain the stability of explicit schemes. As the solution advances in time, the location of refinement grids can move with the complex physics. Refinement grids can also be removed if no longer needed or added if more resolution is required. Berger et al. used Richardson extrapolation to select locations where extra refinement was needed. Adaptive mesh refinement has proved to be an effective method for dealing with high localized resolution requirements [79–85]. The method has also been paired with various Cartesian grid methods [35, 86–89].

1.5 Goals of the Present Study

The work done here was motivated by the desire to study the effects of isolated roughness elements on the stability and transition of hypersonic boundary-layer flow. This original goal was intended to provide a better understanding of the physical mechanisms through which

three-dimensional roughness elements affect transition and assist in the creation of better transition prediction methods. This was to be achieved through direct numerical simulations of flow over isolated roughness elements. The development of the code meant to be used to perform these simulations proved more challenging than originally expected. As a result, the focus of this work has been on the development of a solver capable of simulating high-speed flows over arbitrary geometries.

The solver developed here performs direct numerical simulations of the unsteady compressible Navier-Stokes equations for a calorically perfect gas. The solver utilizes a fifth-order hybrid weighted essentially non-oscillatory scheme for the computation of the inviscid terms. A third-order cut-cell method is employed to impose the geometries. This allows the simulations to be performed on simple Cartesian grids. A multi-zone refinement method is used to precisely control grid resolution. The method is based on adaptive mesh refinement but without the adaptive portion of the method. The combination results in a globally fourth-order solver capable of simulating high-speed flows over complex geometries.

The order of convergence for the cut-cell and multi-zone refinement method were computed separately. Results will be presented showing the components achieve the expected order of convergence. Since the solver was recently implemented, results from two validation cases will be shown. The two cases are subsonic flow over a two-dimensional cylinder and supersonic flow over a three-dimensional hyperbolic tangent roughness. To demonstrate the capabilities of the new code, results from two simulations of flow over complex geometries will be shown. Preliminary results for Mach 6 boundary-layer flow over a cylindrical roughness will also be shown. Although the simulation is not complete, the results compare well with results published by other researchers. The simulation demonstrates that the combination of the multi-zone method and the cut-cell method create a promising tool for the study of hypersonic flow over isolated roughness elements.

In addition to the results from the multi-zone cut-cell code, results from an earlier code will also be presented. The older code was a curvilinear code which would perform simulations on a single body-fitted grid. This code was used to simulate flow in the Boeing/AFOSR

Mach 6 Quiet Tunnel at Purdue University. Originally, it was planned to simulate flow over isolated roughness elements attached to the wall of the Purdue wind tunnel. Due to difficulties with the simulation, the wind tunnel geometry was exchanged for a flat plate. Simulations of Mach 6 boundary-layer flow over a Gaussian roughness element were performed. Four different roughness heights were simulated and will be presented.

CHAPTER 2

Governing Equations

2.1 Cartesian Coordinate System

The governing equations in Cartesian coordinates, $(x_1, x_2, x_3) = (x, y, z)$, are the unsteady three-dimensional compressible Navier-Stokes equations for a calorically perfect gas with density ρ , pressure p , temperature T and velocity $(u_1, u_2, u_3) = (u, v, w)$:

$$\frac{\partial \hat{\mathbf{U}}}{\partial t} + \frac{\partial \hat{\mathbf{F}}_j}{\partial x_j} = \frac{\partial \hat{\mathbf{F}}_{v,j}}{\partial x_j}, \quad j = 1, 2, 3 \quad (2.1)$$

where $\hat{\mathbf{U}}$ is the vector of conservative variables, $[\rho, \rho u, \rho v, \rho w, E]^T$. $\hat{\mathbf{F}}_j$ and $\hat{\mathbf{F}}_{v,j}$ are the vectors of the inviscid and viscous fluxes, respectively, in the j th direction and are given as

$$\hat{\mathbf{F}}_j = \begin{pmatrix} \rho u_j \\ \rho u_1 u_j + \delta_{1j} p \\ \rho u_2 u_j + \delta_{2j} p \\ \rho u_3 u_j + \delta_{3j} p \\ u_j (E + p) \end{pmatrix}, \quad \text{and} \quad \hat{\mathbf{F}}_{v,j} = \begin{pmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \tau_{3j} \\ u_k \tau_{kj} - q_j \end{pmatrix}. \quad (2.2)$$

The total energy per unit mass E , the viscous stress tensor τ_{ij} , and the heat flux vector q_j are defined as

$$E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho u_i u_i, \quad (2.3)$$

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right), \quad (2.4)$$

$$q_j = -k \frac{\partial T}{\partial x_j}, \quad (2.5)$$

where μ is the fluid viscosity and k is the thermal conductivity. The system of equations is closed by the equation of state. Since the gas is calorically perfect, the equation of state is

$$p = \rho R T. \quad (2.6)$$

The gas is assumed to be air so the gas constant, R , has a value of $287.04 \text{ m}^2/\text{s}^2 \text{ K}$, the ratio of the specific heats, γ , is 1.4 and the Prandtl number, Pr , is 0.72. The viscosity is computed using Sutherland's law in the form

$$\mu = \mu_r \left(\frac{T}{T_r} \right)^{1.5} \frac{T_r + T_s}{T + T_s} \quad (2.7)$$

where $\mu_r = 1.7894 \times 10^{-5} \text{ (N s)}/\text{m}^2$, $T_s = 110.33 \text{ K}$, and $T_r = 288 \text{ K}$. All variables are left in dimensional form.

2.2 Curvilinear Coordinate System

Before solving the Navier-Stokes equations, they are transformed into curvilinear coordinates. The transformation takes the governing equations from (x, y, z) physical space and projects them to a square uniform grid in $(\xi, \eta, \zeta) = (\xi_1, \xi_2, \xi_3)$ computational space. The transformation is represented by

$$x = x(\xi, \eta, \zeta), \quad y = y(\xi, \eta, \zeta), \quad z = z(\xi, \eta, \zeta) \quad (2.8)$$

where ξ , η , and ζ each go from 0 to 1.

Derivatives in (x, y, z) space can be related to derivatives in (ξ, η, ζ) space by

$$\frac{\partial}{\partial x_i} = \frac{\partial \xi_j}{\partial x_i} \frac{\partial}{\partial \xi_j} \equiv \xi_i^j \frac{\partial}{\partial \xi_j}. \quad (2.9)$$

The terms $\frac{\partial \xi_j}{\partial x_i} \equiv \xi_i^j$ are called the metrics. The derivation of these terms can be found in most computational fluid dynamics books [90] and will not be shown here. By using Equation 2.9 the governing equations can be transformed into curvilinear coordinates. After the transformation, the governing equations become

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j}{\partial \xi_j} = \frac{\partial \mathbf{F}_{v,j}}{\partial \xi_j}, \quad j = 1, 2, 3 \quad (2.10)$$

where \mathbf{U} is the vector of the conservative variables, $[\rho, \rho u, \rho v, \rho w, E]^T / J$. J is the determinate of the the Jacobian matrix:

$$J = \left| \frac{\partial (\xi, \eta, \zeta)}{\partial (x, y, z)} \right|. \quad (2.11)$$

\mathbf{F}_j and $\mathbf{F}_{v,j}$ are the vectors of the inviscid and viscous fluxes, respectively, in the j th direction of the computational space and are given as

$$\mathbf{F}_j = \frac{1}{J} \begin{pmatrix} \rho U_j \\ \rho u_1 U_j + \delta_{1j} \xi_1^j p \\ \rho u_2 U_j + \delta_{2j} \xi_2^j p \\ \rho u_3 U_j + \delta_{3j} \xi_3^j p \\ U_j (E + p) \end{pmatrix}, \quad \text{and} \quad \mathbf{F}_{v,j} = \frac{1}{J} \begin{pmatrix} 0 \\ \xi_k^j \tau_{1k} \\ \xi_k^j \tau_{2k} \\ \xi_k^j \tau_{3k} \\ \xi_k^j u_n \tau_{nk} - \xi_k^j q_k \end{pmatrix} \quad (2.12)$$

where U_j is given by

$$U_j = u_k \xi_k^j. \quad (2.13)$$

The shear stresses and heat fluxes in curvilinear coordinates are

$$\tau_{ij} = \mu \left(\xi_j^k \frac{\partial u_i}{\partial \xi_k} + \xi_i^k \frac{\partial u_j}{\partial \xi_k} - \frac{2}{3} \delta_{ij} \xi_k^n \frac{\partial u_k}{\partial \xi_n} \right), \quad (2.14)$$

$$q_j = -k \xi_j^k \frac{\partial T}{\partial \xi_k}. \quad (2.15)$$

CHAPTER 3

Numerical Method

The code used in this work is comprised of three main parts: a Navier-Stokes solver, the cut-cell method, and the multi-zone refinement method. Each component will be covered separately, starting with the Navier-Stokes solver.

3.1 Navier-Stokes Solver

The governing equations are solved using the method of lines. The method of lines convert the governing partial differential equations into a system of interdependent ordinary differential equations that approximates the original equations. The system may then be advanced in time using a Runge-Kutta method. To start, Equation 2.10 is rewritten as

$$\begin{aligned}\frac{\partial \mathbf{U}}{\partial t} &= -\frac{\partial \mathbf{F}_j}{\partial \xi_j} + \frac{\partial \mathbf{F}_{v,j}}{\partial \xi_j}, \quad j = 1, 2, 3 \\ &\equiv L_j(\mathbf{U}) + L_{v,j}(\mathbf{U}) \\ &\equiv L(\mathbf{U}).\end{aligned}\tag{3.1}$$

An approximation for the spatial operators L_j and $L_{v,j}$ must be found before advancing the equations in time. In this work, the inviscid terms L_j were approximated with a hybrid finite-difference scheme. The scheme was a combination of a fifth-order weighted essentially non-oscillatory (WENO) scheme and a fifth-order low-dissipation scheme. Since a majority of the flows studied in this work are supersonic, they tend to develop discontinuities and steep gradients which require a shock-capturing scheme to be employed. WENO fills this

role. However, WENO is computationally expensive so the scheme should only be used where it is necessary. This is why WENO is paired with a computational cheap low-dissipation finite-difference scheme. At locations in the flow with discontinuities and steep gradients WENO is used and everywhere else the low-dissipation scheme is used.

The approximation to the viscous term $L_{v,j}$ does not require as much special attention as the inviscid term. A simple central finite-difference scheme was utilized to approximate $L_{v,j}$. In order to be consistent with the inviscid approximation, a sixth-order approximation is utilized.

3.1.1 Inviscid Terms

Due to the physical nature of the inviscid flux terms, some form of upwinding must be present in the approximation to L_j . This is due to the fact that the Jacobian matrix $\frac{\partial \mathbf{F}_j}{\partial \mathbf{U}}$ can have both positive and negative eigenvalues. Local Lax-Friedrichs flux splitting was employed for upwinding. Local Lax-Friedrichs splitting divides the fluxes into two terms:

$$\mathbf{F}_j = \mathbf{F}_j^+ + \mathbf{F}_j^-. \quad (3.2)$$

One term will only have positive eigenvalues (\mathbf{F}_j^+) and the other will only have negative eigenvalues (\mathbf{F}_j^-). With the fluxes split, an upwind stencil or downwind stencil can be applied according to the sign of the eigenvalues.

The flux is split by adding and subtracting a constant times the conservative variables from the fluxes. For the j th direction, the flux is split according to

$$\mathbf{F}_j^\pm = \frac{1}{2} (\mathbf{F}_j \pm \lambda_j \mathbf{U}) \quad (3.3)$$

where λ_j is chosen to be larger than the magnitude of all the eigenvalues of the local Jacobian

in the j th direction. For the Navier-Stokes equations in curvilinear coordinates, λ_j is

$$\lambda_j = \sqrt{(u_k \xi_k^j)^2 + \epsilon_\lambda^2 c^2 \left((\xi_1^j)^2 + (\xi_2^j)^2 + (\xi_3^j)^2 \right)} + c \sqrt{(\xi_1^j)^2 + (\xi_2^j)^2 + (\xi_3^j)^2} \quad (3.4)$$

where c is the speed of sound and ϵ_λ is 0.1. When ϵ_λ is zero, λ_j is equal to the magnitude of the largest eigenvalue of the local Jacobian.

3.1.1.1 Hybrid Scheme

The inviscid terms are computed using a fifth-order finite-difference hybrid scheme which combines WENO with a low-dissipation scheme. The hybrid scheme begins by writing the derivatives in conservative form. As an example, the derivative of the inviscid flux in the ξ direction with respect to ξ at the point ξ_i is computed as

$$\frac{\partial \mathbf{F}_\xi}{\partial \xi}(\xi_i) = \frac{\hat{\mathbf{F}}_\xi^+(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^+(\xi_{i-1/2})}{\Delta \xi} + \frac{\hat{\mathbf{F}}_\xi^-(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^-(\xi_{i-1/2})}{\Delta \xi}. \quad (3.5)$$

$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2})$ is an approximation to the flux at the interface $\xi_{i+1/2}$. The interface flux is a linear combination of a fifth-order WENO scheme and a fifth-order low-dissipation scheme. The interface flux is

$$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2}) = (1 - \sigma_{i+1/2}) \hat{\mathbf{F}}_\xi^{\pm \text{WENO}}(\xi_{i+1/2}) + \sigma_{i+1/2} \hat{\mathbf{F}}_\xi^{\pm \text{LD}}(\xi_{i+1/2}) \quad (3.6)$$

where $\sigma_{i+1/2}$ is the hybrid switch and is based on the work of Ren et al. [91]. The switch is computed as

$$\sigma_{i+1/2} = \begin{cases} 0 & \text{if } r_{i+1/2} < r_c \\ 1 & \text{otherwise} \end{cases} \quad (3.7)$$

where $r_{i+1/2}$ is the smoothness indicator and r_c is a smoothness cutoff. In this work, r_c was set to 0.8. The smoothness indicator is

$$r_{i+1/2} = \min(r_{i-1}, r_i, r_{i+1}, r_{i+2}) \quad (3.8)$$

where

$$r_i = \frac{|2\Delta\rho_{i+1/2}\Delta\rho_{i-1/2}| + \epsilon_r}{(\Delta\rho_{i+1/2})^2 + (\Delta\rho_{i-1/2})^2 + \epsilon_r}, \quad (3.9)$$

and $\Delta\rho_{i+1/2} = \rho_{i+1} - \rho_i$. ϵ_r is a small number used to prevent the denominator from going to zero. The value used is

$$\epsilon_r = \frac{0.9 r_c}{1 - 0.9 r_c} \nu^2 \rho_{\text{ref}}^2 \quad (3.10)$$

where ρ_{ref} is the reference density for the problem and ν is 10^{-3} .

There are a number of differences between the hybrid switch used in this work and the version presented in Ren et al. First, in Ren et al., the smoothness indicators were computed from the fluxes instead of just density. This meant there was a hybrid switch value for each component of the flux. In this work, the smoothness indicators were computed just from density resulting in a single hybrid switch being used for the entire flux vector. The change was made to reduce the number of required calculations and thus reduce computation time. Second, Ren et al. used a continuous function for $\sigma_{i+1/2}$ instead of the on/off switch used in the current work. A switch was used here to ensure WENO was turned on everywhere it was needed and thus improve numerical stability. Lastly, the number of values used to compute $r_{i+1/2}$ is larger in the current work. Ren et al. combined WENO with a compact scheme, while in this work, WENO was combined with an explicit finite-difference scheme. This means, the non-WENO stencil is larger in the current work compared to Ren et al. To accommodate this change, more points were included in the calculation of $r_{i+1/2}$.

The calculation of the low-dissipation flux and the WENO flux will be presented separately. As done so far, only the fluxes for the derivative of the inviscid flux in the ξ direction with respect to ξ at the point ξ_i will be shown. It is straightforward to extend the calculations to other dimensions.

3.1.1.2 Low-Dissipation Scheme

The low-dissipation flux used here is a conservative version of the scheme presented by Zhong [92]. Zhong developed a stable upwind fifth-order finite-difference scheme which uses

a central stencil. The use of a central stencil instead of an upwind-biased stencil with one less point generates a scheme which has an adjustable constant. The constant can be used to control the scheme's dissipation and stability. The scheme presented by Zhong was for a direct calculation of a derivative from the grid point fluxes. To be compatible with Equation 3.6, the scheme was reformulated in conservative form. The low-dissipation flux for the positive eigenvalue fluxes is computed as

$$\hat{\mathbf{F}}_{\xi}^{+LD}(\xi_{i+1/2}) = \sum_{j=-2}^3 a_j \mathbf{F}_{\xi}^{+}(\xi_{i+j}) \quad (3.11)$$

where the constants a_j are given in Table 3.1. The constants for the negative eigenvalue fluxes can be found from symmetry. The α in the constants controls the scheme's dissipation and stability and is equivalent to the α from Zhong [92]. In this work, α was set to -6 .

j	a_j
-2	$1 - \frac{1}{12}\alpha$
-1	$-8 + \frac{5}{12}\alpha$
0	$37 - \frac{5}{6}\alpha$
1	$37 + \frac{5}{6}\alpha$
2	$-8 - \frac{5}{12}\alpha$
3	$1 + \frac{1}{12}\alpha$

Table 3.1: Low-dissipation scheme constants for positive eigenvalue interface flux.

3.1.1.3 Weighted Essentially Non-Oscillatory Scheme

WENO schemes are a family of shock-capturing schemes which can be designed to obtain a high-order of convergence when shocks and high gradients are not present in the solution. The method works by taking a high-order stencil and breaking it into smaller sub-stencils. Figure 3.1 shows the high-order stencil that fifth-order WENO is based on and the WENO sub-stencils that the high-order stencil is divided into. The interface flux is computed on

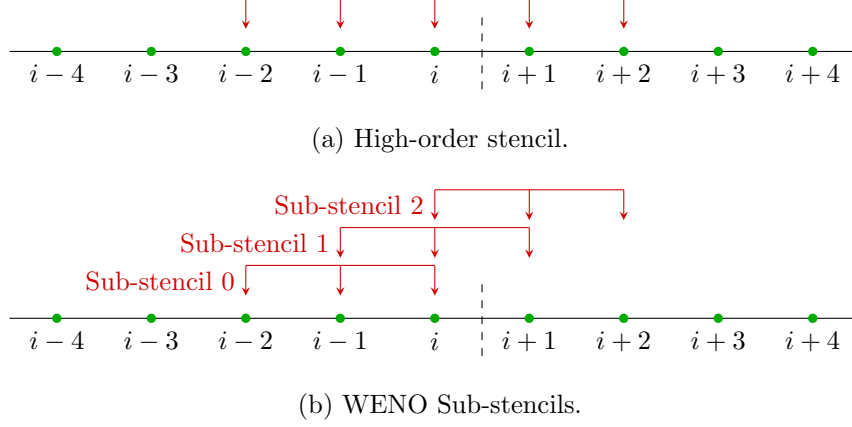


Figure 3.1: High-order stencil and sub-stencils used in fifth-order WENO. The stencils are for computing the positive eigenvalue flux at the interface between points i and $i + 1$. The interface is shown as ---.

each sub-stencil and then linearly combined with weights based on the smoothness of the solution on each sub-stencil. The weights cause the scheme to favor the sub-stencils with the smoothest solution. The weights are designed such that when the solution on all the sub-stencils is smooth, the sub-stencil fluxes combine to obtain the same values as the high-order stencil.

The original formulation of the fifth-order finite-difference WENO scheme comes from Jiang and Shu [93]. However, a slightly different version was used in this work. The version used here comes from Henrick et al. [94]. Henrick et al. addressed some issues with the WENO scheme developed by Jiang and Shu. Under certain conditions, the scheme presented by Jiang and Shu would only obtain third-order convergence instead of the intended fifth-order convergence. Their solution was to apply a mapping to the WENO weights of Jiang and Shu. The resulting scheme is called mapped WENO. The process for computing the positive eigenvalue interface flux $\left(\hat{\mathbf{F}}_{\xi}^{+\text{WENO}}(\xi_{i+1/2})\right)$ will be covered here. The interface flux for the negative eigenvalue can be found from symmetry.

WENO is applied in local characteristic space to avoid spurious oscillations in the solution. To transform the fluxes into characteristic space, the right and left eigenvectors of the Jacobians $\partial \mathbf{F}_{\xi} / \partial \mathbf{U}$, $\partial \mathbf{F}_{\eta} / \partial \mathbf{U}$, and $\partial \mathbf{F}_{\zeta} / \partial \mathbf{U}$ are needed. For three-dimensional problems, the

right eigenvectors for $\partial \mathbf{F}_\xi / \partial \mathbf{U}$ are the columns of

$$\mathbf{R} = \begin{pmatrix} 1 & \tilde{\xi}_x & \tilde{\xi}_y & \tilde{\xi}_z & 1 \\ u - \tilde{\xi}_x c & u \tilde{\xi}_x & u \tilde{\xi}_y - \tilde{\xi}_z \rho & \tilde{\xi}_x \rho + u \tilde{\xi}_z & u + \tilde{\xi}_x c \\ v - \tilde{\xi}_y c & \tilde{\xi}_z \rho + v \tilde{\xi}_x & v \tilde{\xi}_y & v \tilde{\xi}_z - \tilde{\xi}_x \rho & v + \tilde{\xi}_y c \\ w - \tilde{\xi}_z c & w \tilde{\xi}_x - \tilde{\xi}_y \rho & \tilde{\xi}_x \rho + w \tilde{\xi}_y & w \tilde{\xi}_z & w + \tilde{\xi}_z c \\ H - \tilde{\theta}_\xi c & -B_x \rho + b_2 \tilde{\xi}_x & -B_y \rho + b_2 \tilde{\xi}_y & -B_z \rho + b_2 \tilde{\xi}_z & H + \tilde{\theta}_\xi c \end{pmatrix} \quad (3.12)$$

and the left eigenvectors are the rows of

$$\mathbf{L} = \frac{1}{2\rho} \begin{pmatrix} \rho(b_1 b_2 + \tilde{\theta}_\xi / c) & -\rho(b_1 u + \tilde{\xi}_x / c) & -\rho(b_1 v + \tilde{\xi}_y / c) & -\rho(b_1 w + \tilde{\xi}_z / c) & \rho b_1 \\ 2B_x + 2b_3 \tilde{\xi}_x \rho & 2\rho b_1 \tilde{\xi}_x u & 2(b_1 \tilde{\xi}_x \rho v + \tilde{\xi}_z) & 2(b_1 \tilde{\xi}_x \rho w - \tilde{\xi}_y) & -2\rho b_1 \tilde{\xi}_x \\ 2B_y + 2b_3 \tilde{\xi}_y \rho & 2(b_1 \tilde{\xi}_y \rho u - \tilde{\xi}_z) & 2b_1 \tilde{\xi}_y \rho v & 2(b_1 \tilde{\xi}_y \rho w + \tilde{\xi}_x) & -2\rho b_1 \tilde{\xi}_y \\ 2B_z + 2b_3 \tilde{\xi}_z \rho & 2(b_1 \tilde{\xi}_z \rho u + \tilde{\xi}_y) & 2(b_1 \tilde{\xi}_z \rho v - \tilde{\xi}_x) & 2b_1 \tilde{\xi}_z \rho w & -2\rho b_1 \tilde{\xi}_z \\ \rho(b_1 b_2 - \tilde{\theta}_\xi / c) & -\rho(b_1 u - \tilde{\xi}_x / c) & -\rho(b_1 v - \tilde{\xi}_y / c) & -\rho(b_1 w - \tilde{\xi}_z / c) & \rho b_1 \end{pmatrix} \quad (3.13)$$

where

$$b_1 = \frac{\gamma - 1}{c^2}, \quad b_2 = \frac{1}{2}(u^2 + v^2 + w^2), \quad b_3 = 1 - b_1 b_2, \quad (3.14a)$$

$$B_x = \tilde{\xi}_y w - \tilde{\xi}_z v, \quad B_y = \tilde{\xi}_z u - \tilde{\xi}_x w, \quad B_z = \tilde{\xi}_x v - \tilde{\xi}_y u, \quad (3.14b)$$

$$\tilde{\xi}_x = \frac{\xi_x}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}, \quad \tilde{\xi}_y = \frac{\xi_y}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}, \quad \tilde{\xi}_z = \frac{\xi_z}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}, \quad (3.14c)$$

$$H = \frac{e + p}{\rho}, \quad \tilde{\theta}_\xi = \tilde{\xi}_x u + \tilde{\xi}_y v + \tilde{\xi}_z w. \quad (3.14d)$$

ξ_x , ξ_y , and ξ_z are the metrics ξ_1^1 , ξ_2^1 , and ξ_3^1 , respectively. The right and left eigenvectors for $\partial \mathbf{F}_\eta / \partial \mathbf{u}$ or $\partial \mathbf{F}_\zeta / \partial \mathbf{u}$ can be computed from Equations 3.12 and 3.13 by substituting the corresponding metrics. The eigenvectors for the two-dimensional case cannot be easily derived from the three-dimensional case. As such, they are provided in Appendix A.1.

To compute the WENO flux at the interface $\xi_{i+1/2}$, all the fluxes used in the sub-stencils shown in Figure 3.1b must be transformed into characteristic space. To transform into char-

acteristic space, the flux vector \mathbf{F}_ξ is multiplied on the left by the matrix of left eigenvectors. Let \mathbf{l}_s denote the s th left eigenvector (s th row of Equation 3.13). The eigenvectors must be evaluated with some average of the values at ξ_i and ξ_{i+1} . In this work, a Roe average is utilized. The expressions for computing the Roe average are provided in Appendix A.2. The s th component of the characteristic flux is given as

$$\tilde{\mathbf{F}}_{\xi,s}^+(\xi_{i+j}) = \mathbf{l}_s \cdot \mathbf{F}_\xi^+(\xi_{i+j}), \quad j = -2, \dots, 2. \quad (3.15)$$

With the characteristic fluxes known, the interface flux for each of the WENO sub-stencils can be computed. The sub-stencil interface flux is computed as

$$\begin{aligned} \mathbf{q}_0 &= \frac{1}{6} \left(2\tilde{\mathbf{F}}_\xi^+(\xi_{i-2}) - 7\tilde{\mathbf{F}}_\xi^+(\xi_{i-1}) + 11\tilde{\mathbf{F}}_\xi^+(\xi_i) \right), \\ \mathbf{q}_1 &= \frac{1}{6} \left(-\tilde{\mathbf{F}}_\xi^+(\xi_{i-1}) + 5\tilde{\mathbf{F}}_\xi^+(\xi_i) + 2\tilde{\mathbf{F}}_\xi^+(\xi_{i+1}) \right), \\ \mathbf{q}_2 &= \frac{1}{6} \left(2\tilde{\mathbf{F}}_\xi^+(\xi_i) + 5\tilde{\mathbf{F}}_\xi^+(\xi_{i+1}) - \tilde{\mathbf{F}}_\xi^+(\xi_{i+2}) \right). \end{aligned} \quad (3.16)$$

The fluxes are linearly combined to obtain the characteristic WENO interface flux. The s th component of the characteristic WENO flux is given as

$$\tilde{\mathbf{F}}_{\xi,s}^{+\text{WENO}}(\xi_{i+1/2}) = \omega_{0,s}^m \mathbf{q}_{0,s} + \omega_{1,s}^m \mathbf{q}_{1,s} + \omega_{2,s}^m \mathbf{q}_{2,s} \quad (3.17)$$

where $\omega_{k,s}^m$ is the mapped weight for the s th component of the flux on the k th sub-stencil.

The original weights from Jiang and Shu are

$$\omega_{k,s} = \frac{\alpha_{k,s}}{\alpha_{0,s} + \alpha_{1,s} + \alpha_{2,s}} \quad (3.18)$$

where

$$\alpha_{k,s} = \frac{\tilde{\omega}_k}{(\beta_{k,s} + \epsilon)^2}. \quad (3.19)$$

$\tilde{\omega}_k$ are the ideal weights which would result in Equation 3.17 being equivalent to using the

high-order stencil. The values for $\tilde{\omega}_k$ are

$$\tilde{\omega}_0 = \frac{1}{10}, \quad \tilde{\omega}_1 = \frac{6}{10}, \quad \tilde{\omega}_2 = \frac{3}{10}. \quad (3.20)$$

ϵ is a small number which prevents the denominator from going to zero. In this work, ϵ was set to 10^{-40} . $\beta_{k,s}$ is the smoothness indicator for the s th component on the k th sub-stencil.

The smoothness indicators are computed as

$$\begin{aligned} \beta_0 &= \frac{13}{12} \left(\tilde{\mathbf{F}}_\xi^+(\xi_{i-2}) - 2\tilde{\mathbf{F}}_\xi^+(\xi_{i-1}) + \tilde{\mathbf{F}}_\xi^+(\xi_i) \right)^2 + \frac{1}{4} \left(\tilde{\mathbf{F}}_\xi^+(\xi_{i-2}) - 4\tilde{\mathbf{F}}_\xi^+(\xi_{i-1}) + 3\tilde{\mathbf{F}}_\xi^+(\xi_i) \right)^2, \\ \beta_1 &= \frac{13}{12} \left(\tilde{\mathbf{F}}_\xi^+(\xi_{i-1}) - 2\tilde{\mathbf{F}}_\xi^+(\xi_i) + \tilde{\mathbf{F}}_\xi^+(\xi_{i+1}) \right)^2 + \frac{1}{4} \left(\tilde{\mathbf{F}}_\xi^+(\xi_{i-1}) - \tilde{\mathbf{F}}_\xi^+(\xi_{i+1}) \right)^2, \\ \beta_2 &= \frac{13}{12} \left(\tilde{\mathbf{F}}_\xi^+(\xi_i) - 2\tilde{\mathbf{F}}_\xi^+(\xi_{i+1}) + \tilde{\mathbf{F}}_\xi^+(\xi_{i+2}) \right)^2 + \frac{1}{4} \left(3\tilde{\mathbf{F}}_\xi^+(\xi_i) - 4\tilde{\mathbf{F}}_\xi^+(\xi_{i+1}) + \tilde{\mathbf{F}}_\xi^+(\xi_{i+2}) \right)^2. \end{aligned} \quad (3.21)$$

The mapped weights are given by the expression

$$\omega_{k,s}^m = \frac{\alpha_{k,s}^*}{\alpha_{0,s}^* + \alpha_{1,s}^* + \alpha_{2,s}^*} \quad (3.22)$$

where $\alpha_{k,s}^* = g_k(\omega_{k,s})$. $g_k(\omega)$ is the mapping function from Henrick et al. and is given by

$$g_k(\omega) = \frac{\omega (\tilde{\omega}_k + \tilde{\omega}_k^2 - 3\tilde{\omega}_k\omega + \omega^2)}{\tilde{\omega}_k^2 + \omega (1 - 2\tilde{\omega}_k)}. \quad (3.23)$$

With the characteristic WENO interface flux known, it can be transformed back into component space to obtain the WENO interface flux. Let \mathbf{r}_s denote the s th right eigenvector (s th column of Equation 3.12). The transformation back into component space is given by

$$\hat{\mathbf{F}}_\xi^{+\text{WENO}}(\xi_{i+1/2}) = \sum_{s=1}^5 \tilde{\mathbf{F}}_{\xi,s}^{+\text{WENO}}(\xi_{i+1/2}) \mathbf{r}_s. \quad (3.24)$$

3.1.2 Viscous Terms

The derivative of the viscous fluxes in Equation 3.1 are computed with a sixth-order central finite-difference scheme. The viscous fluxes are a function of the shear stress and heat flux. These values also contain derivatives. These derivatives are computed with the same sixth-order central finite-difference scheme. The derivative of a quantity $f(\xi_i)$ with respect to ξ at point ξ_i is given by the expression

$$\frac{\partial f}{\partial \xi}(\xi_i) = \frac{1}{\Delta \xi} \sum_{j=-3}^3 a_j f(\xi_{i+j}) \quad (3.25)$$

where the constants a_j are given in Table 3.2.

j	a_j
-3	$-\frac{1}{60}$
-2	$\frac{9}{60}$
-1	$-\frac{45}{60}$
0	0
1	$\frac{45}{60}$
2	$-\frac{9}{60}$
3	$\frac{1}{60}$

Table 3.2: Constants for sixth-order central finite-difference scheme used for viscous derivatives.

3.1.3 Time Advancement

To advance the solution in time, a third-order total variation diminishing (TVD) Runge-Kutta scheme was utilized. The scheme comes from Jiang and Shu [93]. The equation to advance in time is

$$\frac{\partial \mathbf{U}}{\partial t} = L(\mathbf{U}). \quad (3.26)$$

To advance from step n to $n+1$, the third-order TVD Runge-Kutta scheme is

$$\begin{aligned}\mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t L(\mathbf{U}^n), \\ \mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{U}^{(1)}), \\ \mathbf{U}^{n+1} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{U}^{(2)}).\end{aligned}\tag{3.27}$$

3.2 Cut-Cell Method

The cut-cell method used in this work is based on the method from Duan et al. [66]. Duan et al. developed a globally fourth-order method for simulating flows with non-body-conforming grids. The method allows numerical simulations of flow over complex geometries to be performed on simple Cartesian grids. The main idea behind the cut-cell method of Duan et al. is to switch to a non-uniform one-sided finite-difference stencil whenever the normally used stencil will cross the immersed boundary. A summary of the method will be presented here with the differences between the current implementation and the original implementation of Duan et al. pointed out.

3.2.1 Point Classification

The cut-cell method begins by classifying all the grid points as either a regular point, an irregular point, a dropped point, or a boundary point. Figure 3.2 shows the classification for a sample ξ grid line. The immersed boundary intersects the grid line at the point ξ_1 . The solid is assumed to be to the left of ξ_1 and the fluid to the right. The point ξ_1 is classified as a boundary point. Boundary points are created anywhere the immersed surface crosses a grid line. Boundary points are not part of the original Cartesian grid. All the points to the left of ξ_1 are classified as dropped points. Dropped points are usually points within the immersed surface and should not be used in any of the calculations. The first point to the right of ξ_1 is classified as dropped in the ξ direction only. Although this point is in the fluid, the point is treated as a dropped point while performing computations in the ξ direction. The point

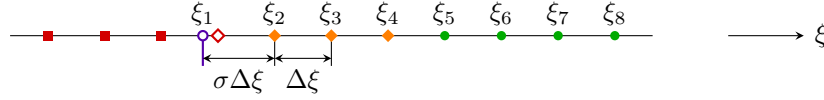


Figure 3.2: Point classification for ξ grid line with $\Theta \approx 1/3$. Immersed boundary —; Regular point ●; Irregular point ◆; Boundary point ○; Dropped point ■; Dropped point in ξ direction only ◇.

was dropped because it was deemed too close to the boundary point and its inclusion would severely limit the time step size. The criteria for determining when a point is too close is based on a specified parameter Θ and will be explained later. ξ_2 , ξ_3 , and ξ_4 are all classified as irregular points. Irregular points are any points whose finite-difference stencil would normally include a dropped point. The points ξ_5 to ξ_8 are regular points. Regular points are any points whose finite-difference stencil does not include any dropped points. When applying the method to two-dimensional and three-dimensional problems, a grid point may have a different classification in each of the computational directions. Figure 3.3 shows the point classification for a sample two-dimensional domain. The treatment for regular, irregular, boundary, and dropped points will be covered one at a time.

3.2.2 Dropped Points

There are two different types of dropped points. The first type is when a point is dropped in all directions of the computational domain. In this case, the point is never used in the calculations and therefore no special treatment is needed. All the points inside the immersed surface are of this type. The second type is when a point is dropped in at least one computational direction but not all of them. If the distance from a grid point to a boundary point along a grid line divided by the point spacing for that grid line is less than the specified value Θ , then the grid point is dropped in the direction of that grid line. For the fourth-order cut-cell method, Duan et al. recommend a value of 1.0 for Θ in all directions of the computational domain. In this work, separate Θ 's were used in each of the directions. For all the two-dimensional problems done in this work, Θ for the x direction was set 0.25 and Θ for the y direction was set to 0.75. For three-dimensional problems, the same values were

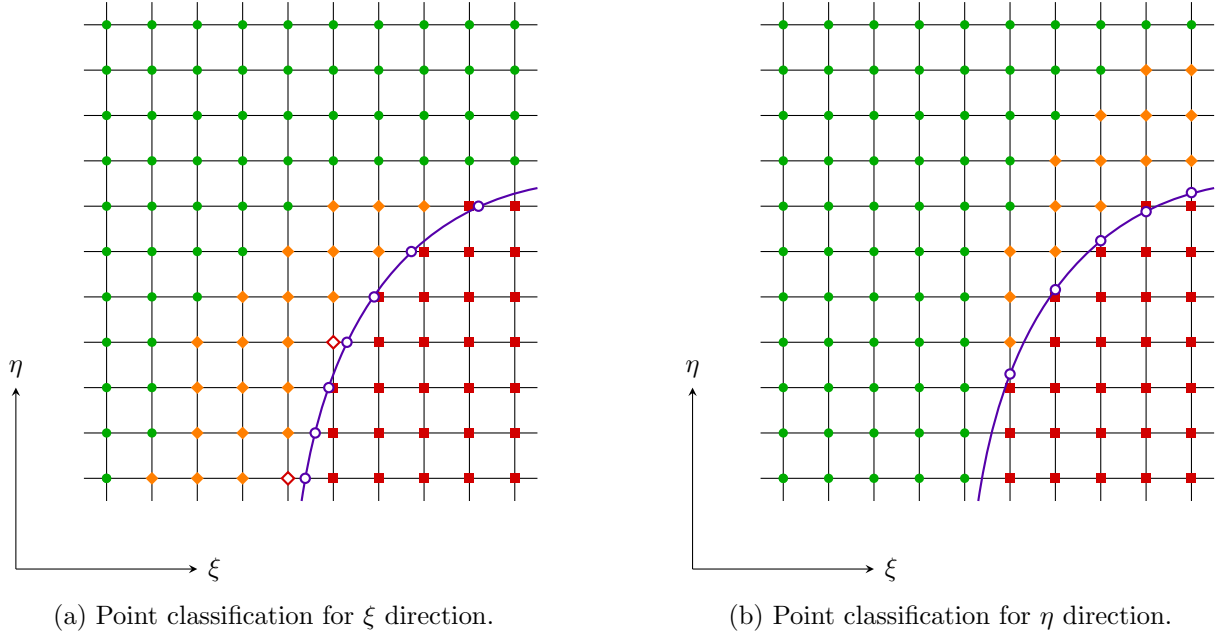


Figure 3.3: Point classification for two-dimensional computational space for $\Theta \approx 1/3$. Immersed boundary —; Regular point ●; Irregular point ◆; Boundary point ○; Dropped point in both directions ■; Dropped point in specified direction only ◇.

used in the x and y directions and the Θ in the z direction was set to 0.25.

For the second type of dropped point, the point will not be used in stencils in the dropped direction(s) but can be used in the remaining direction(s). Due to this, the values at these points must be computed but they cannot be advanced in time like points which are regular or irregular in all the directions of the domain. Duan et al. interpolated the conservative flow variables at these points using a fourth-order interpolation along the grid line in one of the dropped directions. The interpolation stencil always includes the boundary point adjacent to the dropped point. In this work, the conservative flow variables at these points are computed using a second-degree least-squares polynomial in x , y , and z of the form

$$\begin{aligned} \mathbf{U}(x, y, z) = & c_1 + c_2\Delta x + c_3\Delta y + c_4\Delta z + c_5\Delta x\Delta y + c_6\Delta x\Delta z + c_7\Delta y\Delta z \\ & + c_8\Delta x^2 + c_9\Delta y^2 + c_{10}\Delta z^2 \end{aligned} \quad (3.28)$$

where

$$\Delta x = x - x_d, \quad \Delta y = y - y_d, \quad \Delta z = z - z_d, \quad (3.29)$$

and x_d , y_d , and z_d are the x , y , and z values of the dropped point. For three-dimensional simulations, the constants are computed from the 35 closest points by index to the dropped point. For two-dimensional flows, the terms with Δz were removed and only the closest 10 points are used. The points can include regular, irregular, and boundary points. Other dropped points cannot be included in the points. Including them in the interpolation will cause the flow variables at one dropped point to be related to the flow variables at all the other dropped points resulting in a large implicit system. By not including the other dropped points, the flow variables at each point may be computed individually.

In addition to computing the flow variables at points dropped in one or more directions, the shear stresses and heat fluxes must also be computed at these points. Duan et al. do not mention how they compute the shear stresses and heat fluxes at these points. In this work, they are computed using a second-degree least-squares polynomial and the same set of grid points as the conservative flow variables.

3.2.3 Regular Points

Regular points are grid points whose finite-difference stencil does not contain any dropped points. This allows the regular interior finite-difference stencils to be used and hence does not require any special attention. In the original implementation by Duan et al., the fifth-order upwind scheme of Zhong [92] was used at regular points. In the current work, the scheme covered in Section 3.1 was used at these points.

3.2.4 Irregular Points

Irregular points are any points whose normal finite-difference stencil includes dropped points. Since dropped points should not be used in the calculations, the finite-difference stencil for irregular points must be changed. For the irregular points ξ_2 , ξ_3 , and ξ_4 from Figure 3.2,

	$a_{i,1}^+$	$a_{i,2}^+$	$a_{i,3}^+$	$a_{i,4}^+$	$a_{i,5}^+$
$i = 1$	$\frac{3}{2\sigma(\sigma+1)(\sigma+2)}$	$\frac{22\sigma-9}{12\sigma}$	$\frac{2-7\sigma}{6(\sigma+1)}$	$\frac{4\sigma-1}{12(\sigma+2)}$	0
$i = 2$	$\frac{-1}{2\sigma(\sigma+1)(\sigma+2)}$	$\frac{4\sigma+3}{12\sigma}$	$\frac{5\sigma+2}{6(\sigma+1)}$	$\frac{-(2\sigma+1)}{12(\sigma+2)}$	0
$i = 3$	$\frac{1}{2\sigma(\sigma+1)(\sigma+2)}$	$\frac{-(2\sigma+3)}{12\sigma}$	$\frac{5\sigma+8}{6(\sigma+1)}$	$\frac{4\sigma+5}{12(\sigma+2)}$	0
	$a_{i,1}^-$	$a_{i,2}^-$	$a_{i,3}^-$	$a_{i,4}^-$	$a_{i,5}^-$
$i = 1$	0	$\frac{25}{12}$	$-\frac{23}{12}$	$\frac{13}{12}$	$-\frac{3}{12}$
$i = 2$	0	$\frac{3}{12}$	$\frac{13}{12}$	$-\frac{5}{12}$	$\frac{1}{12}$

Table 3.3: One-sided non-uniform interpolation constants for inviscid terms at irregular points.

Duan et al. change the stencil for the inviscid terms to a third-order or fourth-order one-sided non-uniform upwind finite-difference stencil. The stencils can include the boundary point at ξ_1 . The inclusion of the boundary point is what makes the stencil non-uniform since the spacing between the boundary point and ξ_2 is $\sigma\Delta\xi$ instead of just $\Delta\xi$. In this work, the derivatives are first cast in conservative form and then the fluxes are interpolated using a non-uniform stencil. The derivative of the inviscid flux in the ξ direction with respect to ξ is computed as

$$\frac{\partial \mathbf{F}_\xi(\xi_i)}{\partial \xi} = \frac{\hat{\mathbf{F}}_\xi^+(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^+(\xi_{i-1/2})}{\Delta\xi} + \frac{\hat{\mathbf{F}}_\xi^-(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^-(\xi_{i-1/2})}{\Delta\xi}. \quad (3.30)$$

The fluxes at the half grid points are computed as

$$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2}) = \sum_{j=1}^5 a_{i,j}^\pm \mathbf{F}_\xi^\pm(\xi_j) \quad (3.31)$$

where the constants are given in Table 3.3. The fluxes at $\xi_{1+1/2}$ are not computed at the half way point between ξ_1 and ξ_2 . They are instead evaluated at $\xi_2 - \Delta\xi/2$. The fluxes $\hat{\mathbf{F}}_\xi^+(\xi_{4+1/2})$ and $\hat{\mathbf{F}}_\xi^-(\xi_{4+1/2})$ are computed using the interior hybrid scheme since there are sufficient uniform points to form the hybrid stencil. The flux $\hat{\mathbf{F}}_\xi^-(\xi_{3+1/2})$ is computed using

i	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	$a_{i,5}$	$a_{i,6}$
1	$\frac{-2(2\sigma+3)(\sigma^2+3\sigma+1)}{\sigma(\sigma+1)(\sigma+2)(\sigma+3)}$	$\frac{(\sigma+1)(\sigma+2)(\sigma+3)}{6\sigma}$	$\frac{-\sigma(\sigma+2)(\sigma+3)}{2(\sigma+1)}$	$\frac{\sigma(\sigma+1)(\sigma+3)}{2(\sigma+2)}$	$\frac{-\sigma(\sigma+1)(\sigma+2)}{6(\sigma+3)}$	0
2	$\frac{-6}{\sigma(\sigma+1)(\sigma+2)(\sigma+3)}$	$\frac{6-11\sigma}{6\sigma}$	$\frac{3\sigma}{(\sigma+1)}$	$\frac{-3\sigma}{2(\sigma+2)}$	$\frac{\sigma}{3(\sigma+3)}$	0
3	$\frac{2}{\sigma(\sigma+1)(\sigma+2)(\sigma+3)}$	$\frac{-2(\sigma+1)}{6\sigma}$	$\frac{1-\sigma}{2(\sigma+1)}$	$\frac{\sigma+1}{(\sigma+2)}$	$\frac{-(\sigma+1)}{6(\sigma+3)}$	0
4	0	$\frac{5}{60}$	$-\frac{40}{60}$	0	$\frac{40}{60}$	$-\frac{5}{60}$

Table 3.4: One-sided non-uniform finite-difference constants for viscous terms at irregular and boundary points.

just WENO since there are enough uniform grid points for the WENO stencil but not enough for the hybrid stencil.

To compute the viscous terms, Duan et al. use a fourth-order one-sided non-uniform finite-difference stencil at ξ_2 and ξ_3 and a fifth-order one-sided non-uniform finite-difference stencil at ξ_4 . In this work, the same stencil is used for ξ_2 and ξ_3 but a uniform fourth-order central finite-difference stencil is used at point ξ_4 . The derivative of the viscous flux in the ξ direction with respect to ξ at all three points can be expressed as

$$\frac{\partial \mathbf{F}_{v,\xi}}{\partial \xi}(\xi_i) = \frac{1}{\Delta \xi} \sum_{j=1}^6 a_{i,j} \mathbf{F}_{v,\xi}(\xi_j) \quad (3.32)$$

where the constants are given in Tables 3.4. The table also includes the constants for computing the derivative at the boundary point ξ_1 . These values will be used in the next section.

3.2.5 Boundary Points

Boundary points are created anywhere a grid line crosses the immersed surface. The boundary conditions for the flow will be imposed at the boundary points. Since the immersed surface is always treated as a wall in this work and the flow is viscous, the no-slip boundary condition are imposed there. The temperature at the wall will depend on the temperature boundary condition for the problem and will be either isothermal or adiabatic in this work.

The pressure at the boundary points will be extrapolated from the flow.

In Duan et al., a fourth-order extrapolation in the direction of the grid line which created the boundary point is used to compute the wall pressure. For adiabatic walls, only the temperature gradient in the direction of the grid line which created the boundary point is set to zero. In the current work, it was found that for two-dimensional simulations this method works for shorter roughness elements but can cause numerical instabilities for taller roughness elements. A least-squares polynomial was used instead.

The pressure at boundary points is computed using a second-degree least-squares polynomial in x , y , and z of the form

$$p(x, y, z) = c_1 + c_2\Delta x + c_3\Delta y + c_4\Delta z + c_5\Delta x\Delta y + c_6\Delta x\Delta z + c_7\Delta y\Delta z + c_8\Delta x^2 + c_9\Delta y^2 + c_{10}\Delta z^2 \quad (3.33)$$

where

$$\Delta x = x - x_b, \quad \Delta y = y - y_b, \quad \Delta z = z - z_b, \quad (3.34)$$

and x_b , y_b , and z_b are the x , y , and z values of the boundary point. For three-dimensional simulations, the constants are computed using 25 regular or irregular points near the boundary point. For two-dimensional simulations, the terms with a Δz are removed and only 15 points are used. The first three points along the grid line which created the boundary point should be included in the set of points. Including these points reduces the likelihood of obtaining a singular matrix when computing the constants. The remaining points are simply the closest points by index to the boundary point. Other boundary points cannot be included. Including them would result in a large implicit system. In addition, points dropped in any direction cannot be included. The least-squares polynomial used for points dropped in one or two directions includes boundary points and therefore must be updated after the boundary points.

A similar method is used to enforce the adiabatic boundary condition. A second-degree polynomial for the temperature is created using the same points as the pressure polynomial.

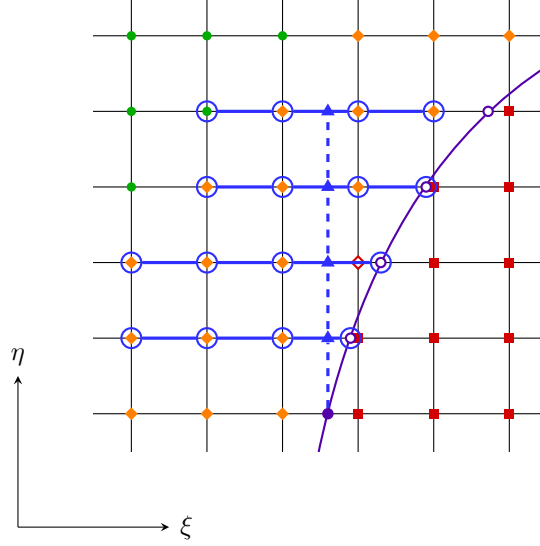


Figure 3.4: Points used to compute a derivative in η direction for a ξ boundary point. Point classification for the ξ direction with $\Theta \approx 1/3$ shown. Immersed boundary —; Regular point ●; Irregular point ◆; Boundary point ○; Dropped point in both directions ■; Dropped point in ξ direction only ◆; Boundary point where derivative is being computed ●; Interpolated point for finite-difference stencil ▲; Stencil used to interpolate points for finite-difference stencil ○—○; Finite-difference stencil used to compute derivative ●—▲.

The polynomial is evaluated at four evenly spaced points in the wall normal direction. These values are then used along with a uniform one-sided fourth-order finite-difference stencil to enforce a zero temperature gradient at the wall.

In addition to imposing the boundary conditions on the flow variables, the shear stresses must be computed at the boundary points. To compute the shear stresses, derivatives are required in all the directions of the computational domain. To compute the required derivatives in the direction of the grid line which created the boundary point, Duan et al. use a fourth-order non-uniform finite-difference stencil. The same stencil is used in this work and is evaluated using Equation 3.32 with the constants coming from Table 3.4. However, Duan et al. do not explain how they compute the derivatives in the other directions of the computational domain. In this work, these derivatives are computed using a combination of interpolation and a uniform one-sided finite-difference stencil. Figure 3.4 shows how the derivative in η direction is computed for a boundary point defined by a ξ grid line in a two-dimensional domain. Four points above the boundary point are first computed by

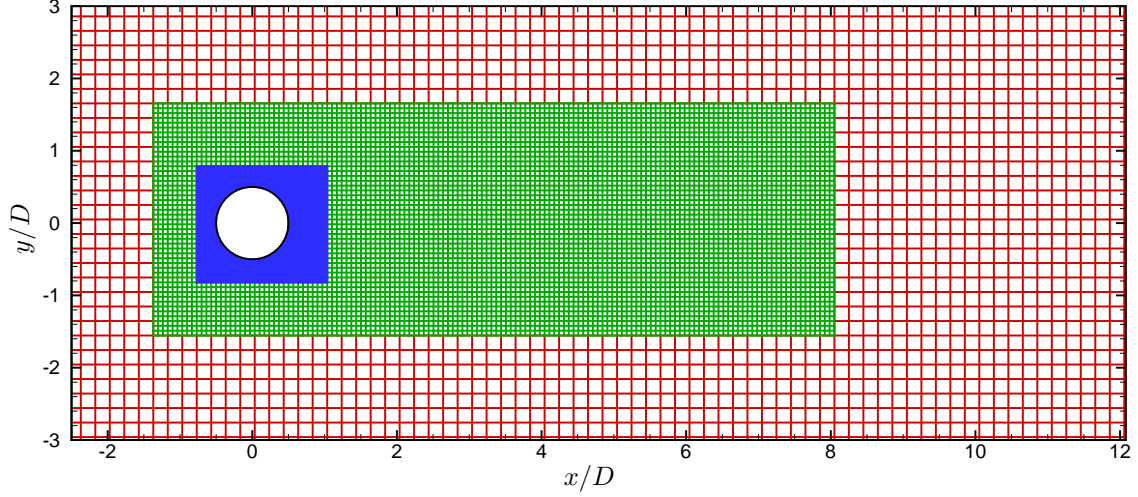


Figure 3.5: Sample multi-zone refinement grid for flow over a cylinder. Points inside the cylinder are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue.

interpolating values in the ξ direction. The interpolation stencils should not include any points dropped in the direction of the grid line on which the boundary point lies (e.g. the ξ grid line in Figure 3.4). The interpolation stencil may include boundary points. These interpolated values are then used in a uniform one-side finite-difference stencil to compute the derivative.

3.3 Multi-Zone Refinement Method

The multi-zone refinement method utilizes overlapping grids to precisely control grid point resolution. The method used in this work comes from Shen et al. [84]. Shen et al. developed a high-order finite-difference WENO version of the adaptive mesh refinement method. Since their method was adaptive, the refinement zones would move with the flow physics. For the problems considered in the current study, the locations needing increased resolution tend to be fixed in time. Therefore, a stationary refinement grid version was used in this work.

Figure 3.5 shows a sample multi-zone refinement grid used to simulate two-dimensional flow over a cylinder. There is a coarse grid placed over the entire shown in red. The figure

only shows the grid near the cylinder. The coarse grid extends much further than what is shown. The first refinement zone is shown in green. This zone increases the resolution by a factor of three. Its purpose is to resolve the cylinder's wake. The second refinement zone is shown in blue. This zone increases the resolution by an additional factor of three. This zone is present to ensure the cut-cell method is properly enforcing the cylinder geometry.

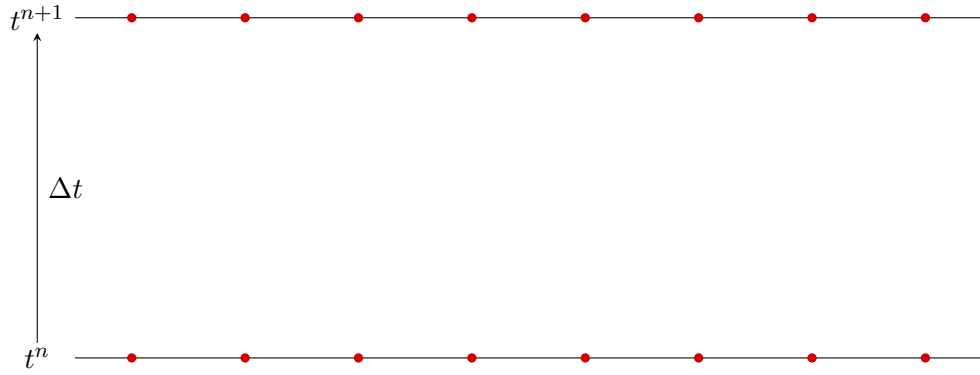
Figure 3.6 depicts the process for generating a one-dimensional domain with a single refinement zone. The method begins by placing a single coarse grid over the entire flow field. Figure 3.6a shows the coarse grid at the t^n and t^{n+1} time steps. Next smaller grids with increased resolution are placed on top of the coarse grid in locations where complex physics are expected. Figure 3.6b shows a refinement grid with a factor of three increase in resolution placed over the coarse grid. Six ghost points are attached to the edge of the refinement grid. The flow variables at the ghost points are computed from the coarse grid using fifth-order interpolation. Most of the coarse grid points which are overlapped by refinement grid points become coarse grid ghost points. The flow variables at these points are simply set equal to the corresponding refinement grid point values. In addition to the spatial refinement, the temporal resolution of the refinement grid is also increased by the same factor. Figure 3.6c shows the additional time steps taken by the refinement grid. The flow variables at the refinement grid ghost points for the additional time steps are computed from the ghost points at steps t^n and t^{n+1} using fourth-order Hermite interpolation. This requires that the coarse grid be advanced in time before advancing the refinement grid since coarse grid values at t^n and t^{n+1} are required to compute the refinement grid ghost points. The exact process for advancing the solution one time step is as follows:

Step 1: Advance coarse grid from t^n to t^{n+1} .

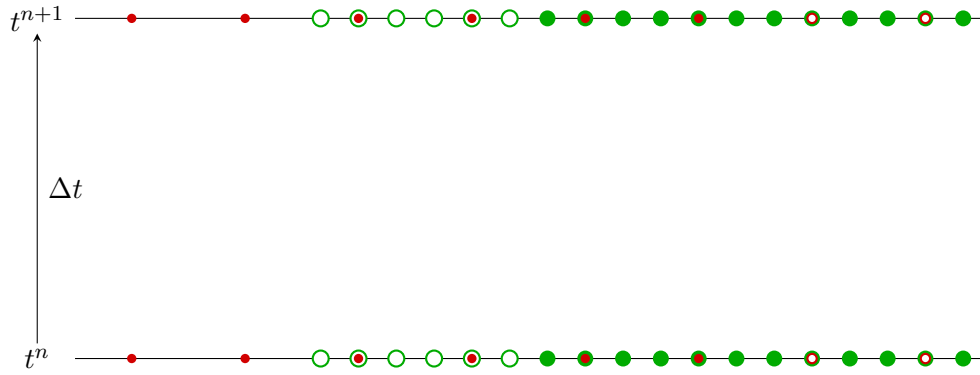
Step 2: Compute $\frac{\partial \mathbf{U}^{n+1}}{\partial t}$ on coarse grid.

Step 3: Compute refinement grid ghost point values for steps t^n and t^{n+1} .

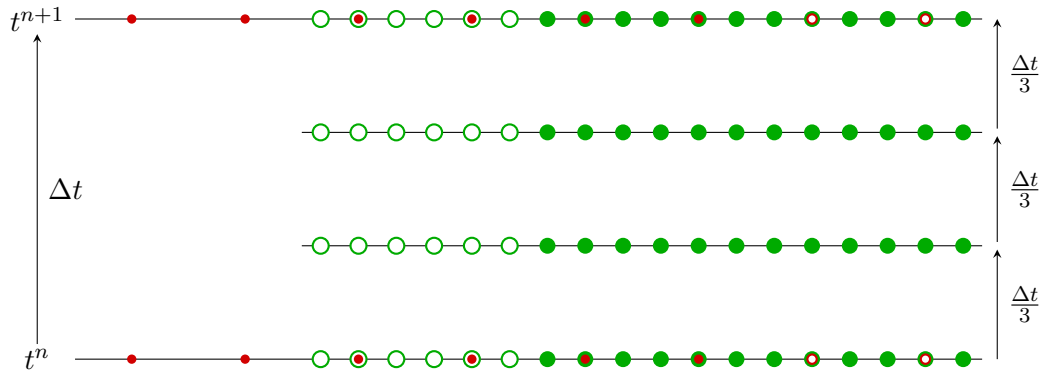
Step 4: Compute refinement grid ghost point values for temporal refinement sub-steps.



(a) Step 1: Place a single coarse grid over entire domain.



(b) Step 2: Add refinement zone where extra resolution is required.



(c) Step 3: Add refinement in time.

Figure 3.6: Multi-zone refinement grid generation and point classification. Coarse grid point \bullet ; coarse grid ghost point \circ ; refinement grid point \bullet ; refinement grid ghost point \circ .

Step 5: Advance refinement grid from t^n to t^{n+1} .

Step 6: Update coarse grid ghost points.

The details for each step are given in the following sections. Note that all of the steps are applied in computational space and that $\mathbf{U} = [\rho, \rho u, \rho v, \rho w, E]^T / J$. The inclusion of the Jacobian in \mathbf{U} will account for the possibility of non-uniform spacing in physical space when performing any interpolation.

3.3.1 Advance Coarse Grid From t^n to t^{n+1}

The first step in advancing the solution one time step is to advance the coarse grid one time step. During this process, no special steps are needed due to the mesh refinement. The solution on the coarse grid can be advanced as if the refinement grids were not present.

3.3.2 Compute $\frac{\partial \mathbf{U}^{n+1}}{\partial t}$ on Coarse Grid

Fourth-order Hermite interpolation will be employed to compute the value of the refinement grid ghost points for the temporal refinement sub-steps. To perform the interpolation, the coarse grid state variable at steps t^n to t^{n+1} (\mathbf{U}^n and \mathbf{U}^{n+1}) and their derivative with respect to time ($\frac{\partial \mathbf{U}^n}{\partial t}$ and $\frac{\partial \mathbf{U}^{n+1}}{\partial t}$) will be needed. At this point, the state variables are known at both time steps since the coarse solution has already been advanced to step t^{n+1} . The derivative with respect to time can be computed from spatial derivatives through the governing equations (Equation 2.10). $\frac{\partial \mathbf{U}^n}{\partial t}$ was computed to advance to step t^{n+1} . The only remaining value which needs to be computed is $\frac{\partial \mathbf{U}^{n+1}}{\partial t}$.

3.3.3 Compute Refinement Grid Ghost Point Values for Steps t^n and t^{n+1}

The next step is to compute the refinement grid ghost point values for steps t^n and t^{n+1} . The values are interpolated from the coarse grid values. Shen et al. used a WENO interpolation to compute the values. In this work, it was found that simple high-order interpolation

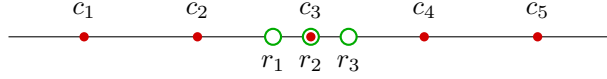


Figure 3.7: Coarse points used to compute refinement grid ghost point values. The stencil is for a refinement grid with a factor of three increase in resolution. Coarse grid point \bullet ; refinement grid ghost point \circ .

i	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	$a_{i,5}$
1	$-\frac{7}{243}$	$\frac{70}{243}$	$\frac{70}{81}$	$-\frac{35}{243}$	$\frac{5}{243}$
2	0	0	1	0	0
3	$\frac{5}{243}$	$-\frac{35}{243}$	$\frac{70}{81}$	$\frac{70}{243}$	$-\frac{7}{243}$

Table 3.5: Constants for refinement grid ghost point value interpolation. Constants are for a refinement grid with a factor of three increase in resolution.

was sufficient. Figure 3.7 shows the coarse points used to compute the values for a set of refinement ghost points on a one-dimensional domain. The figure depicts a refinement grid with a factor of three increase in resolution. The value at a refinement ghost point is computed from the closest five coarse grid points. This results in fifth-order interpolation.

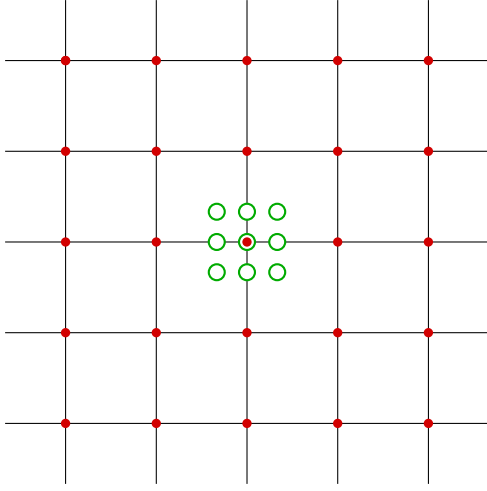
The state variable at point r_i ($\mathbf{U}(r_i)$) is interpolated from the state variable at the coarse points ($\mathbf{U}(c_j)$) from

$$\mathbf{U}(r_i) = \sum_{j=1}^5 a_{i,j} \mathbf{U}(c_j) \quad (3.35)$$

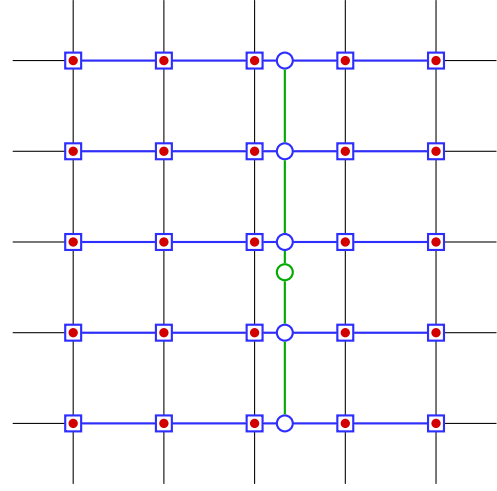
where the constants $a_{i,j}$ can be found in Table 3.5. The derivative of the state variable with respect to time is interpolated using the same constants.

$$\frac{\partial \mathbf{U}}{\partial t}(r_i) = \sum_{j=1}^5 a_{i,j} \frac{\partial \mathbf{U}}{\partial t}(c_j) \quad (3.36)$$

For two-dimensional and three-dimensional problems, the interpolation stencil can easily be extended. Figure 3.8a shows a group of refinement grid ghost points and the 25 closest coarse grid point from which their values are interpolated. Figure 3.8b shows the interpolation stencil for one of the ghost points. First, interpolation is used along the horizontal grid



(a) Group of refinement grid ghost points and the coarse grid points used to compute their values.



(b) Stencils used to compute values at sample refinement grid ghost point.

Figure 3.8: Coarse points and sample stencil used to compute two-dimensional refinement grid ghost point values. Coarse grid point \bullet ; refinement grid ghost point \circ ; temporary refinement grid ghost points \square ; stencil for temporary refinement grid ghost points $\square-\circ-\square$; stencil for refinement grid ghost points $\circ-\circ-\circ$.

lines to obtain values at the temporary refinement grid ghost points. These values are only temporary in the sense that they are not the desired final ghost point. In an actual problem, these points will likely correspond to actual refinement grid ghost points. The interpolation is the same as depicted in Figure 3.7 and Equation 3.35. Next, interpolation is performed in the vertical direction to obtain the desired ghost point values. The interpolation constants found in Table 3.5 can also be used here.

3.3.4 Compute Refinement Grid Ghost Point Values for Temporal Refinement Sub-Steps

In addition to the spatial refinement, the refinement grid is also refined in time as shown in Figure 3.6c. The values for the refinement grid ghost points at the temporal sub-steps are computed from the ghost point values at steps t^n and t^{n+1} using fourth-order Hermite interpolation. Let $\mathbf{H}(t)$ be the Hermite interpolation of $\mathbf{U}(t)$. The expression for $\mathbf{H}(t)$

between t^n and t^{n+1} is

$$\mathbf{H}(t) = b_3 \left(\frac{t - t^n}{\Delta t} \right)^3 + b_2 \left(\frac{t - t^n}{\Delta t} \right)^2 + b_1 \left(\frac{t - t^n}{\Delta t} \right) + b_0. \quad (3.37)$$

The unknown constants (b_0 , b_1 , b_2 , and b_3) can be computed by evaluating the expression and its derivative at t^n and t^{n+1} . Doing so generates the following system of equations:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} \mathbf{U}^n \\ \frac{\partial \mathbf{U}^n}{\partial t} \\ \mathbf{U}^{n+1} \\ \frac{\partial \mathbf{U}^{n+1}}{\partial t} \end{bmatrix}. \quad (3.38)$$

Solving the system produces the following values for the constants:

$$\begin{aligned} b_0 &= \mathbf{U}^n, \\ b_1 &= \frac{\partial \mathbf{U}^n}{\partial t}, \\ b_2 &= -3\mathbf{U}^n - 2\frac{\partial \mathbf{U}^n}{\partial t} + 3\mathbf{U}^{n+1} - \frac{\partial \mathbf{U}^{n+1}}{\partial t}, \\ b_3 &= 2\mathbf{U}^n + \frac{\partial \mathbf{U}^n}{\partial t} - 2\mathbf{U}^{n+1} + \frac{\partial \mathbf{U}^{n+1}}{\partial t}. \end{aligned} \quad (3.39)$$

3.3.5 Advance Refinement Grid From t^n to t^{n+1}

With the refinement grid ghost point values at the temporal refinement steps known, the refinement grid solution can be advanced in time until it reaches t^{n+1} . The refinement grid will need to take a number of steps equal to the spatial refinement ratio. The solver used to advance the code uses a Runge-Kutta method for time advancement. Therefore, ghost point values will be needed at the Runge-Kutta sub-steps. Carpenter et al. [95] showed that to maintain the order of the Runge-Kutta method, unsteady boundary values for sub-steps should be computed using the Runge-Kutta method and not simply an evaluation of the unsteady function at the sub-step solution time. If t^* is the time at the start of the Runge-

Kutta step (t^n , $t^n + \Delta t/3$, or $t^n + 2\Delta t/3$ for a refinement of three), then the ghost point values during the Runge-Kutta sub-steps are

$$\begin{aligned}\mathbf{U}(t^*) &= \mathbf{H}(t^*), \\ \mathbf{U}^{(1)} &= \mathbf{H}(t^*) + \delta t \frac{\partial \mathbf{H}}{\partial t}(t^*), \\ \mathbf{U}^{(2)} &= \mathbf{H}(t^*) + \frac{\delta t}{2} \frac{\partial \mathbf{H}}{\partial t}(t^*) + \frac{\delta t^2}{4} \frac{\partial^2 \mathbf{H}}{\partial t^2}(t^*),\end{aligned}\tag{3.40}$$

where δt is the time step for the refinement grid and $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are the values at the first and second Runge-Kutta sub-step, respectively.

3.3.6 Update Coarse Grid Ghost Points

Once the refinement grid has been advanced to step t^{n+1} , the coarse grid ghost points can be updated. Any coarse grid point which has a corresponding point on a refinement grid is a coarse grid ghost point. The values at the ghost points are copied from the refinement grid values at t^{n+1} . The only exception is the first two coarse points next to a refinement grid edge. This is shown in Figure 3.6b. The first two coarse points to the right of the refinement grid ghost points are normal coarse points. Although these two points are covered by the refinement grid, they will be advanced in time independently of the refinement grid. This is done for numerical stability. These two points are used during the interpolation to compute the refinement grid ghost points. If they were updated with the coarse grid points, the interpolation would essentially be one-sided and would become unstable.

Updating the coarse grid ghost points is the final step in the multi-zone refinement method. The steps can now be repeated until the desired final time is achieved. For problems with more than one nested refinement grid, the method outline above can be applied recursively to each grid.

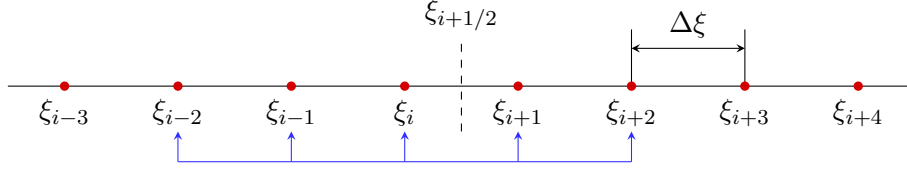


Figure 3.9: Sample uniformly space grid. Typical fifth-order stencil used to compute $\hat{f}(\xi_{i+1/2})$ shown in blue.

3.3.7 Non-Conservation of Multi-Zone Refinement Method

A downside of the multi-zone refinement method used in this work is the fact that the method does not enforce conservation. This is due to the use of interpolation at the refinement grid ghost points. This deficiency is also addressed by the original developers of the method Shen et al. [84]. Before implementing the current version of the multi-zone refinement method, an attempt to formulate a high-order conservative version was made. The attempt proved to be unfruitful but will still be presented here for reference.

3.3.7.1 Conservative Derivative on Uniform Grid

Before discussing the conservative multi-zone refinement method, the process for developing a conservative scheme on a uniform grid will be covered. Consider the uniformly spaced computational domain shown in Figure 3.9. The conservative derivative of a quantity $f(\xi)$ with respect to ξ at point ξ_i is

$$\frac{\partial f}{\partial \xi}(\xi_i) = \frac{h(\xi_{i+1/2}) - h(\xi_{i-1/2})}{\Delta \xi} \quad (3.41)$$

where $h(\xi)$ is a function which makes the derivative exact instead of an approximation. This function is called the numerical flux function in the literature. To formulate a conservative method, the value for $h(\xi_{i+1/2})$ used while computing the derivative at points ξ_i and ξ_{i+1} must be the equal. The function $h(\xi)$ is defined by

$$f(\xi) = \frac{1}{\Delta \xi} \int_{\xi - \Delta \xi/2}^{\xi + \Delta \xi/2} h(\hat{\xi}) d\hat{\xi}. \quad (3.42)$$

High-order conservative numerical methods can be formed by using high-order approximations of $h(\xi)$. For a fifth-order scheme, a fourth-degree polynomial is used.

$$h(\xi) \approx \hat{f}(\xi) = a_0 + a_1\xi + a_2\xi^2 + a_3\xi^3 + a_4\xi^4 \quad (3.43)$$

Substituting Equation 3.43 into Equation 3.42 and integrating gives

$$f(\xi) = a_0 + a_1\xi + a_2\left(\xi^2 + \frac{\Delta\xi}{12}\right) + a_3\left(\xi^3 + \frac{\xi\Delta\xi^2}{4}\right) + a_4\left(\xi^4 + \frac{\xi^2\Delta\xi^2}{2} + \frac{\Delta\xi^4}{80}\right). \quad (3.44)$$

The values for the coefficients a_j can now be computed by evaluating Equation 3.44 at the points used in the numerical method's stencil. Figure 3.9 shows the typical stencil used for an upwind fifth-order conservative scheme for the case where the eigenvalues of the Jacobian of $f(\xi)$ are positive. Using this stencil, gives $\hat{f}(\xi_{i+1/2})$ as

$$\hat{f}(\xi_{i+1/2}) = \frac{1}{60}\left(2f(\xi_{i-2}) - 13f(\xi_{i-1}) + 47f(\xi_i) + 27f(\xi_{i+1}) - 3f(\xi_{i+2})\right). \quad (3.45)$$

The derivative of a quantity $f(\xi)$ with respect to ξ at point ξ_i is now

$$\frac{\partial f}{\partial \xi}(\xi_i) = \frac{\hat{f}(\xi_{i+1/2}) - \hat{f}(\xi_{i-1/2})}{\Delta\xi} - \frac{\Delta\xi^5}{60} \frac{\partial^6 f}{\partial \xi^6}(\xi_i) + O(\Delta\xi^6). \quad (3.46)$$

Using $\hat{f}(\xi)$ instead of $h(\xi)$ will result in a fifth-order approximation of the derivative.

3.3.7.2 Conservative Multi-Zone Refinement Method

To derive a conservative version of the multi-zone refinement method, consider the grid shown in Figure 3.10. The figure has the coarse grid on the left and a refinement zone with three times the resolution on the right. The interface between the coarse grid and the refinement zone is located at $\xi_{i+3/2}$. To create a conservative multi-zone refinement method, the value for $\hat{f}(\xi_{i+3/2})$ used to compute the derivative at the coarse grid point ξ_i and the refinement point ξ_{i+2} must be the equal. This can be enforced, but will result in a first-order derivative

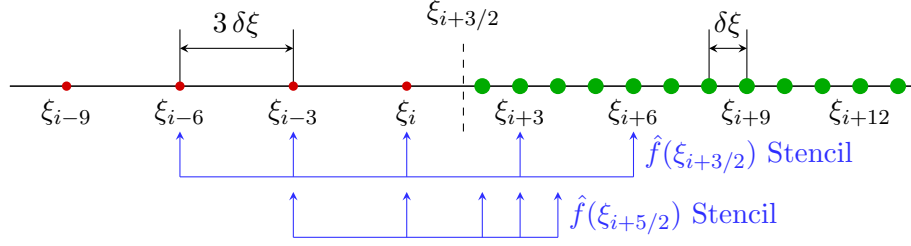


Figure 3.10: Sample uniformly spaced grid with refinement zone. Coarse grid point •; refinement grid point •.

near the interface. This is due to the numerical flux function $h(\xi)$ being a function of the grid spacing as shown in Equation 3.42.

The two stencils shown in Figure 3.10 will be used as an example. The stencil for $\hat{f}(\xi_{i+3/2})$ is the same one used in Section 3.3.7.1. $\hat{f}(\xi_{i+3/2})$ can be computed with all the same equations after substituting $3\delta\xi$ for $\Delta\xi$. This will give $\hat{f}(\xi_{i+3/2})$ as

$$\hat{f}(\xi_{i+3/2}) = \frac{1}{60} \left(2f(\xi_{i-6}) - 13f(\xi_{i-3}) + 47f(\xi_i) + 27f(\xi_{i+3}) - 3f(\xi_{i+6}) \right) \quad (3.47)$$

and the derivative at point ξ_i will be

$$\frac{\partial f}{\partial \xi}(\xi_i) = \frac{\hat{f}(\xi_{i+3/2}) - \hat{f}(\xi_{i-3/2})}{3\delta\xi} - \frac{81\delta\xi^5}{20} \frac{\partial^6 f}{\partial \xi^6}(\xi_i) + O(\delta\xi^6). \quad (3.48)$$

Using the stencil for $\hat{f}(\xi_{i+3/2})$ shown in the figure, will ensure that the derivative at point ξ_i is still a fifth-order approximation.

The derivative at point ξ_{i+2} will be computed as

$$\frac{\partial f}{\partial \xi}(\xi_{i+2}) = \frac{h(\xi_{i+5/2}) - h(\xi_{i+3/2})}{\delta\xi}. \quad (3.49)$$

To make the derivative conservative, $h(\xi_{i+3/2})$ must be approximated by the value of $\hat{f}(\xi_{i+3/2})$ given in Equation 3.47. Since the point ξ_{i+2} has a different grid point spacing than ξ_i , a different approximation for $h(\xi_{i+5/2})$ must be derived. The function $h(\xi)$ for the refinement

zone is defined by

$$f(\xi) = \frac{1}{\delta\xi} \int_{\xi-\delta\xi/2}^{\xi+\delta\xi/2} h(\hat{\xi}) d\hat{\xi}. \quad (3.50)$$

The same fourth-degree polynomial as Equation 3.43 is used to approximate $h(\xi)$. Substituting this approximation for $h(\xi)$ into Equation 3.50 and integrating gives

$$f(\xi) = a_0 + a_1\xi + a_2 \left(\xi^2 + \frac{\delta\xi}{12} \right) + a_3 \left(\xi^3 + \frac{\xi\delta\xi^2}{4} \right) + a_4 \left(\xi^4 + \frac{\xi^2\delta\xi^2}{2} + \frac{\delta\xi^4}{80} \right). \quad (3.51)$$

Using the grid points in the stencil shown in Figure 3.10 to compute the coefficients a_j gives $\hat{f}(\xi_{i+5/2})$ as

$$\hat{f}(\xi_{i+5/2}) = \frac{1}{6300} \left(13f(\xi_{i-3}) - 245f(\xi_i) + 3297f(\xi_{i+2}) + 3745f(\xi_{i+3}) - 510f(\xi_{i+4}) \right). \quad (3.52)$$

The derivative at point ξ_{i+2} is now

$$\frac{\partial f}{\partial \xi}(\xi_{i+2}) = \frac{\hat{f}(\xi_{i+5/2}) - \hat{f}(\xi_{i+3/2})}{\delta\xi} + \frac{\delta\xi}{3} \frac{\partial^2 f}{\partial \xi^2}(\xi_{i+2}) + O(\Delta\xi^2) \quad (3.53)$$

which is only a first-order approximation of the derivative. In order to obtain a high-order conservative scheme, the function used to approximate $h(\xi)$ must be the same at both interfaces. However, since $h(\xi)$ is a function of the grid point spacing and the spacing will always be different between the coarse grid and the refinement zone, the same approximation cannot be used. This will always result in a first-order approximation of the derivative at either the first refinement grid point or the coarse grid point next the refinement grid edge.

CHAPTER 4

Order of Convergence

The order of convergence was computed to ensure that the target high-order of convergence was achieved. Two test cases were used to find the order of convergence. The simulations tested the cut-cell and mesh refinement methods individually.

4.1 Cut-Cell Method

The order of convergence for the cut-cell portion of the code was tested using circular Couette flow. The problem consists of two concentric cylinders with a viscous fluid between them. For the test case, the gap between the cylinders is twice the inner cylinder radius (R). The inner cylinder rotated with a Mach number of one while the outer cylinder was stationary. Both walls were isothermal. The inner cylinder was set to a temperature of 300 K and the outer cylinder was set to 400 K. The Reynolds number based on the inner cylinder radius is 500. To prevent the pressure from continuously changing during the simulation, the pressure at the inner wall was set to 200 Pa. The simulation was performed assuming a constant viscosity. This permitted the comparison to an analytical exact solution.

The simulation was performed on a uniform Cartesian grid measuring $6.5R \times 6.5R$. Both cylinders were imposed by the cut-cell method. Figure 4.1 shows a sample of the grid using 50×50 grid points. The simulation was carried out on four grids: 100×100 , 200×200 , 400×400 , and 800×800 . Figure 4.2 shows the Mach contours for the solution computed on the 200×200 grid. Despite being simulated on a Cartesian grid, the contours are circular as the exact solution predicts. Figure 4.3 shows the pressure, temperature, and Mach number

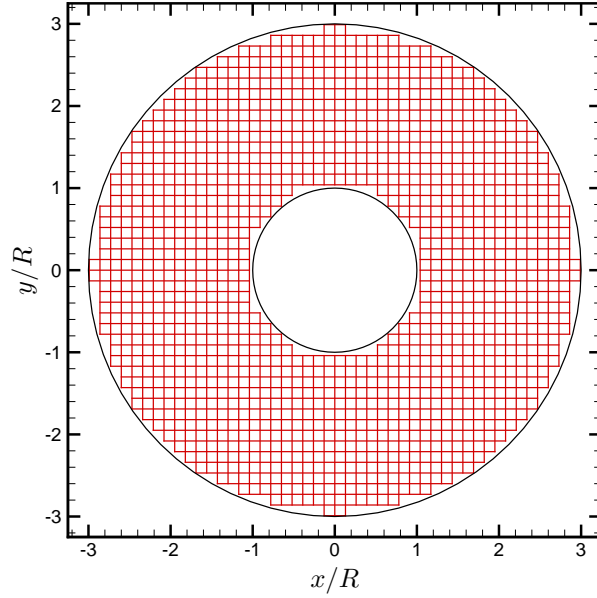


Figure 4.1: Sample grid for circular Couette flow with 50×50 points. Points inside the cylinders are not shown.

profiles on the 200×200 grid. All the profiles match the exact solution.

Figure 4.4 shows the L^2 , L^1 , and L^∞ errors for the conservative variables from each of the grid resolutions. The number next to each line segment is the slope of that segment which is equal to the order of convergence. In some of the plots, the L^2 error is not visible because it is approximately equal to the L^1 error for that variable. The cut-cell method uses third-order differencing near the immersed surface. When combined with the fifth-order interior scheme, the expected global order of convergence is four. The L^2 and L^1 errors show that the order of convergence is near the desired value of four. The L^∞ error is showing a value of three since it is being computed from the maximum error which is located at the third-order cut-cell stencils.

4.2 Multi-Zone Refinement Method

The order of convergence for the multi-zone refinement portion of the code was tested by simulating the propagation of a low-amplitude density wave. The wave was sinusoidal in

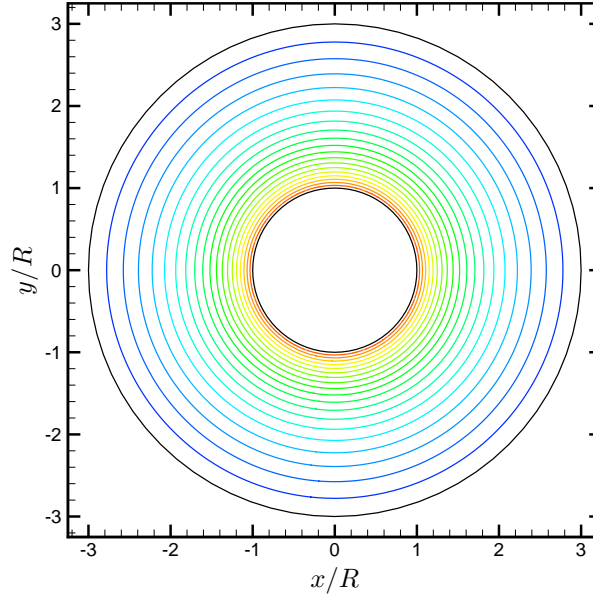
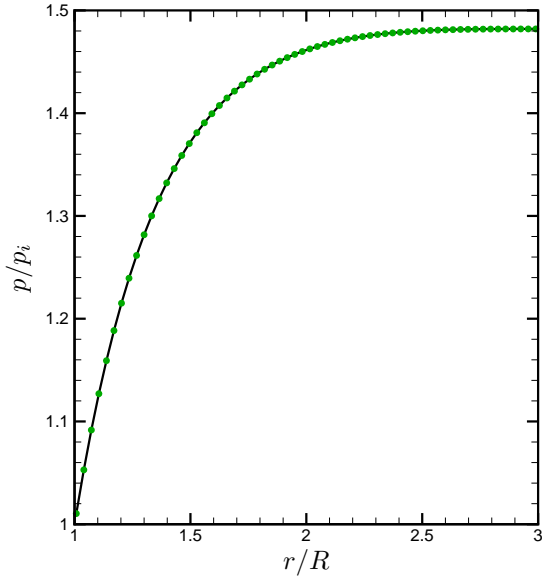


Figure 4.2: Mach contour for circular Couette flow. Solution computed on 200×200 grid. 21 evenly spaced levels from 0.0 to 1.0.

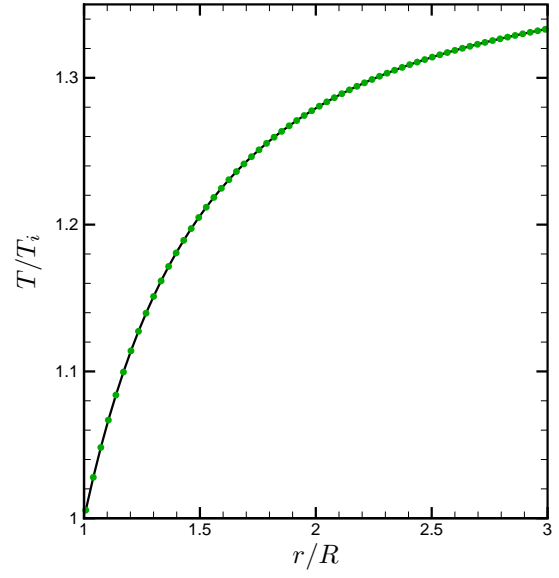
shape with an amplitude of 10% of the mean density. The wave propagates at an angle of approximately 26° with the x axis. The viscous terms were turned off for this simulation.

Figure 4.5 shows the density contour for the simulation. The figure also shows the locations of the first and second refinement zones. The first zone increases the resolution by a factor of three compared to the coarse grid and the second increases the resolution by another factor of three. Periodic boundary conditions are used along the coarse grid edges.

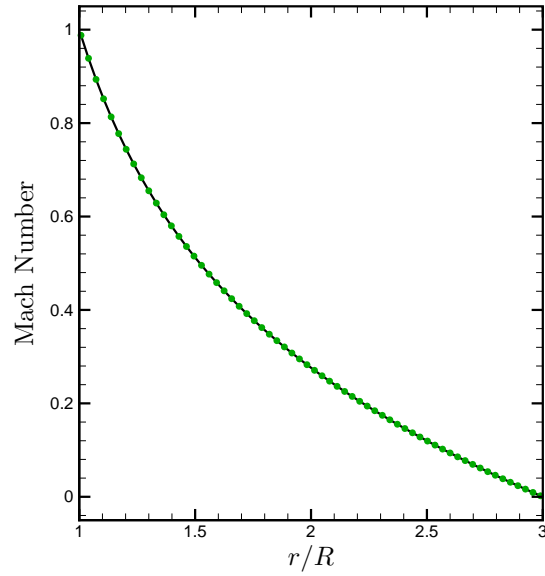
The simulation was performed on a uniform Cartesian grid. In order to compute the spatial and temporal order of convergence, two sets of simulations were performed. In the first, the CFL number was fixed at 0.1 while the number of grid points was changed. Four simulations were carried out with the coarse grid having 100×50 , 200×100 , 400×200 , and 800×400 points. The results from this set of simulations were used to compute the spatial order of convergence. Figure 4.6 shows the L^2 , L^1 , and L^∞ errors for density from each of the grid resolutions. The error computed on the coarse grid and each refinement zone are shown separately. Since the cut-cell method is not used in this case and the domain is periodic, the optimal order of the interior scheme should be obtained. Since a fifth-order hybrid scheme



(a) Pressure profile. Non-dimensionalized by pressure at inner wall.

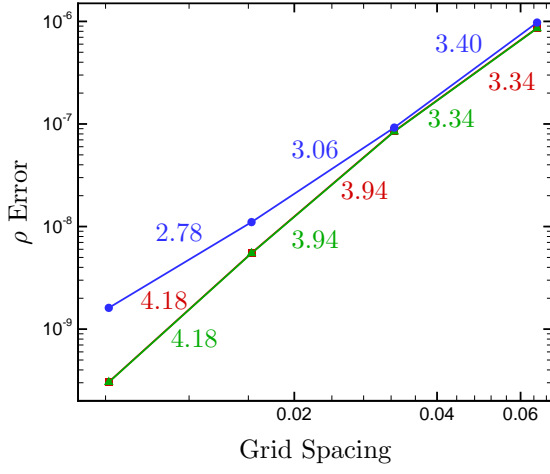


(b) Temperature profile. Non-dimensionalized by temperature at inner wall.

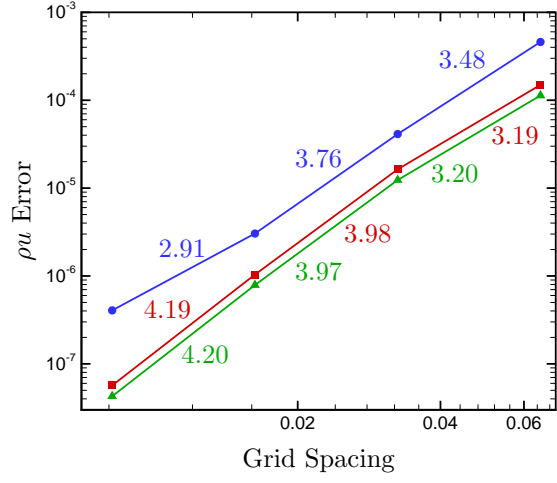


(c) Mach profile.

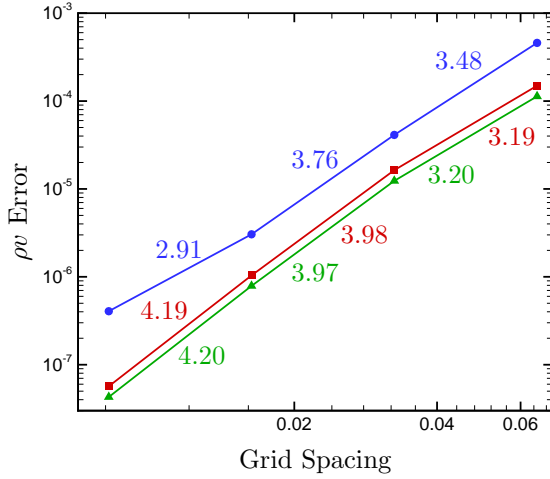
Figure 4.3: Pressure, temperature, and Mach number profiles for circular Couette flow. Solution computed on 200×200 grid. Computed Solution \bullet ; Exact Solution — .



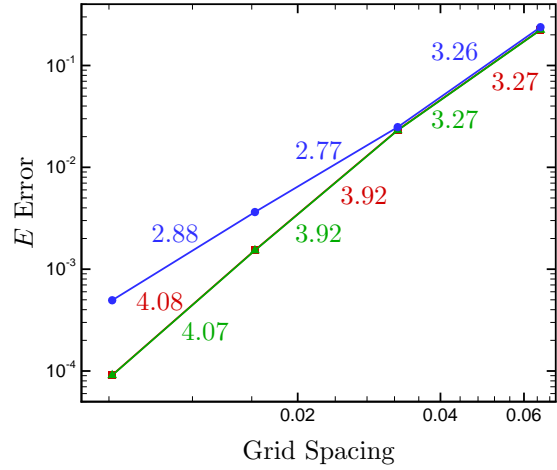
(a) Error in density.



(b) Error in x -direction momentum.



(c) Error in y -direction momentum.



(d) Error in energy.

Figure 4.4: Error in the conservative variables versus grid spacing for circular Couette flow. The number next to each line segment is the order of convergence computed using the line segment end points. L^∞ error —•—; L^1 error —▲—; L^2 error —■—.

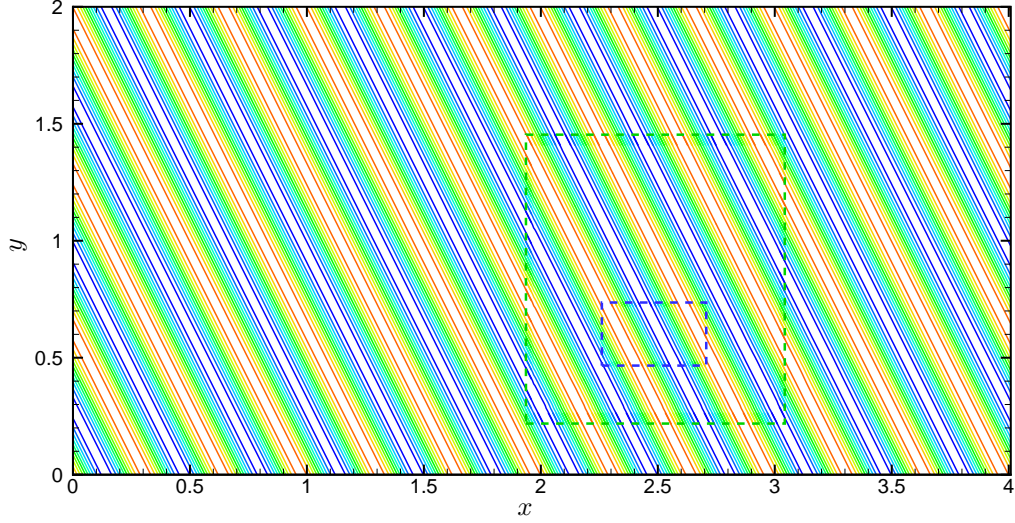
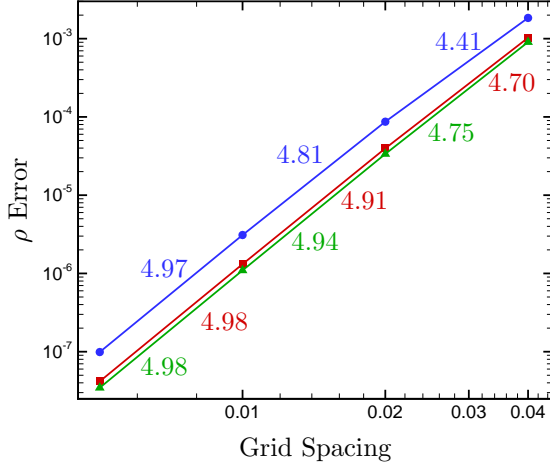


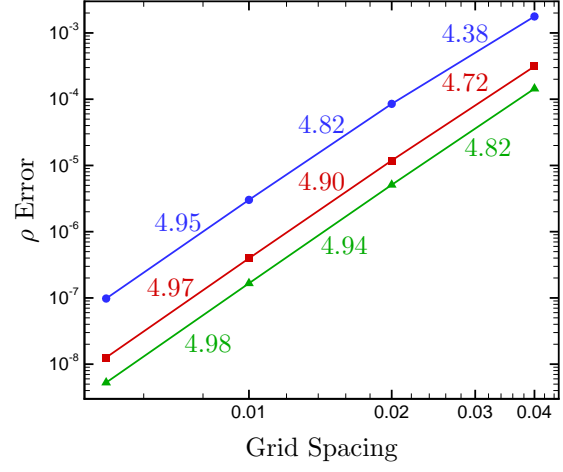
Figure 4.5: Density contour for propagating density wave problem. First refinement zone edge ---; Second refinement zone edge ----.

is being used, the expected order of convergence is five. As all the errors show, the desired value of five is obtained.

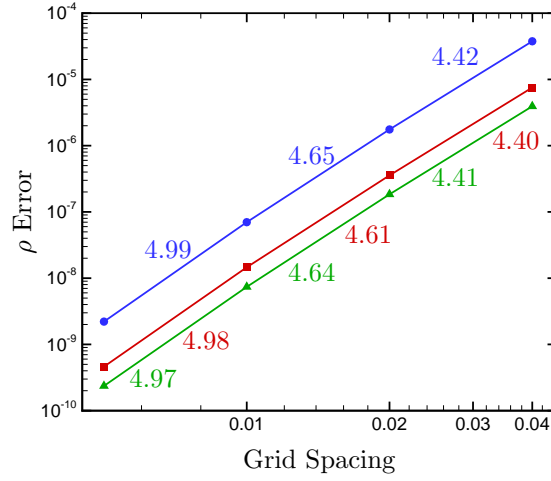
In the second set of simulations, the grid was kept constant while the CFL number was changed. Four simulations were performed with a CFL number of 0.9, 0.81, 0.729, and 0.6561. The results from this set of simulations were used to compute the temporal order of convergence. Figure 4.7 shows the L^2 , L^1 , and L^∞ errors for density from each of the CFL values. The error computed on the coarse grid and each refinement zone are shown separately. Since a third-order Runge-Kutta scheme is used to advance the simulation in time and the multi-zone refinement uses a fourth-order Hermite interpolation in time, the expected temporal order of convergence is three. All three errors show the expected order of convergence.



(a) Error on coarse grid.

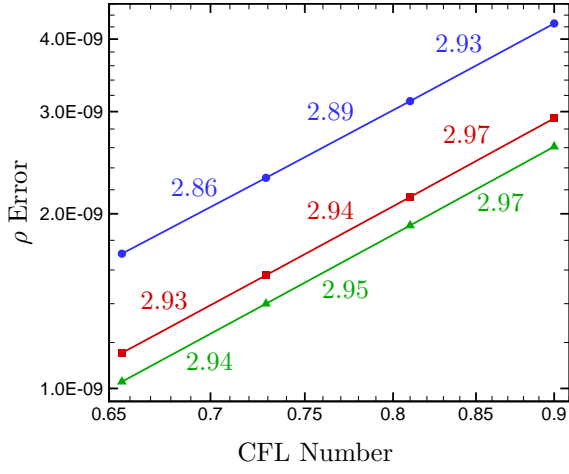


(b) Error on first refinement zone.

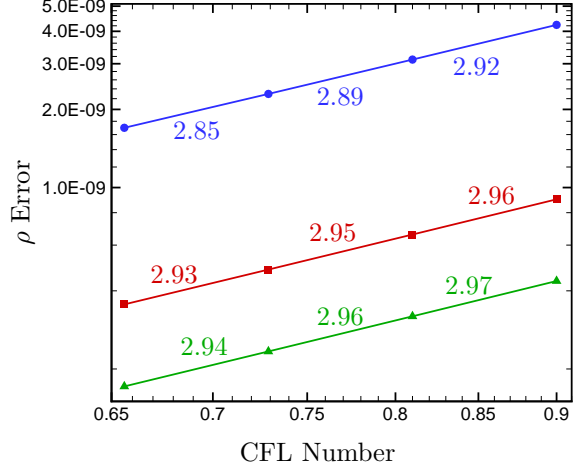


(c) Error on second refinement zone.

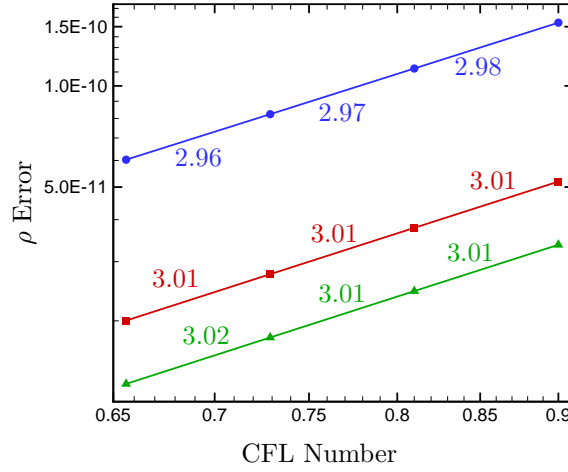
Figure 4.6: Error in density for propagating density wave problem with different grid spacing. The number next to each line segment is the order of convergence computed using the line segment end points. L^∞ error —•—; L^1 error —▲—; L^2 error —■—.



(a) Error on coarse grid.



(b) Error on first refinement zone.



(c) Error on second refinement zone.

Figure 4.7: Error in density for propagating density wave problem with different time step size. The number next to each line segment is the order of convergence computed using the line segment end points. L^∞ error —●—; L^1 error —▲—; L^2 error —■—.

CHAPTER 5

Cut-Cell and Immersed Boundary Method Comparison

Two Cartesian grid methods were explored in this work: the cut-cell method and the immersed boundary method. Both methods are capable of simulating flows over arbitrary geometries with a simple Cartesian grids. Since only one method was needed, a two-dimensional test case was used to evaluate and compare the cut-cell method and the immersed boundary method. The test case is Mach 4.8 boundary-layer flow over an isolated roughness element constructed from hyperbolic tangents. This simulation was also performed by Marxen and Iaccarino [96]. The use of a smooth roughness geometry allows this simulation to also be performed on a body-fitted grid. The results from the body-fitted grid matched very well with results from Marxen and Iaccarino and is assumed to be correct. The body-fitted results are used as a basis for comparison for the cut-cell and immersed boundary methods.

5.1 Immersed Boundary Method

The immersed boundary method implemented here comes from Marxen and Iaccarino [96] and von Terzi et al. [97]. It was selected because it is easy to implement and Marxen and Iaccarino have shown that the method works for supersonic compressible flow. The immersed boundary method mimics the immersed surface through a forcing term added to the right hand side of the governing equations. In this work, the method has only been implemented for two-dimensional simulations. The governing equations with the forcing term become

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j}{\partial \xi_j} = \frac{\partial \mathbf{F}_{v,j}}{\partial \xi_j} + \mathbf{B}, \quad j = 1, 2 \quad (5.1)$$

where \mathbf{B} is the forcing term. Equation 5.1 is solved both inside and outside the immersed body. The forcing term takes the form

$$\mathbf{B} = \begin{pmatrix} 0 \\ B_x \\ B_y \\ 0 \end{pmatrix} \quad (5.2)$$

where B_x and B_y enforce $u = 0$ and $v = 0$ on the immersed surface, respectively. Only the equations for B_x will be covered here. The expressions for B_y can be found by changing u to v in the following equations.

The forcing term B_x is a line integral of the product of a time-dependent body force, b_x , and an approximation to the delta function, d , along the immersed surface S :

$$B_x = \rho \oint_S b_x(\mathbf{x}_s) d(\mathbf{x} - \mathbf{x}_s) dS. \quad (5.3)$$

\mathbf{x} is the location where the force is being computed and \mathbf{x}_s is a location on the immersed surface. The line integral is evaluated using the trapezoidal rule. The approximation to the delta function is

$$d(\mathbf{x} - \mathbf{x}_s) = \exp \left\{ -\frac{(x - x_s)^2}{\sigma_x} - \frac{(y - y_s)^2}{\sigma_y} \right\} \quad (5.4)$$

where σ_x and σ_y are problem dependent constants. The body force is given by

$$b_x = \alpha \int_0^t u(\mathbf{x}_s, t') dt' + \beta u(\mathbf{x}_s, t) \quad (5.5)$$

where $u(\mathbf{x}_s, t)$ is the u velocity component evaluated at \mathbf{x}_s . α and β are additional problem dependent constants. Since the velocity is not solved for on the immersed surface, bilinear interpolation from the four neighboring grid points is used to compute $u(\mathbf{x}_s, t)$. The integral in the equation for b_x is computed as a Riemann sum.

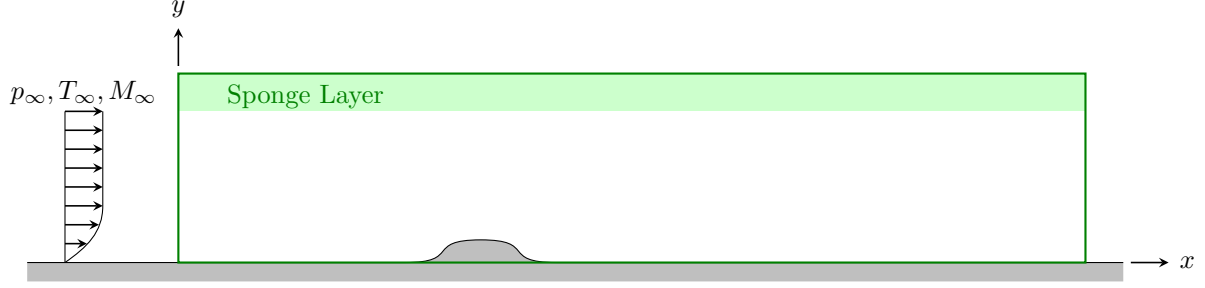


Figure 5.1: Diagram for boundary-layer flow over two-dimensional hyperbolic tangent roughness. Numerical domain edge shown as —.

5.2 Simulation Setup

A diagram of the test case is shown in Figure 5.1 where flow is from left to right. The problem consists of boundary-layer flow over a flat plate. The x axis is aligned with the flow direction and the y axis is normal to the plate. An isolated roughness element constructed from hyperbolic tangent curves is located on the flat plate. The roughness height to undisturbed boundary-layer thickness ratio (k/δ) is 0.547. The equation for the roughness geometry is

$$h(x) = \frac{k}{2} \left[\tanh \left(S_r (x + L_r) \right) - \tanh \left(S_r (x - L_r) \right) \right] \quad (5.6)$$

where $S_r = 20$ and $L_r = 0.2$. The freestream conditions for the flow are summarized in Table 5.1 where $Re_\infty = \rho_\infty u_\infty L_{ref} / \mu_\infty$. The constants used in the immersed boundary method are shown in Table 5.2.

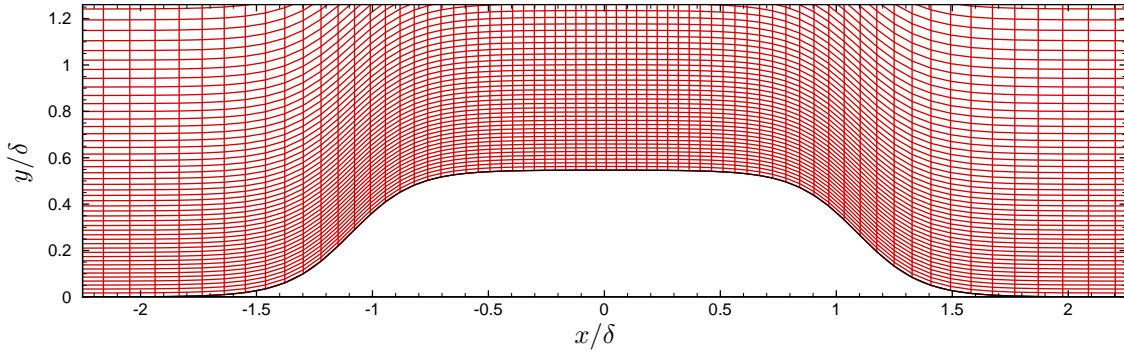
The leading edge of the flat plate is not included in the simulation. Instead, a compressible similarity solution is imposed at the inlet of the numerical domain and held fixed with time. The outlet of the domain is mostly supersonic so the conservative variables are extrapolated from the interior of the domain. At the top of the domain, the v velocity component and temperature are set to their freestream values while the derivative of the vertical velocity and pressure with respect to y was set to zero. In addition, a sponge layer [98] is used at the top of the domain to prevent the shock created by the roughness from reflecting back into the domain. The flat plate and the roughness are no-slip adiabatic walls.

Freestream Mach number	M_∞	4.8
Freestream velocity	u_∞	720 m/s
Freestream pressure	p_∞	239.426 Pa
Freestream temperature	T_∞	55.4 K
Freestream density	ρ_∞	1.51×10^{-2} kg/m ³
Freestream Reynolds number	Re_∞	10^5
Reference length	L_{ref}	3.365×10^{-2} m
Roughness height	k	$0.1 L_{ref}$
Ratio of roughness height to boundary-layer thickness	k/δ	0.547

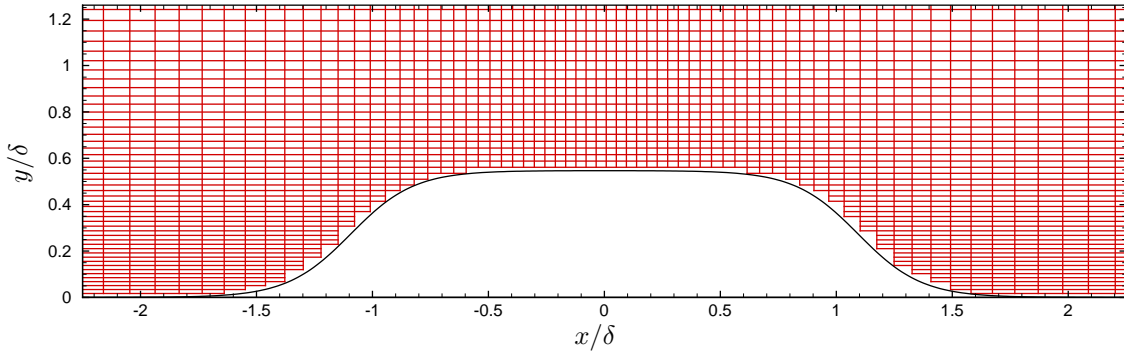
Table 5.1: Flow conditions for two-dimensional hyperbolic tangent simulation.

α	β	σ_x	σ_y
$-10^5 \frac{c_\infty^2}{L_{ref}^3}$	$-5 \times 10^5 \frac{c_\infty}{L_{ref}^2}$	$3.2 \times 10^{-4} L_{ref}$	$5 \times 10^{-4} L_{ref}$

Table 5.2: Constants for immersed boundary method.



(a) Body-fitted grid.



(b) Cartesian grid used for cut-cell method. Points within the roughness element were removed.

Figure 5.2: Grids in vicinity of the two-dimensional hyperbolic tangent roughness element. Every fourth point in both directions shown.

The grids in the vicinity of the roughness element for the body-fitted and cut-cell method are shown in Figure 5.2. The grids use 800 points in the x direction and 400 points in the y direction. The grids were generated by the following equations:

$$x = (x_c - x_{in}) \left(1 + \frac{\sinh(a_1)}{\sinh(a_2)} \right) + x_{in}, \quad (5.7)$$

$$y = (H - B(x)) \frac{\sinh(S_\eta \eta)}{\sinh(S_\eta)} + B(x). \quad (5.8)$$

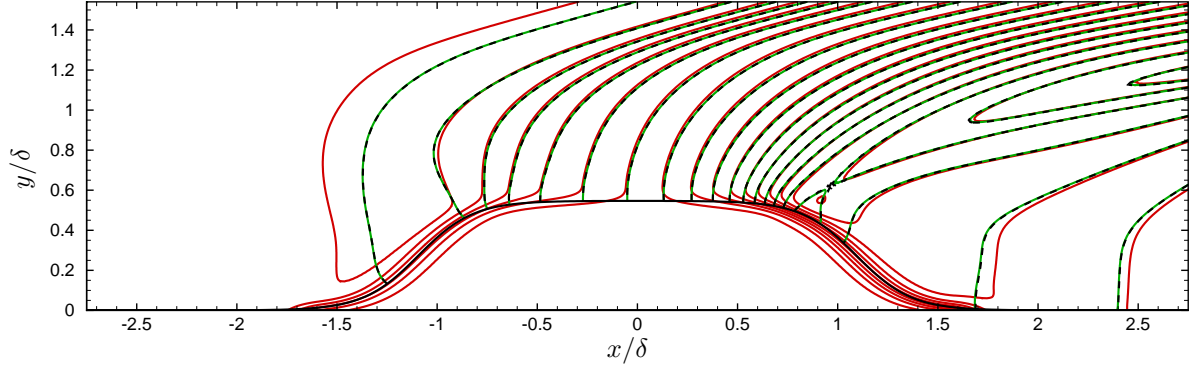
Where

$$\begin{aligned} a_1 &= S_\xi \xi - \frac{1}{2} \log \left(\frac{b_+}{b_-} \right), \\ a_2 &= \frac{1}{2} \log \left(\frac{b_+}{b_-} \right), \\ b_\pm &= 1 + (\exp(\pm S_\xi) - 1) \frac{x_c - x_{in}}{x_{out} - x_{in}}. \end{aligned} \quad (5.9)$$

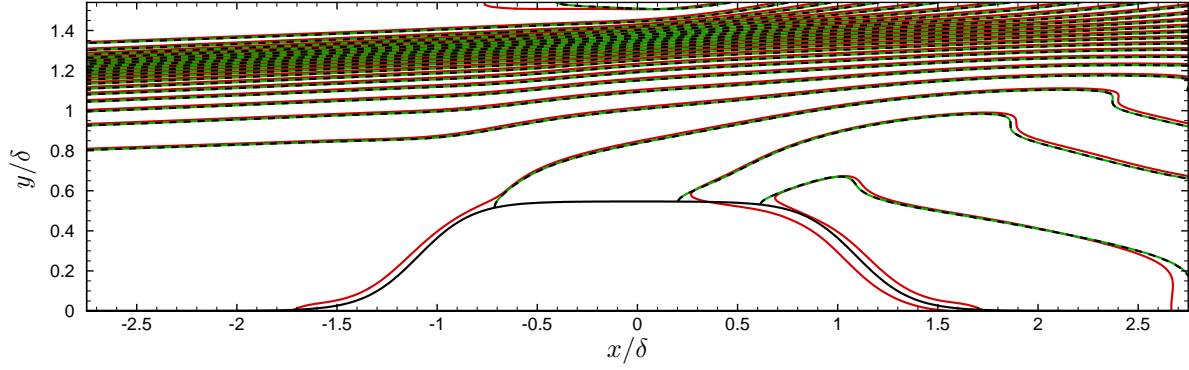
ξ and η are the computational domain variables and vary from 0 to 1. x_{in} and x_{out} are the x locations of the domain inlet and outlet, respectively. x_c is the center of the clustering in the x direction. H is the domain height. $B(x)$ is the function which defines the shape of the bottom edge of the domain. For the body-fitted grid, $B(x)$ is equal to the function governing the roughness geometry, $h(x)$. For the cut-cell method and the immersed boundary method, $B(x)$ is simply zero. S_ξ and S_η control the amount of clustering in the x and y directions, respectively. The grids were generated with $S_\xi = 10$, $S_\eta = 3.6$, and $x_c/\delta = 0.0$. The numerical domain starts at $x/\delta = -50.0$ and ends at $x/\delta = 83.0$. The top of the domain is located at $y/\delta = 8.62$. The grid used for the immersed boundary method is the same as the cut-cell method grid except 600 points are used in the y direction.

5.3 Results

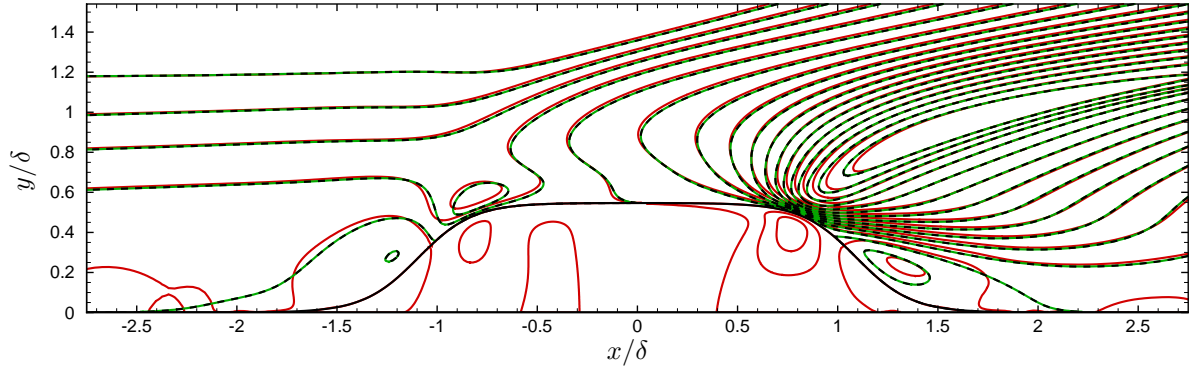
Figure 5.3 shows contour plots of pressure, density, and v velocity in the vicinity of the roughness element for the three methods. The results from the body-fitted grid and the cut-



(a) Pressure contour. Non-dimensionalized as p/p_∞ . 18 levels from 0.45 to 1.3.

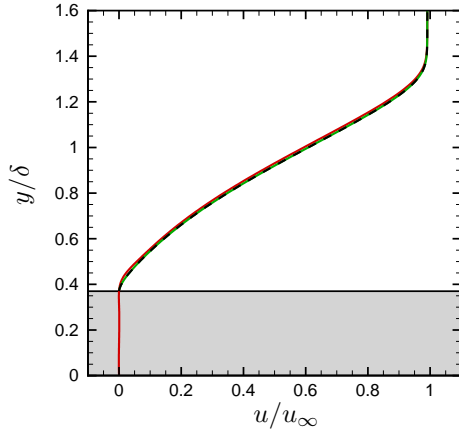


(b) Density contour. Non-dimensionalized as ρ/ρ_∞ . 22 levels from 0.15 to 1.2.

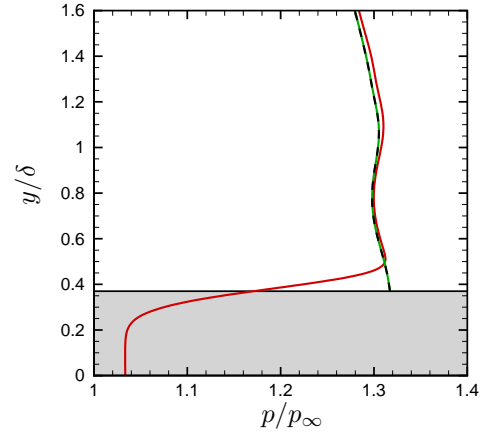


(c) v velocity contour. Non-dimensionalized as v/u_∞ . 17 levels from -0.12 to 0.04.

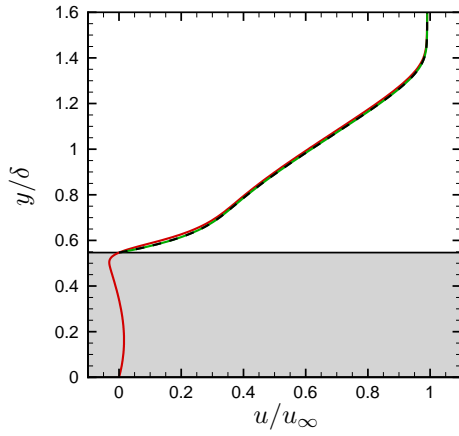
Figure 5.3: Contours in vicinity of the two-dimensional hyperbolic tangent roughness element. Immersed boundary method —; Cut-cell method —; Body-fitted grid ---.



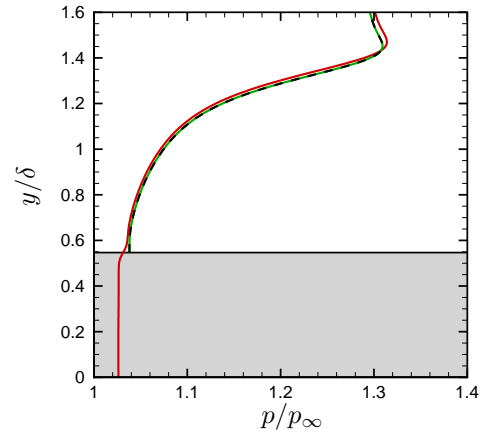
(a) Streamwise velocity at $x/\delta = -1.0$.



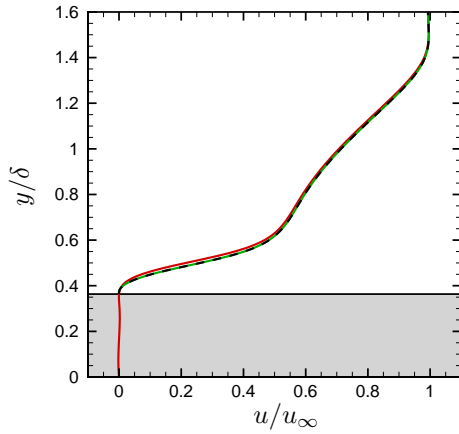
(b) Pressure at $x/\delta = -1.0$.



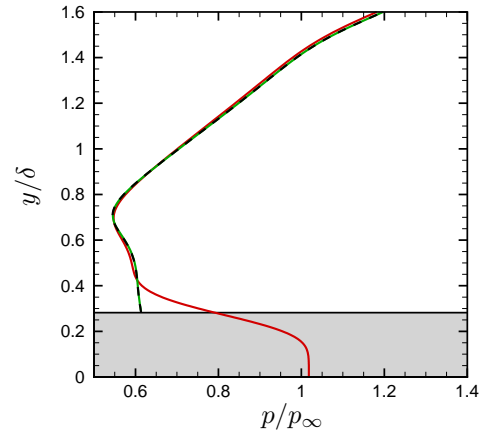
(c) Streamwise velocity at $x/\delta = 0.0$.



(d) Pressure at $x/\delta = 0.0$.



(e) Streamwise velocity at $x/\delta = 1.0$.



(f) Pressure at $x/\delta = 1.0$.

Figure 5.4: Streamwise velocity and pressure profiles for two-dimensional hyperbolic tangent roughness element. Roughness element shown in gray. Immersed boundary method —; Cut-cell method —; Body-fitted grid ---.

cell method match exactly. Both results were confirmed to be grid converged by increasing the grids points in the x direction to 1200 and the points in the y direction to 600. There were no visible changes in the results. The results from the immersed boundary method appear similar to the body-fitted results away from the roughness but there are significant differences near the roughness. The immersed boundary method results were not checked for grid convergence due to the method's low spatial convergence rate.

Figure 5.4 shows u velocity and pressure profiles at various x/δ locations for the three methods. The profiles are at the upstream curved portion of the roughness ($x/\delta = -1.0$), the center of the roughness ($x/\delta = 0.0$), and the downstream curved portion of the roughness ($x/\delta = 1.0$). Just as with the contours, the profiles for the body-fitted grid and cut-cell method match. There are noticeable differences between the body-fitted and immersed boundary method which are greatest near the roughness.

Despite the immersed boundary simulation being performed on a slightly denser grid, the cut-cell method was able to reproduce the body-fitted results while the immersed boundary results still showed significant differences. It may be possible to improve the immersed boundary results. There are many slight variations of the immersed boundary method but only one was tested here. However, all the variations will use a delta function to apply the forcing. This will always extend the influence of the body force away from the wall. The cut-cell method does not suffer from this since the boundary conditions are applied only at the boundary. Based on the results of this test, it was decided to use the cut-cell method for the remainder of the simulations in this study.

CHAPTER 6

Code Validation

Since a new code was developed for this work, several validation cases were performed. The results from two cases will be presented here. The first problem is subsonic flow over a cylinder. The second is supersonic flow over a three-dimensional roughness constructed from hyperbolic tangents.

6.1 Subsonic Flow Over a Cylinder

The first validation case being presented is for Mach 0.2 flow over a cylinder. The purpose of this simulation was to test the new solver on an unsteady problem. Since the multi-zone method requires interpolation in time, it was important to test the method on an unsteady problem. Although the code solves the compressible Navier-Stokes equations, this incompressible problem was selected due to its widely known solution. The Reynolds number for the flow based on the cylinder diameter (D) is 150. This places the solution in the periodic vortex shedding regime.

6.1.1 Simulation Setup

Figure 6.1 shows the grid used in the simulation. The red grid is the coarse grid, which covers the entire domain. The grid is uniformly spaced in both directions. The coarse grid has 300 grid points in the x direction and 150 grid points in the y direction. The first refinement zone is shown in green and the second in blue. Each refinement zone increases the resolution by a factor of three. The figure only shows a portion of the domain near the

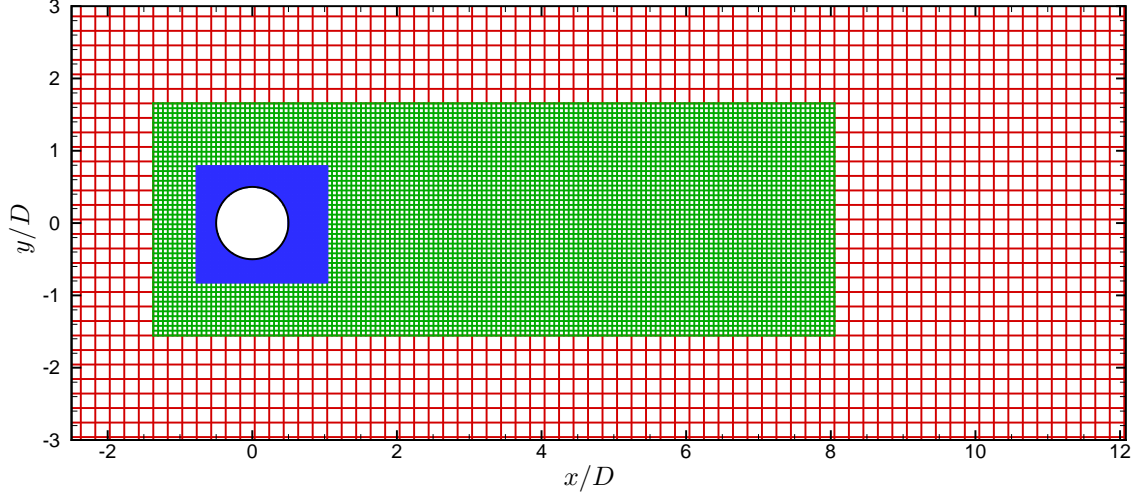
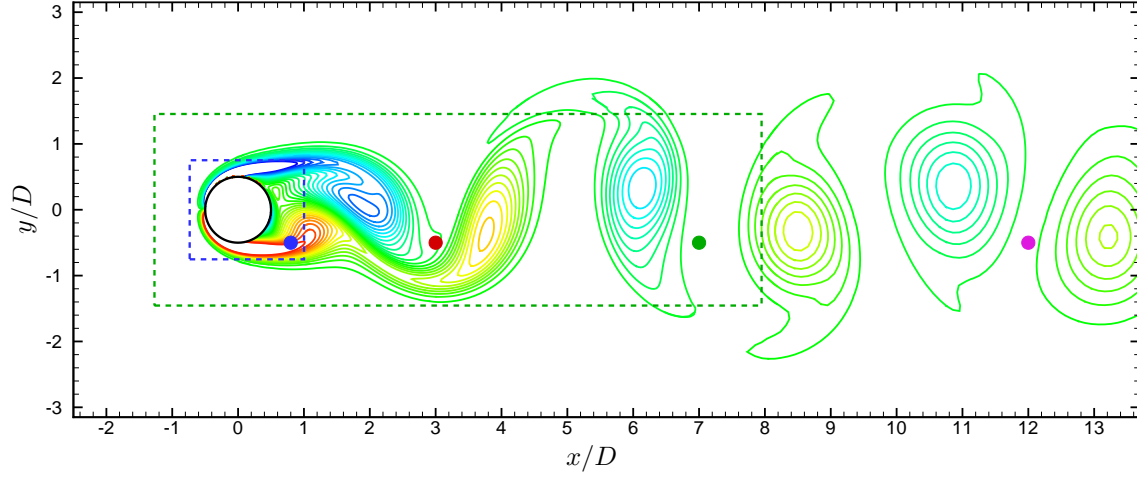


Figure 6.1: Grid in vicinity of cylinder. Points inside the cylinder are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue. Only every second point shown in both directions.

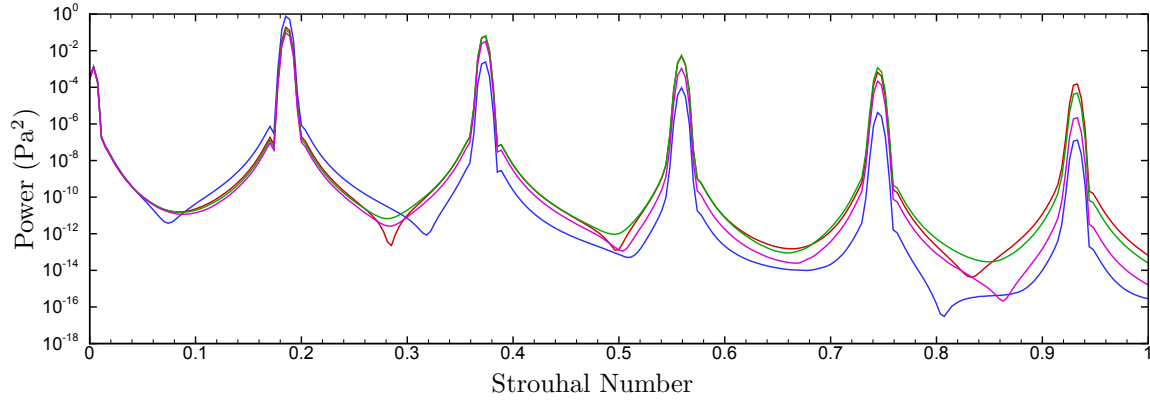
cylinder. In the x direction, the domain starts at $x/D = -5$ and ends at $x/D = 15$. The domain has a length of $10D$ in the y direction and is centered at $y/D = 0$. Non-reflective boundary conditions [99] are used along all edges of the domain. A sponge layer [98] was also used at the inlet with the freestream conditions used as the target. This ensured that the non-reflective inlet did not deviate from the desired freestream conditions.

6.1.2 Results

Figure 6.2a shows contours of vorticity for the flow after reaching a periodic state. The results show that the vortices are able to pass across the refinement zone boundaries unimpeded. Figure 6.2a also shows the location of four probes placed in the flow. Once the solution reached a periodic state, the pressure at each probe location was recorded as a function of time. The dominant frequency of the pressure history was computed using a fast Fourier transform (FFT). The results of the FFT are shown in Figure 6.2b. The different line colors correspond to the same colored probe in Figure 6.2a. Figure 6.2b shows that all the probes measured the same dominant frequency. This suggests that the multi-zone refinement did



(a) Vorticity contour. Non-dimensionalized as $\omega D/u_\infty$ with 30 levels from -4.0 to 4.0. Colored dots denote the location of pressure probes. First refinement zone edge - - -; Second refinement zone edge - - -.



(b) FFT of the pressure history at each probe. The line color corresponds to the probe color in Figure (a).

Figure 6.2: Results from subsonic flow over a cylinder.

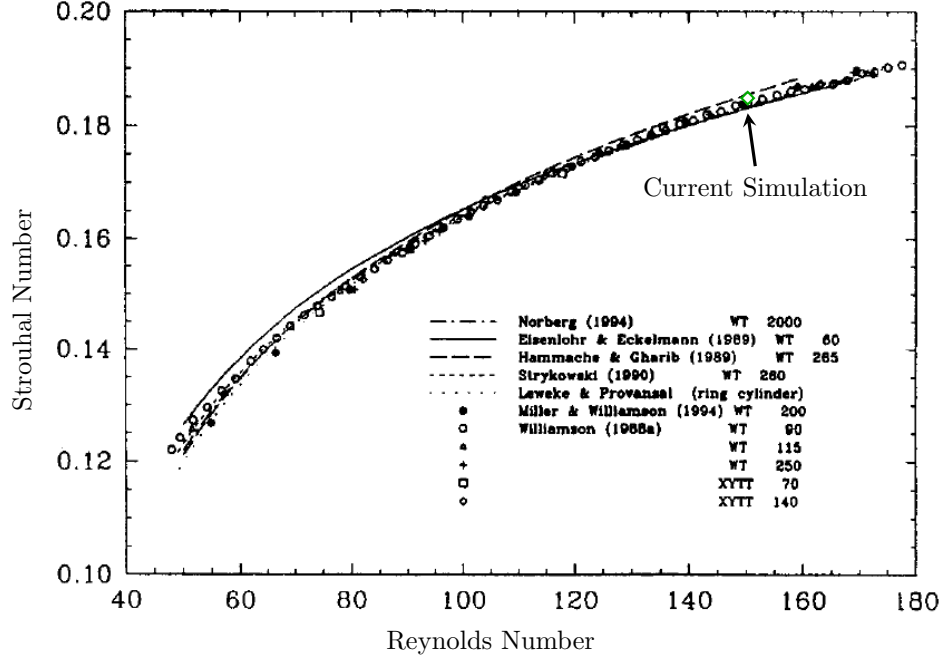


Figure 6.3: Strouhal number versus Reynolds number for incompressible flow over a cylinder. Taken from Williamson [100]. Results from current simulation shown as \diamond .

not have a noticeable negative effect on the solution. The first peak measured by all the probes is at a Strouhal number of 0.185. The remaining peaks are harmonics of the first.

Figure 6.3 is a plot taken from the cylinder wake review paper by Williamson [100]. The plot shows the Strouhal number versus Reynolds number for the current problem. The data comes from a number of different experiments. The Strouhal number from the current simulation was added to the plot. The results agree very well with the experimental data. These results suggest that the cut-cell method and the multi-zone refinement are working correctly.

6.2 Hyperbolic Tangent Roughness

The second validation case is Mach 4.8 boundary-layer flow over a three-dimensional isolated roughness element constructed from hyperbolic tangent curves. The purpose of this simulation was to test the code on a problem closer to the problem of interest, hypersonic

flow over an isolated roughness element.

6.2.1 Simulation Setup

This simulation is a three-dimensional version of the case used to compare the cut-cell and immersed boundary methods in Chapter 5. The x and y axis are still in the flow direction and normal to the wall, respectively. The z axis is in the spanwise direction. The roughness height to undisturbed boundary-layer thickness ratio (k/δ) is still 0.547. The equation for the three-dimensional roughness geometry is

$$h(x, z) = \frac{k}{4} h_x(x) h_z(z) \quad (6.1)$$

where

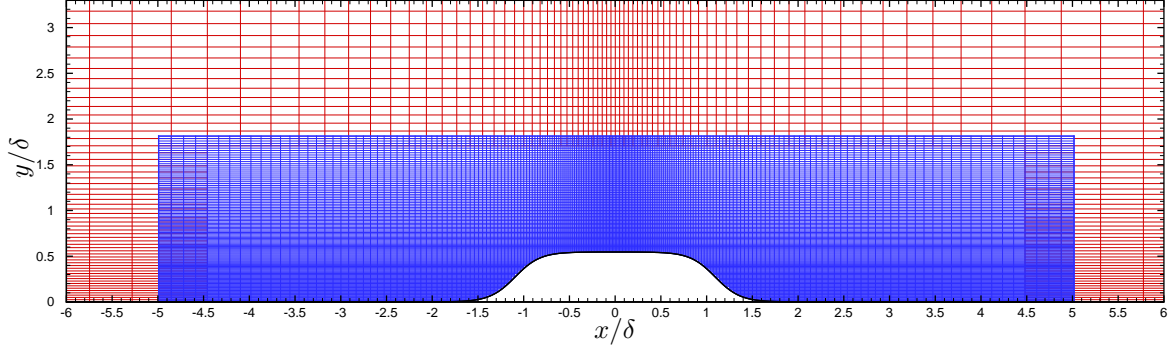
$$h_x(x) = \tanh(S_r(x + L_r)) - \tanh(S_r(x - L_r)), \quad (6.2)$$

$$h_z(z) = \tanh(S_r(z + L_r)) - \tanh(S_r(z - L_r)). \quad (6.3)$$

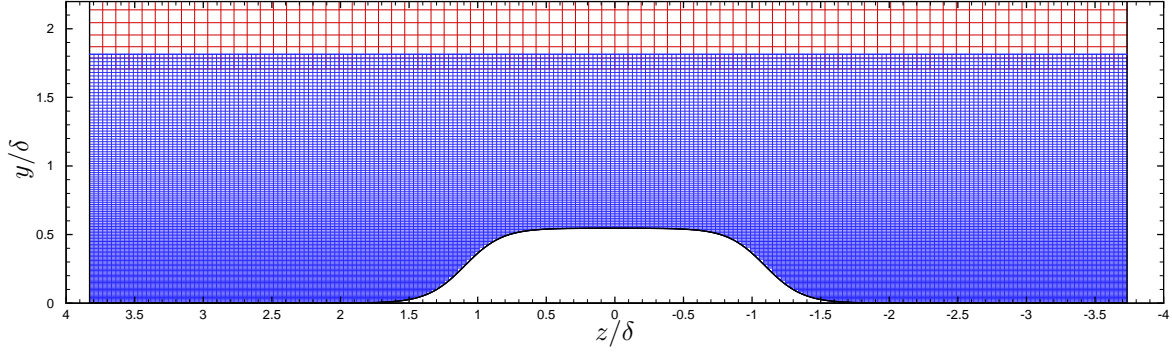
The same values for S_r and L_r are used here ($S_r = 20$, $L_r = 0.2$). The same freestream conditions are used in this simulation and are summarized in Table 5.1.

The boundary conditions used in this simulation are similar to the ones used in Chapter 5. A compressible similarity solution is imposed at the inlet of the numerical domain and held fixed with time. At the outlet, the conservative variables are extrapolated from the interior of the domain. At the top of the domain, the u velocity component and temperature are set to their freestream values while the derivative of the v velocity component, w velocity component, and pressure with respect to y was set to zero. The same sponge layer is also used. The flat plate and the roughness are no-slip adiabatic walls. Symmetry boundary conditions are used at both edges on the domain in the z direction.

The grid in the vicinity of the roughness is shown in Figure 6.4. The coarse grid has 120 points in the x direction, 80 points in the y direction, and 80 points in the z direction. The



(a) $x - y$ plane at $z/\delta = 0$.

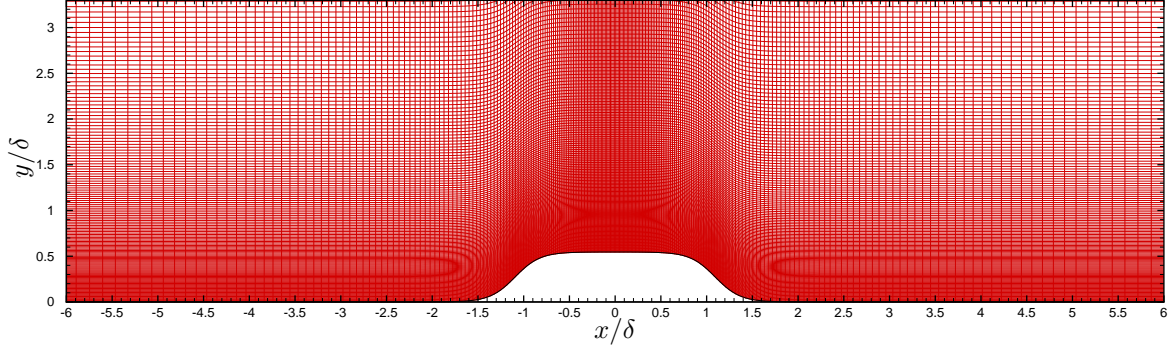


(b) $y - z$ plane at $x/\delta = 0$.

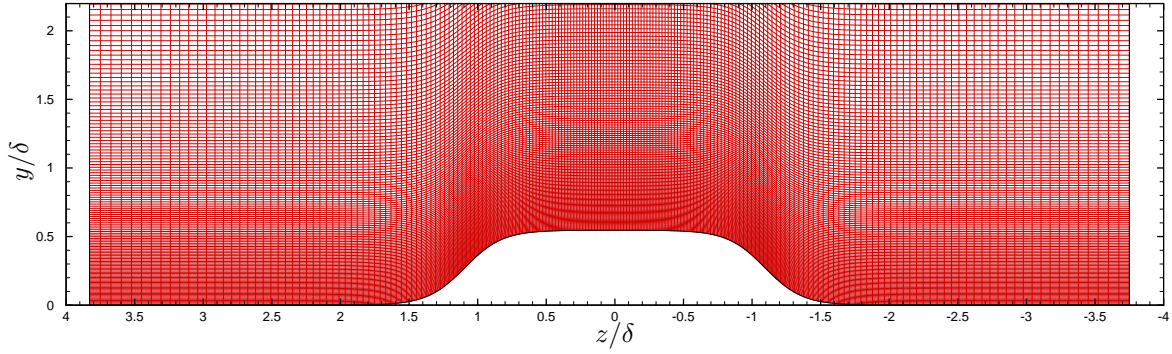
Figure 6.4: Grid in the vicinity of the three-dimensional hyperbolic tangent roughness element used in multi-zone cut-cell simulation. Grid points inside the roughness were removed. Coarse grid shown in red. Refinement zone with three times the resolution shown in blue.

grids are clustered near the roughness element in the x direction and near the wall in the y direction. The grid is uniformly spaced in the z direction. The grid was generated using Equations 5.7 and 5.8 for the x and y coordinates, respectively, with $S_\xi = 10$, $S_\eta = 3.4$, and $x_c/\delta = 0.0$. Only the domain around the roughness element is shown in the figure. The full numerical domain starts at $x/\delta = -24.0$ and ends at $x/\delta = 72.0$. The top of the domain is located at $y/\delta = 8.61$ and the domain half width is $W/\delta = 3.82$. A single refinement zone was used in the simulation which increased the resolution by a factor of three.

The simulation was also performed on a body-fitted grid for comparison. The body-fitted grid in the vicinity of the roughness is shown in Figure 6.5. The grid has 600 points in the x direction, 300 points in the y direction, and 300 points in the z direction. The x and y point distribution is given by Equations 5.7 and 5.8 with $S_\xi = 10.0$, $S_\eta = 2.0$, and $x_c/\delta = 0.0$. In



(a) $x - y$ plane at $z/\delta = 0$.



(b) $y - z$ plane at $x/\delta = 0$.

Figure 6.5: Body-fitted grid in the vicinity of the three-dimensional hyperbolic tangent roughness element.

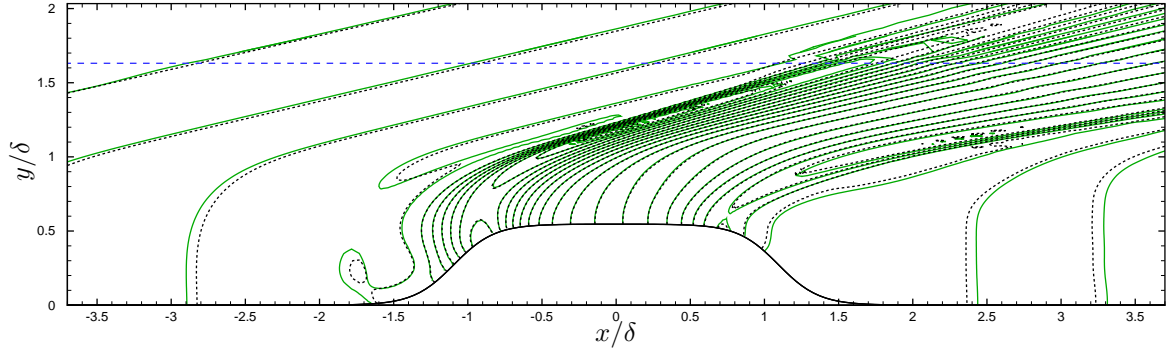
Equation 5.8, $B(x)$ is now a function of x and z and is equal to $h(x, z)$. The z distribution is given by

$$z = W \frac{\sinh(S_\zeta (2k/KL - 1))}{\sinh(S_\zeta)}, \quad k = 1, \dots, KL \quad (6.4)$$

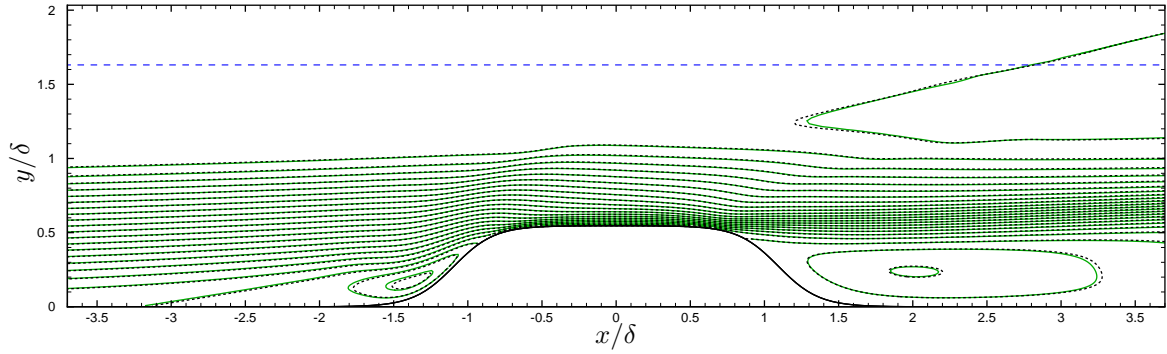
where KL is the total number of grid points in the z direction and $S_\zeta = 2.0$. The extents of the domain are the same as the multi-zone cut-cell grid.

6.2.2 Results

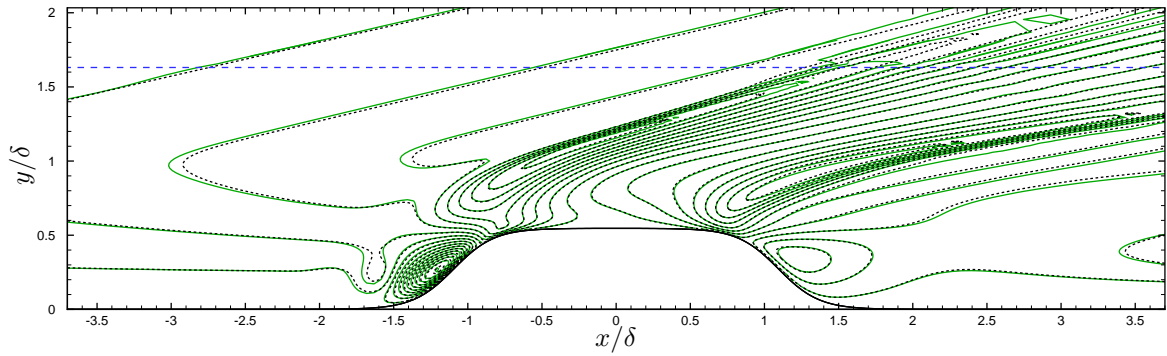
The results from the body-fitted grid were compared to results presented by Marxen and Iaccarino [96]. The results matched very well with Marxen and Iaccarino's results and are assumed to be correct. Contours of pressure, u velocity, and v velocity near the roughness at the $z/\delta = 0$ plane are shown in Figure 6.6. Contours of pressure, u velocity, and w velocity



(a) Pressure contour. Non-dimensionalized as p/p_∞ . 32 levels from 0.05 to 1.6.

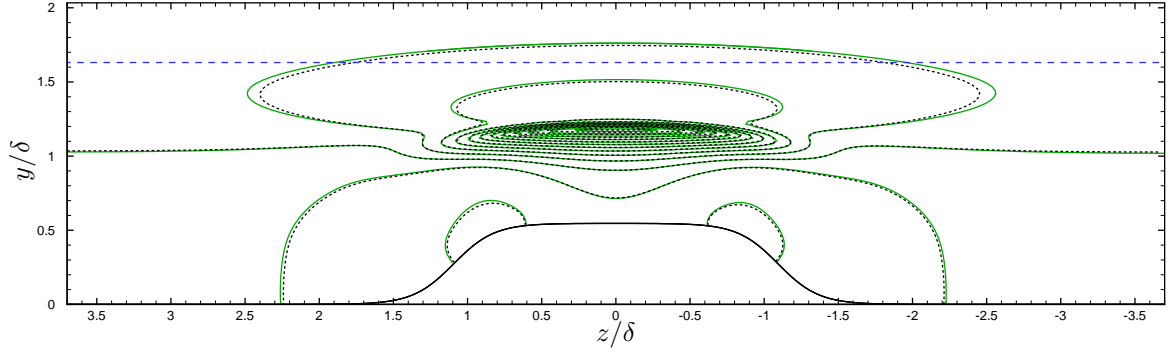


(b) u velocity contour. Non-dimensionalized as u/u_∞ . 23 levels from -0.1 to 1.0.

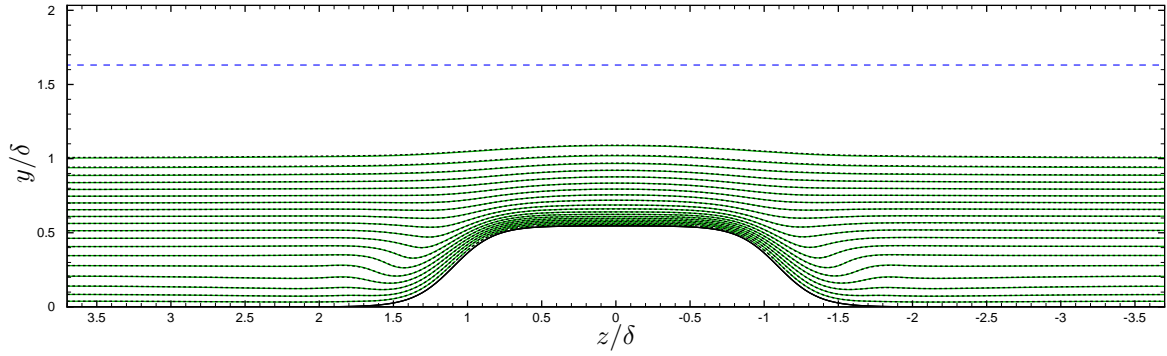


(c) v velocity contour. Non-dimensionalized as v/u_∞ . 24 levels from -0.1 to 0.1.

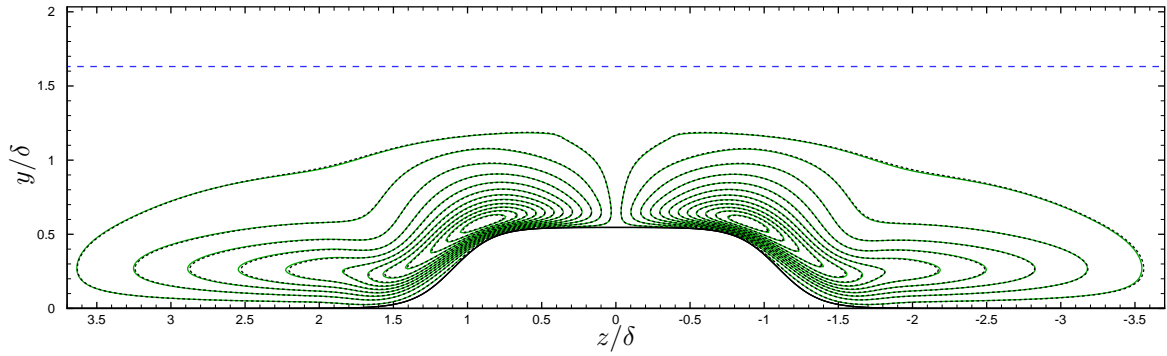
Figure 6.6: Contours in vicinity of three-dimensional hyperbolic tangent roughness element at $z/\delta = 0$ plane. Multi-zone cut-cell simulation —; Refinement zone edge ---; Body-fitted simulation ---.



(a) Pressure contour. Non-dimensionalized as p/p_∞ . 32 levels from 0.1 to 1.6.



(b) u velocity contour. Non-dimensionalized as u/u_∞ . 19 levels from 0.05 to 0.95.



(c) w velocity contour. Non-dimensionalized as w/u_∞ . 24 levels from -0.1 to 0.1.

Figure 6.7: Contours in vicinity of three-dimensional hyperbolic tangent roughness element at $x/\delta = 0$ plane. Multi-zone cut-cell simulation —; Refinement zone edge ---; Body-fitted simulation ---.

near the roughness at the $x/\delta = 0$ plane are shown in Figure 6.7. The results from the two methods agree very well. Only small differences between the body-fitted results and the multi-zone cut-cell results are visible. Neither set of results were confirmed to be grid converged. Increasing the grid resolution may remove the differences. The results suggest that the code is working properly and can simulate high-speed flow over isolated roughness elements. In addition, the multi-zone method provided a 75% reduction in runtime compare to the single-grid method.

CHAPTER 7

Complex Geometries

The code developed in this work was primarily designed to simulate flow over isolated roughness elements. Most roughness elements that have been studied here and by other researches have had rather simple geometries such as cylinders, hemispheres, and cuboids. To demonstrate the code’s capabilities, two-dimensional simulations of flow over much more complex geometries were performed.

7.1 Supersonic Flow Over the Ubuntu Logo

The first simulation to be presented is supersonic freestream flow over the logo of the Ubuntu Linux distribution [101]. The Ubuntu logo is shown in Figure 7.1. The freestream flow was set to Mach 3 with a Reynolds number based on the outer diameter of the logo’s ring (D) of 5,000.



Figure 7.1: Logo for the Ubuntu Linux distribution [101].

ℓ_ξ	ℓ_η	S_ξ	S_η	P_ξ	P_η
0.195	0.195	5.0	5.0	4	4

Table 7.1: Grid constants for flow over the Ubuntu logo.

7.1.1 Simulation Setup

Figure 7.2a shows the grid used for the simulation. The simulation used a coarse grid and two refinement zones. Both refinement zones increase the resolution by a factor of three. The coarse grid has 300 points in the x direction and 150 points in the y direction. The figure only shows the grid near the logo. The coarse grid has a total length of $40D$ and is centered about the logo. The coarse grid has a height of $20D$ and is also centered about the logo. Figure 7.2b shows the grid near the upstream circle of the logo. The grid was generated by the following equations:

$$x = \frac{L}{2} \left(A_\xi + |2\xi - 1|^{P_\xi} (B_\xi - A_\xi) \right), \quad (7.1)$$

$$y = \frac{H}{2} \left(A_\eta + |2\eta - 1|^{P_\eta} (B_\eta - A_\eta) \right), \quad (7.2)$$

where

$$A_\xi = \ell_\xi(2\xi - 1), \quad (7.3)$$

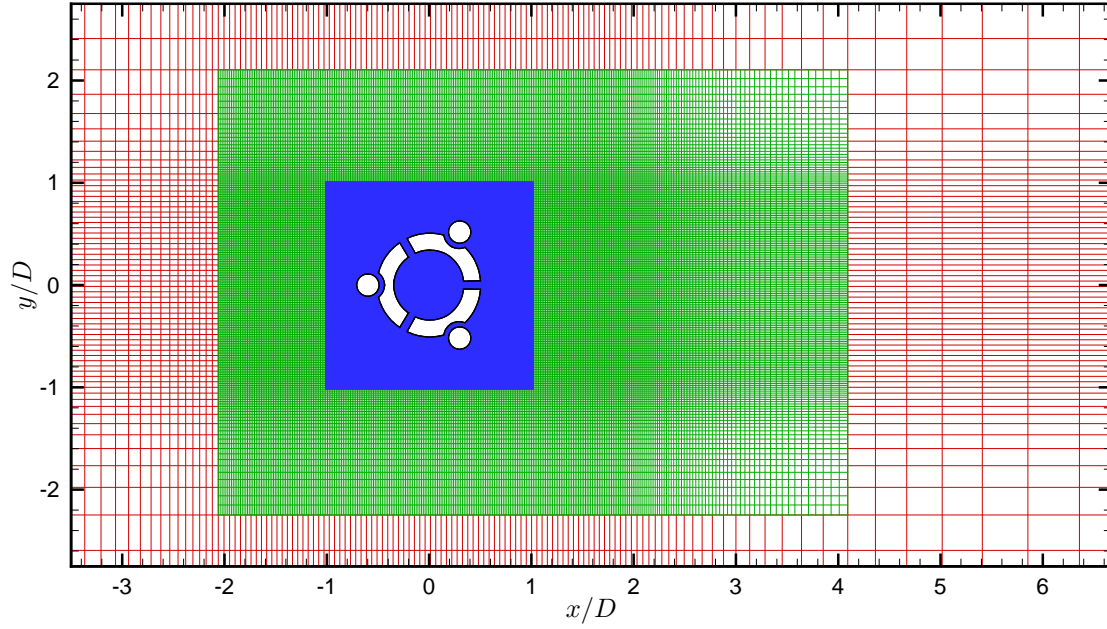
$$A_\eta = \ell_\eta(2\eta - 1), \quad (7.4)$$

$$B_\xi = \frac{\sinh(S_\xi(2\xi - 1))}{\sinh(S_\xi)}, \quad (7.5)$$

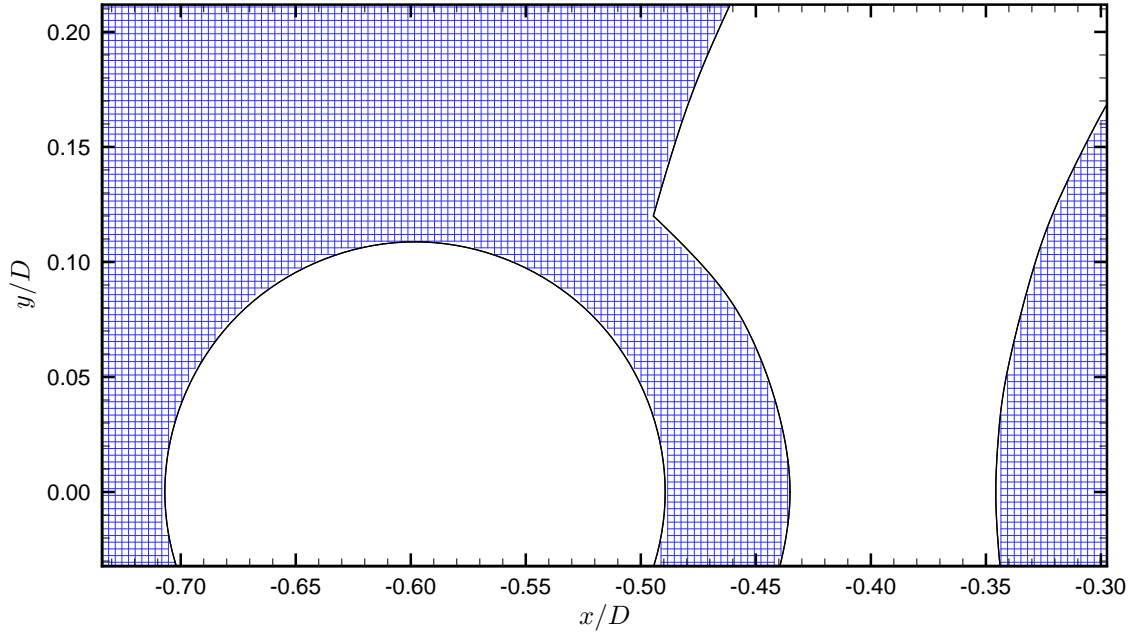
$$B_\eta = \frac{\sinh(S_\eta(2\eta - 1))}{\sinh(S_\eta)}. \quad (7.6)$$

ξ and η are the computational domain variables and vary from 0 to 1. L and H are the total length and height of the domain. The remaining constants are used to control the grid point clustering and the values are provided in Table 7.1.

Since the flow is supersonic, the freestream conditions are imposed at the inlet of the



(a) Grid near logo. Only every second point shown in both directions.



(b) Grid near upstream circle of logo. Every point shown.

Figure 7.2: Grid for flow over the Ubuntu logo. Points inside the logo are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue.

Mach number	M_∞	3.0
Velocity	u_∞	601.4 m/s
Pressure	p_∞	400.0 Pa
Temperature	T_∞	100.0 K
Reynolds number	Re_D	5000

Table 7.2: Freestream conditions for flow over the Ubuntu logo.

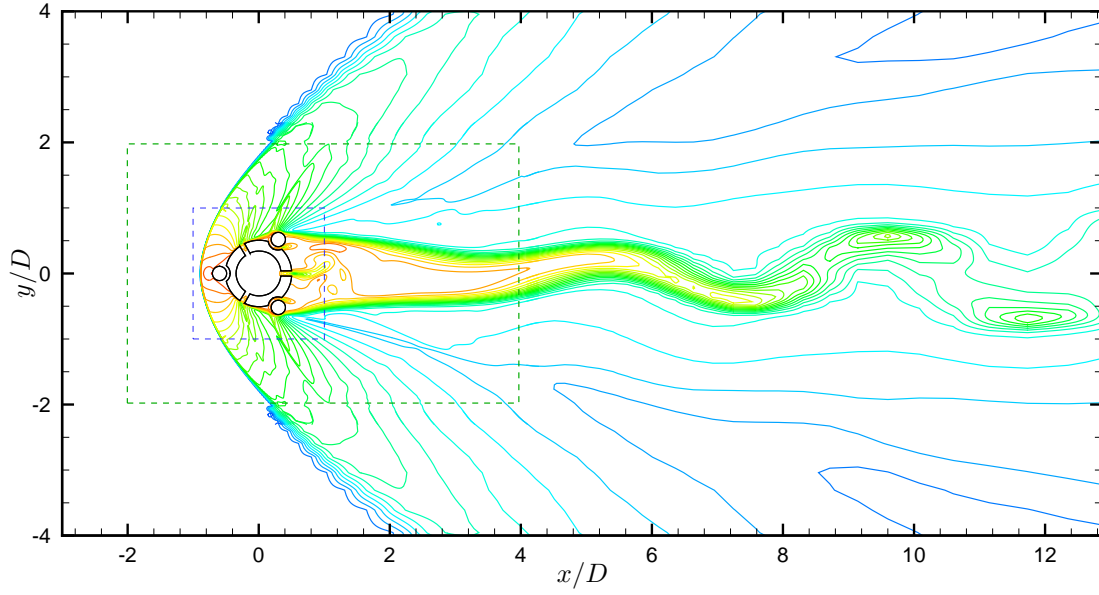
domain and held constant with time. The freestream conditions are given in Table 7.2. The outlet is supersonic so the conservative variables are extrapolated from the interior of the domain. At the top and bottom of the domain, the u velocity component and temperature are set to their freestream values while the derivative of the v velocity component and pressure with respect to y was set to zero. Sponge layers [98] are placed at the top and bottom of the domain to prevent any shocks from reflecting back into the domain. The logo has no-slip adiabatic walls.

7.1.2 Results

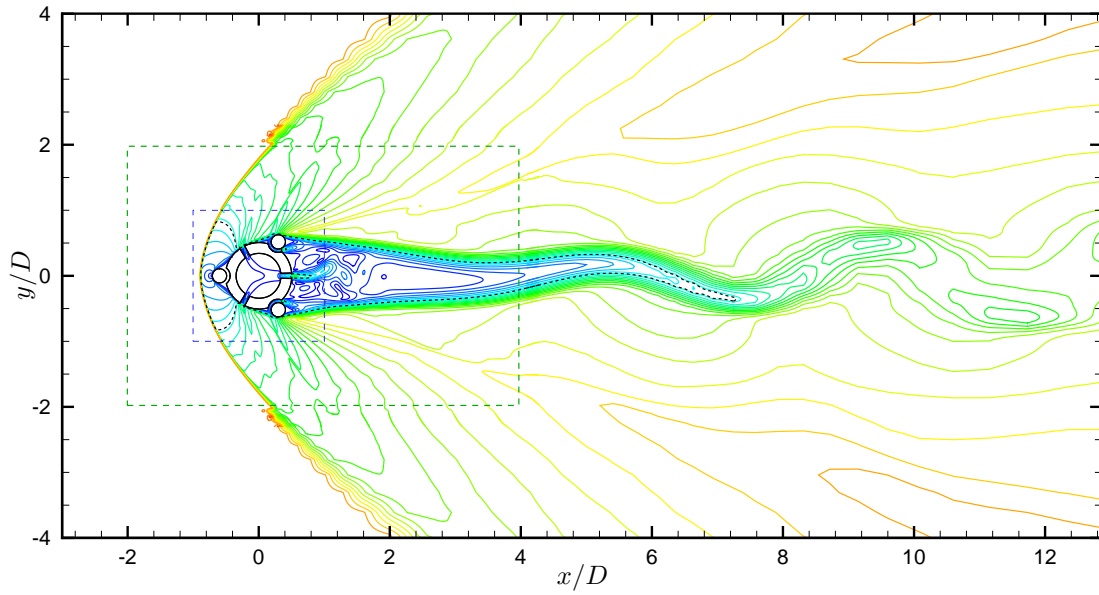
Contours of temperature and Mach number are shown in Figure 7.3. The dominant features of the flow are the bow shock created upstream of the logo and the unsteady wake. The results shown in the figure have reached a periodic state. The figure also demonstrates one of the weaknesses of the current multi-zone method. As the shock passes across a refinement zone boundary, the thickness of the shock increases abruptly. A weak reflected shock can sometimes also be propagated back into the refinement grid. The difference in the shock width and the strength of the reflected shock appears to be proportional to the strength of the shock.

Figure 7.4 shows contours of temperature and Mach number in the vicinity of the logo. Figure 7.5 shows the streamlines near the logo. Although there are no results to compare with for this flow, nothing stands out as incorrect in the flow field. This suggests the code is capable of handling complex geometries.

Once the solution reached a periodic state, the pressure was recorded at five locations

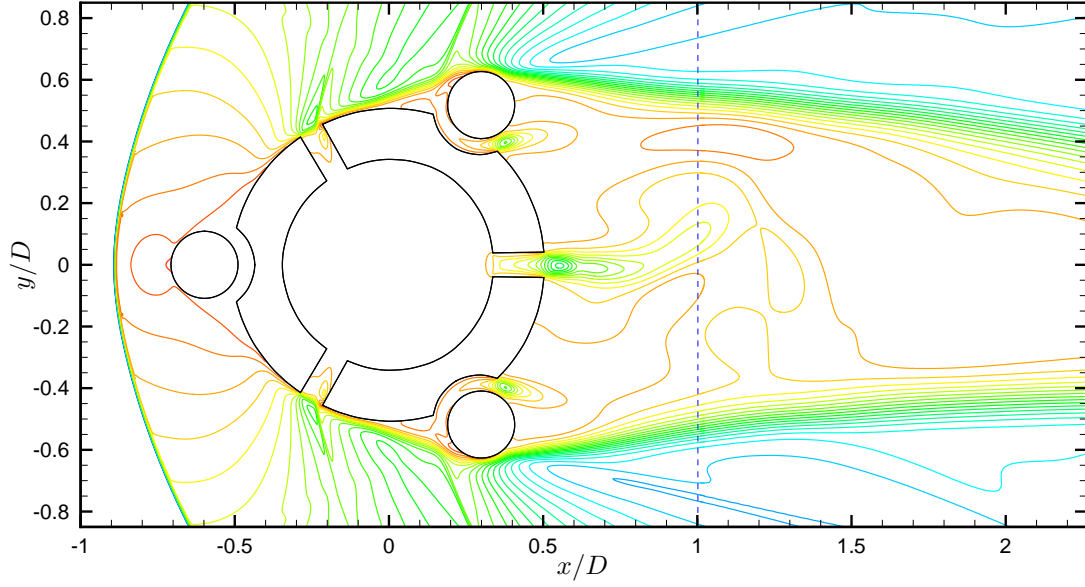


(a) Temperature contour. Non-dimensionalized as T/T_∞ . 25 levels from 0.9 to 2.8.

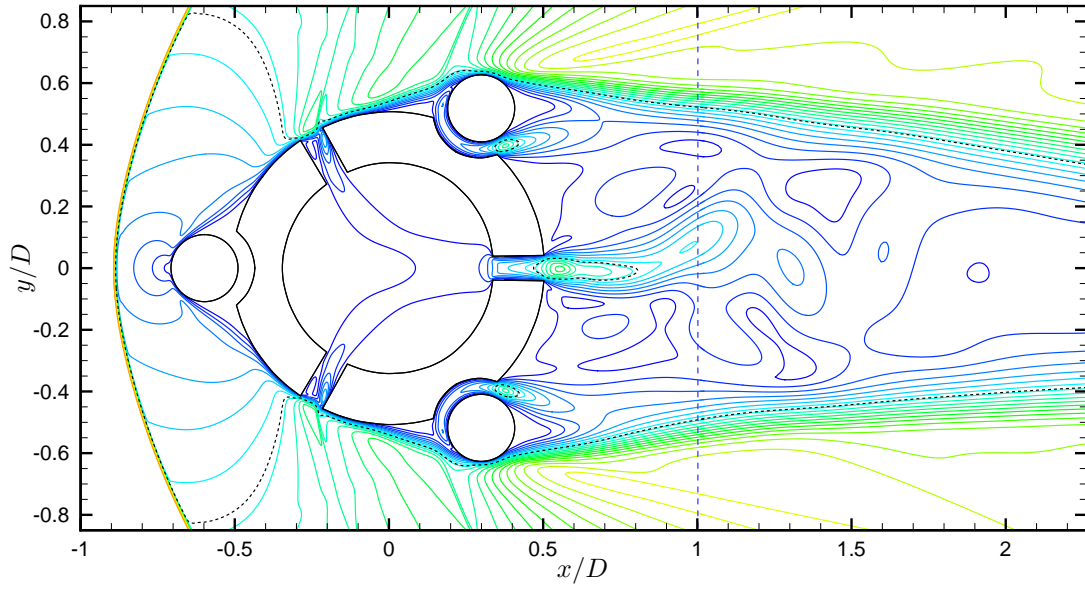


(b) Mach number contour. 25 levels from 0.1 to 3.3. Sonic contour shown as ---.

Figure 7.3: Temperature and Mach number contours for flow over the Ubuntu logo. First refinement zone edge ---. Second refinement zone edge ---.



(a) Temperature contour. Non-dimensionalized as T/T_∞ . 25 levels from 0.9 to 2.8.



(b) Mach number contour. 25 levels from 0.1 to 3.3. Sonic contour shown as ---.

Figure 7.4: Temperature and Mach number contours near the Ubuntu logo. Second refinement zone edge ---.

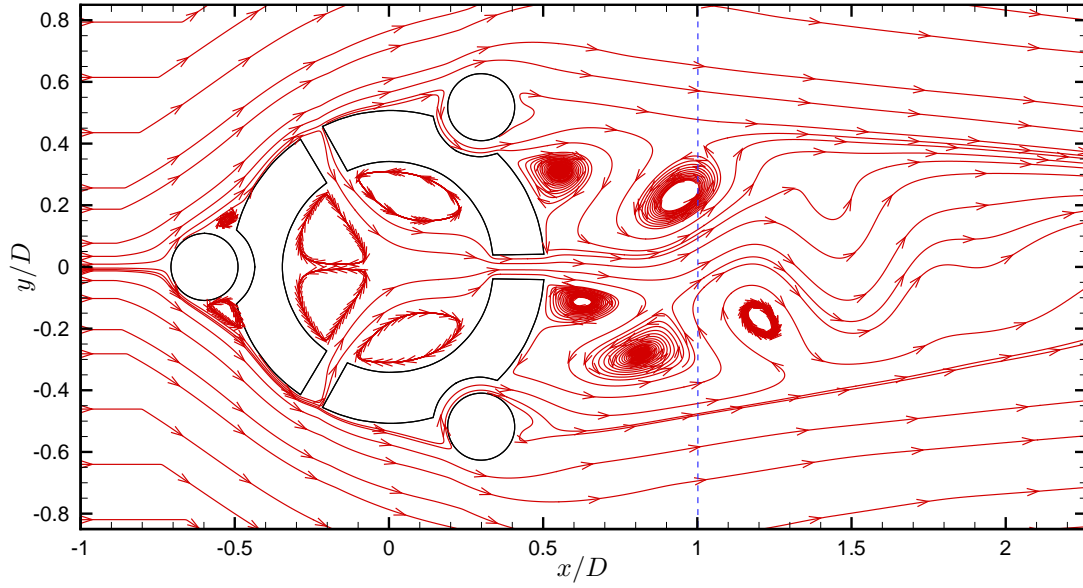
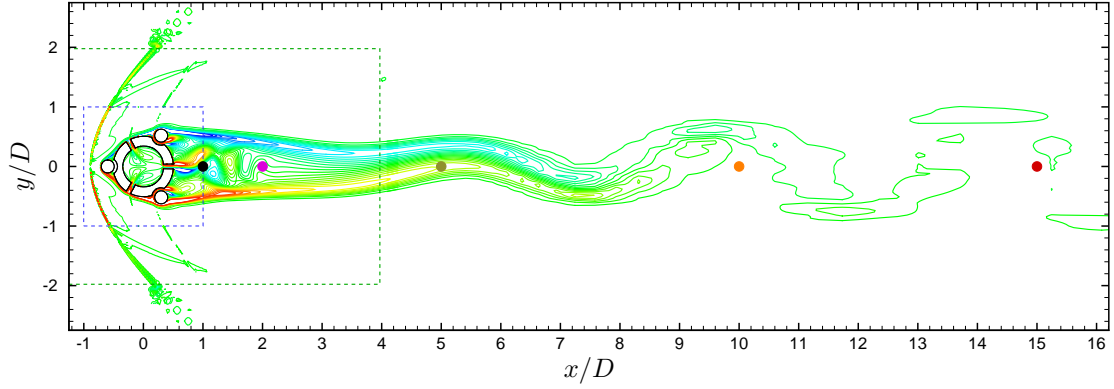


Figure 7.5: Streamlines near the Ubuntu logo. Second refinement zone edge ---.

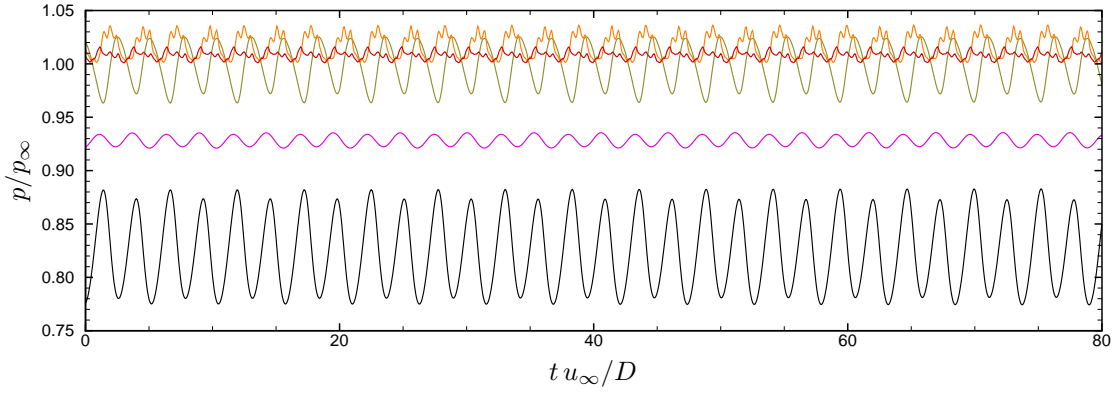
in the flow. Figure 7.6a shows contours of vorticity and the location of the five pressure probes as colored dots. Figure 7.6b shows the pressure recorded at the pressure probes. An FFT was used to compute the dominant frequency in the flow. Before computing the FFT, the mean was removed from each of the pressure histories and a Blackman window was applied. Figure 7.6c shows the results of the FFT. The different line colors correspond to the same colored probe in the Figure 7.6a. The same dominant frequency is measured by all the probes. The first peak is at a Strouhal number of 0.189. The remaining peaks are harmonics of the first.

7.2 Subsonic Flow Over the UCLA Logo

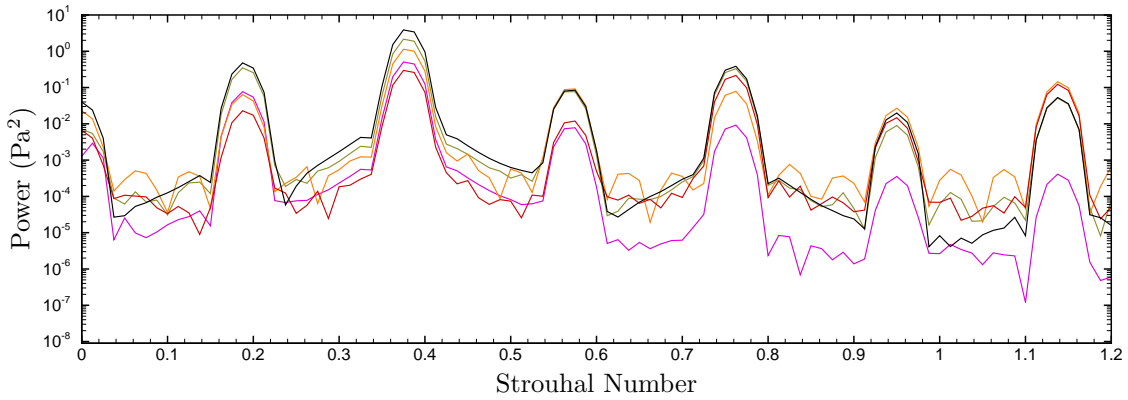
The second simulation with a complex geometry is subsonic flow over the logo for the University of California, Los Angeles. The UCLA logo is shown in Figure 7.7. The freestream flow has a Mach number of 0.8 and a Reynolds number based on the logo's height (h) of 1500.



(a) Vorticity contour. Colored dots denote the location of pressure probes. First refinement zone edge ---. Second refinement zone edge ---.



(b) Pressure history record at each probe. The line color corresponds to the probe color in Figure (a).



(c) FFT of the pressure history record at each probe. The line color corresponds to the probe color in Figure (a).

Figure 7.6: Frequency analysis results for flow over the Ubuntu logo.



Figure 7.7: Logo for the University of California, Los Angeles.

ℓ_ξ	ℓ_η	S_ξ	S_η	P_ξ	P_η
0.293	0.401	4.5	3.5	5	6

Table 7.3: Grid constants for flow over the UCLA logo.

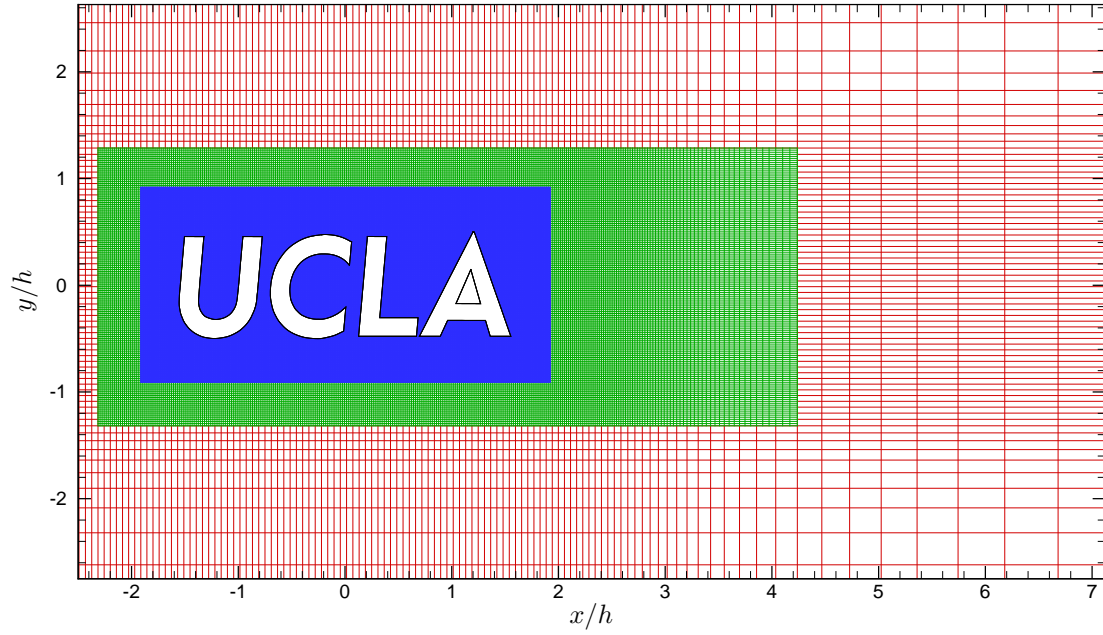
7.2.1 Simulation Setup

Figure 7.8a shows the grid used for the simulation. The simulation used a coarse grid and two refinement zones. Both refinement zones increased the resolution by a factor of three. Figure 7.8b shows the grid near the top of the gap between the letters “C” and “L” in the logo. The grid was generated using Equations 7.1 and 7.2. The length and height of the coarse grid are $30h$ and $10h$, respectively. The constants used in Equations 7.1 and 7.2 are given in Table 7.3. The coarse grid has 300 points in the x direction and 150 points in the y direction.

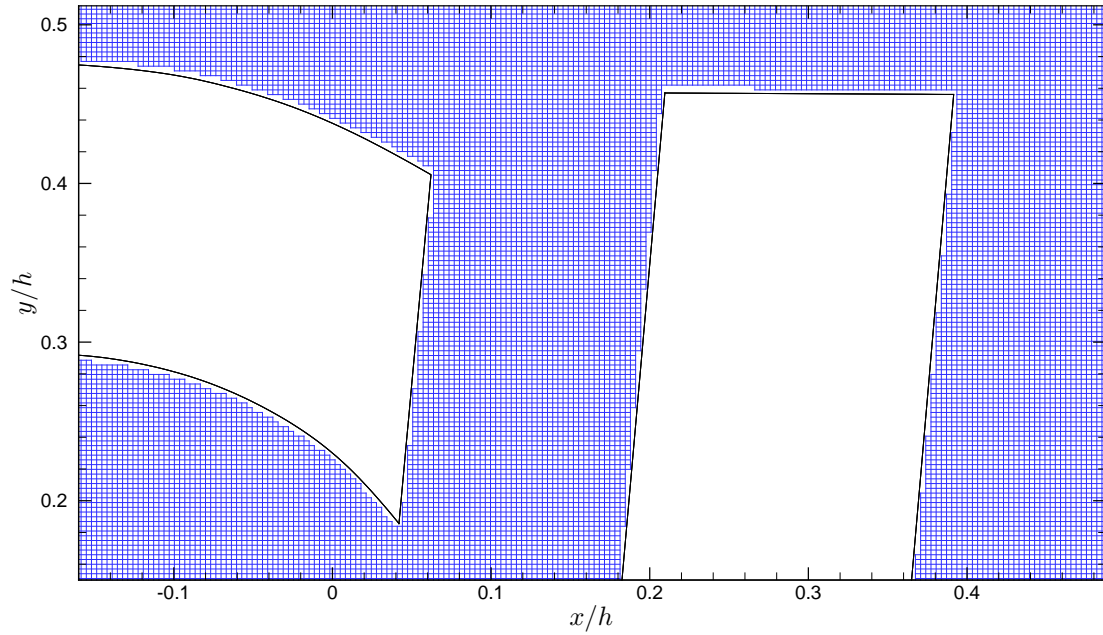
Non-reflective boundary conditions [99] are used along all the edges of the computational domain. A sponge layer [98] was also used at the inlet with the freestream conditions used as the target. This ensured that the non-reflective inlet did not deviate from the desired freestream conditions. The freestream conditions are given in Table 7.4. The logo has no-slip adiabatic walls.

Mach number	M_∞	0.8
Velocity	u_∞	160.4 m/s
Pressure	p_∞	400.0 Pa
Temperature	T_∞	100.0 K
Reynolds number	Re_h	1500

Table 7.4: Freestream conditions for flow over the UCLA logo.



(a) Grid near the logo. Only every second point shown in both directions.

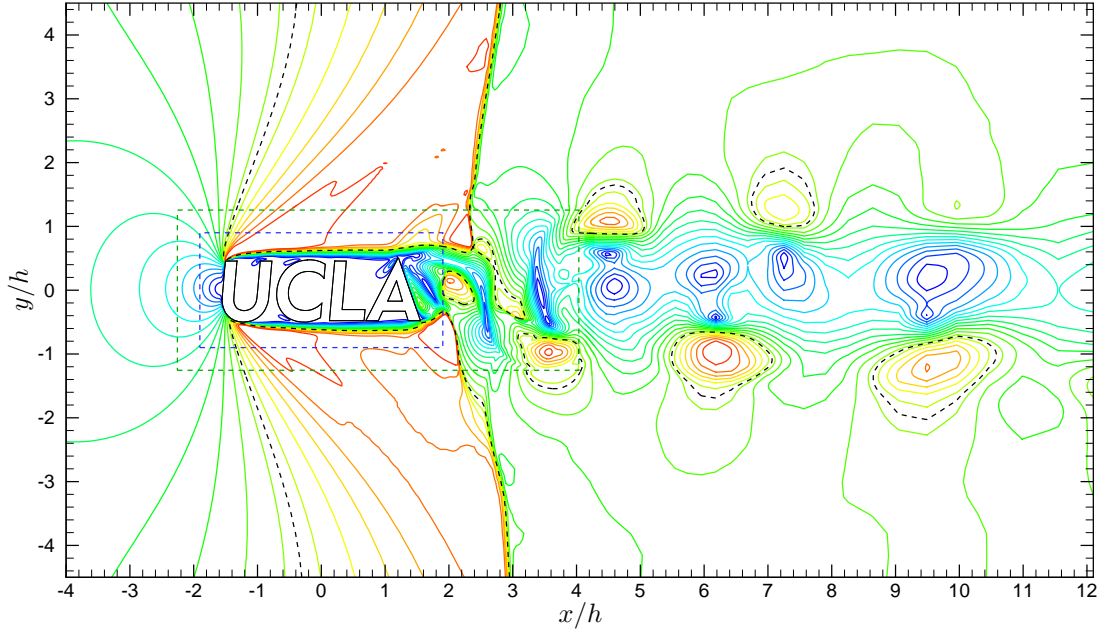


(b) Grid at the top of gap between the letters "C" and "L." Every point shown.

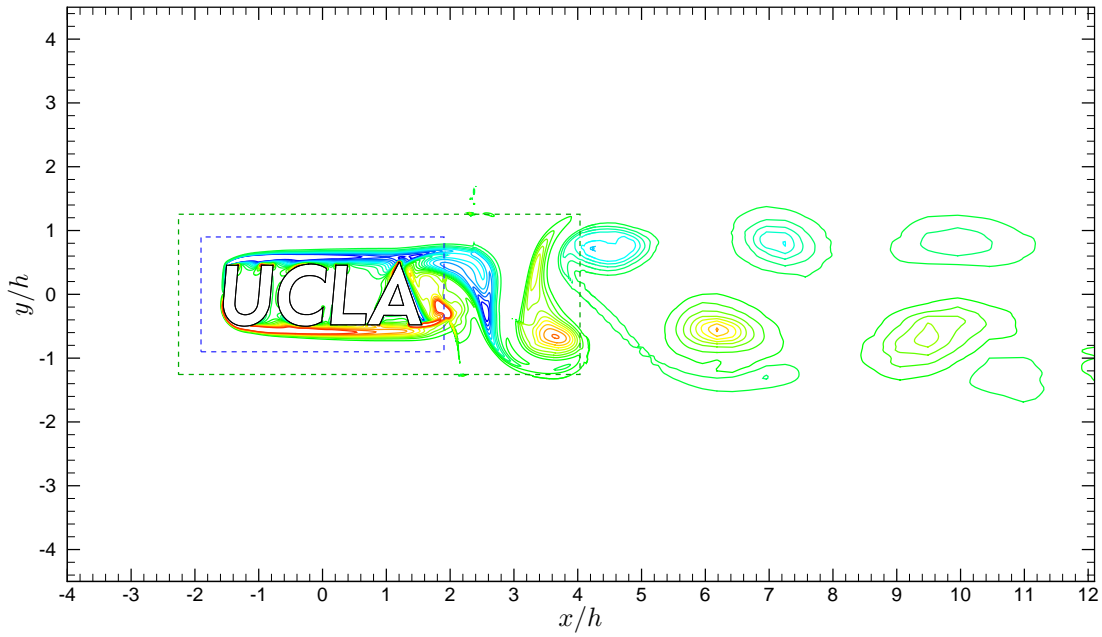
Figure 7.8: Grid for flow over the UCLA logo. Points inside the logo are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue.

7.2.2 Results

Figure 7.9 shows contours of Mach number and vorticity. As the flow moved over the logo, it was accelerated and became supersonic. Once past the logo, the flow became subsonic again although pockets of supersonic flow still exist in the unsteady wake region. Figure 7.10 shows the same contours but much closer to the logo. Shocklets can be seen in the wake of the logo. Figure 7.11 depicts the streamlines near the UCLA logo. Like the Ubuntu logo, there are no results to compare with for this flow.

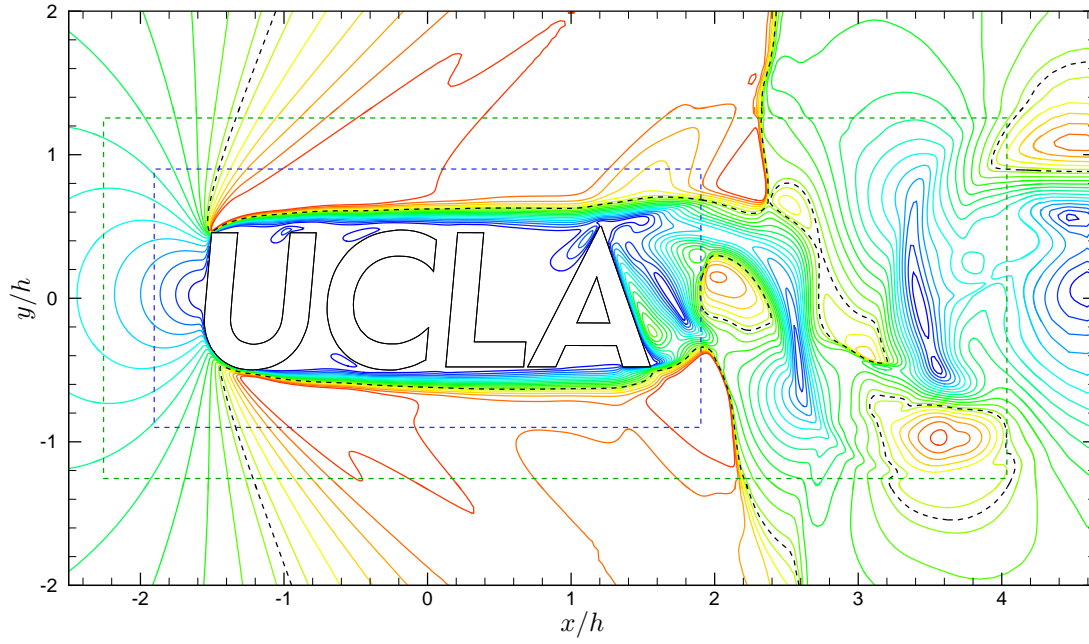


(a) Mach number contour. 19 levels from 0.1 to 1.4. Sonic contour shown as - - -.

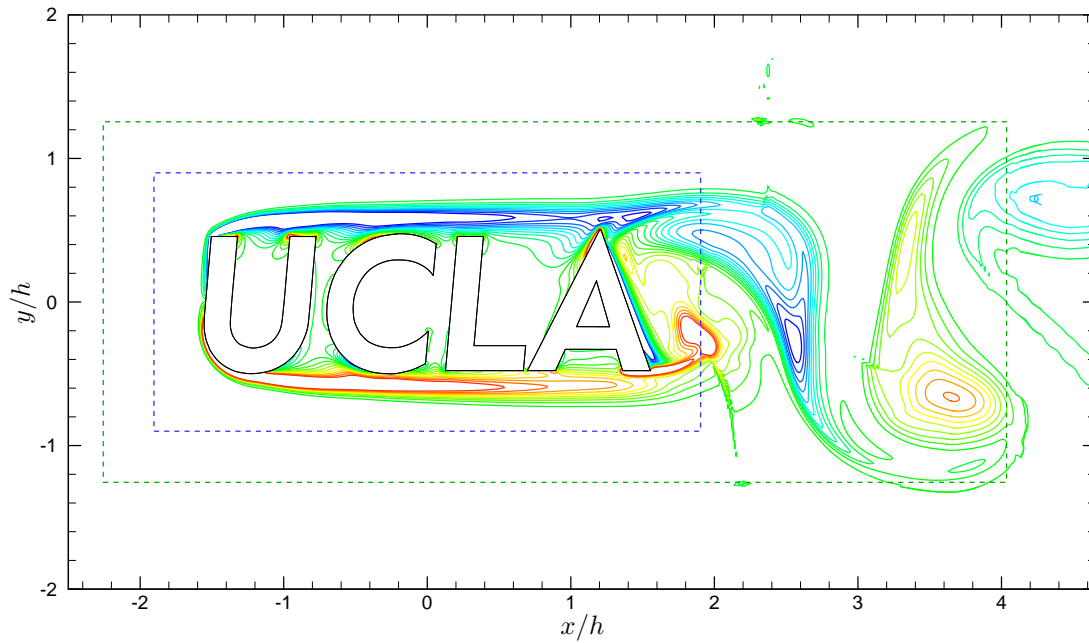


(b) Vorticity contour. Non-dimensionalized as $\omega h/u_\infty$. 20 levels from -10.0 to 10.0.

Figure 7.9: Mach number and vorticity contours for flow over the UCLA logo. First refinement zone edge - - -. Second refinement zone edge - - -.



(a) Mach number contour. 19 levels from 0.1 to 1.4. Sonic contour shown as ---.



(b) Vorticity contour. Non-dimensionalized as $\omega h/u_\infty$. 20 levels from -10.0 to 10.0.

Figure 7.10: Mach number and vorticity contours near the UCLA logo. First refinement zone edge ---. Second refinement zone edge ---.

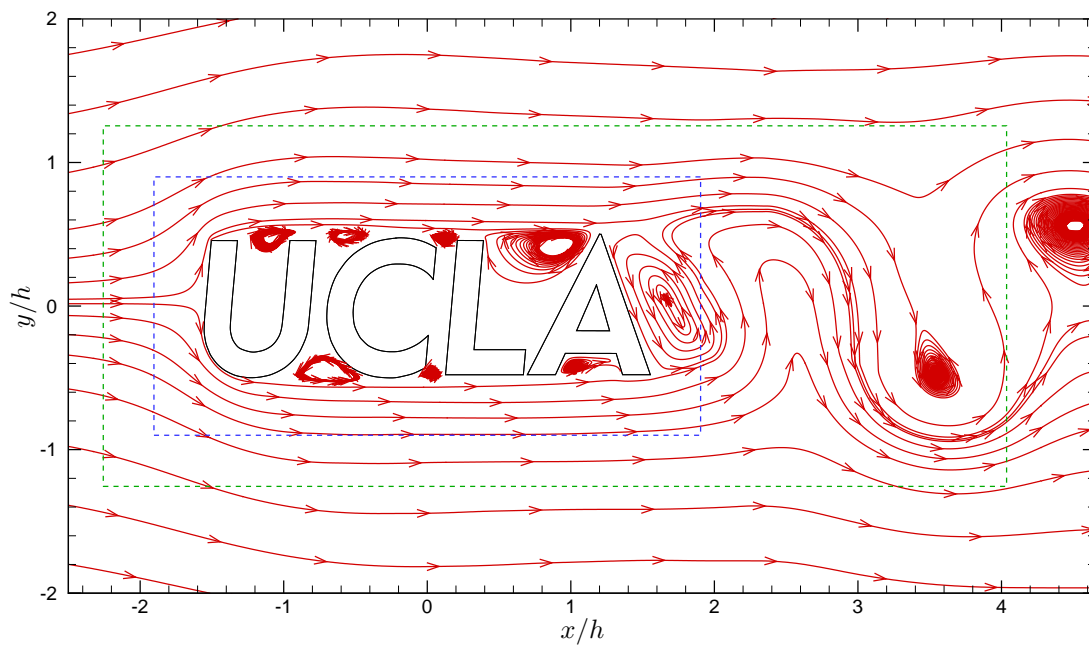


Figure 7.11: Streamlines near the UCLA logo. First refinement zone edge ---. Second refinement zone edge ---.

CHAPTER 8

Boeing/AFOSR Mach 6 Quiet Tunnel

The original goal of this work was to study the effects of isolated roughness elements on hypersonic boundary-layer flow by simulating flow inside the Boeing/AFOSR Mach 6 Quiet Tunnel (BAM6QT) at Purdue University. The wind tunnel at Purdue is a Ludwig tube capable of generating quiet Mach 6 flow and is maintained by Professor Steven Schneider [102]. By simulating flow in this geometry, simulations could be validated by comparing to the experimental results obtain by Prof. Schneider and his coworkers. Only simulations of flow inside the BAM6QT without a roughness element were performed.

8.1 Simulation Setup

A diagram of the Boeing/AFOSR Mach 6 Quiet Tunnel is shown in Figure 8.1. The main portion of the wind tunnel consists of a long driver tube followed by a converging-diverging nozzle and the test section. The tunnel is approximately 40 m long from the driver tube to the test section. A majority of the length of the tunnel is the driver tube which is 37.3 m long. The nozzle utilizes a bleed just before the throat. The bleed restarts the boundary layer which helps ensure quiet flow in the test section of the nozzle. After the test section is a diffuser, burst diaphragm, and vacuum tank.

The performed simulations are only concerned with a small portion of the total wind tunnel, specifically, from a small portion of the converging section to the end of the test section. Simulating the bleed is also beyond the interest of the current study. The bleed is modeled by using different boundary conditions along the wall. The section of the nozzle

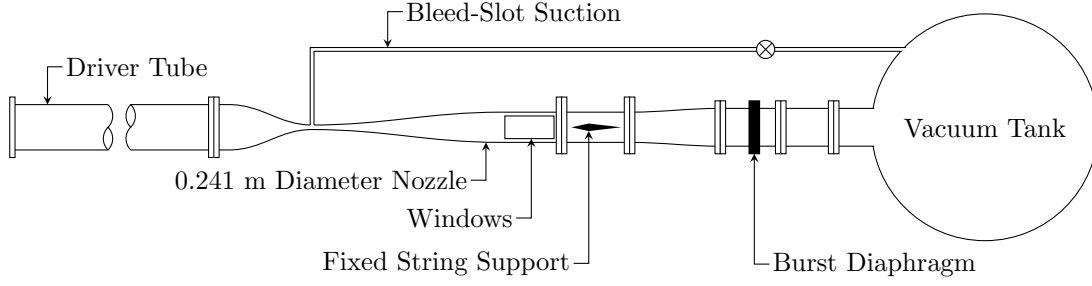


Figure 8.1: Diagram of Boeing/AFOSR Mach 6 Quiet Tunnel. Adapted from Schneider [102].

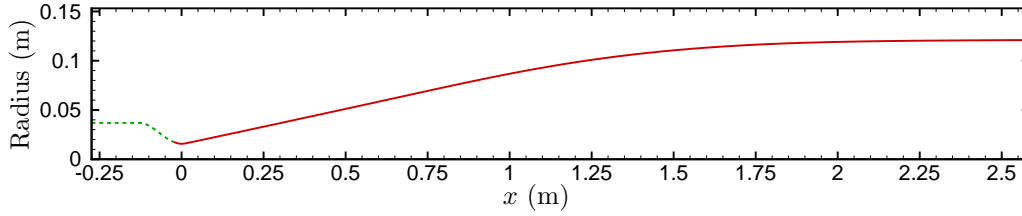


Figure 8.2: Portion of Boeing/AFOSR Mach 6 Quiet Tunnel being simulated. The vertical axis is enlarged by a factor of 3 compared to x axis. Actual BAM6QT physical wall —; Non-physical wall added to simulation ---.

used in the simulations is shown in Figure 8.2, where the flow goes from left to right. The converging section in the simulation is an arbitrary addition that does not match the BAM6QT geometry. The converging section of the BAM6QT opens to the driver tube where the flow velocity is nearly zero. If the converging section in the simulations matched the BAM6QT geometry, the simulations would take an unacceptable amount of time to converge due to the low inlet velocity. The non-physical converging section has an inlet velocity of approximately Mach 0.3. A straight portion was also added upstream of the converging section to simplify the implementation of the inlet boundary conditions of the simulations. The straight portion allows the inlet to be treated as one dimensional. These simulations were performed before the multi-zone cut-cell code was developed. At the time, a third-order curvilinear code was being used. The code is based on the scheme by Zhong [92].

The grid used in the simulations begins 0.27 m upstream of the nozzle's throat and ends 2.58 m downstream of the throat and uses 800 points in both the streamwise and radial directions. The grid is shown in Figure 8.3. The throat is located at $x = 0$ m and the bleed is 24.6 mm upstream of the throat. The grid is algebraically generated with clustering near

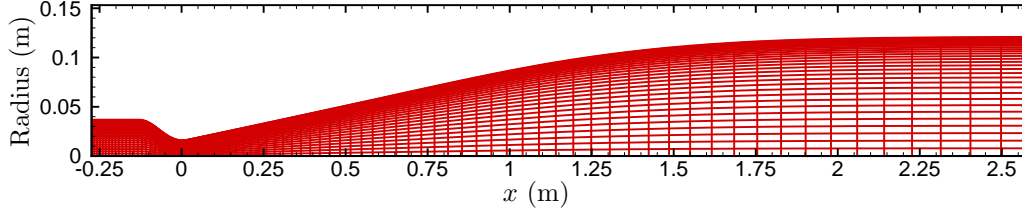


Figure 8.3: Grid for Boeing/AFOSR Mach 6 Quiet Tunnel simulation. The vertical axis is enlarged by a factor of 3 compared to x axis. Every 10th point in x direction shown. Every 25th point in y direction shown.

the walls and inlet. The clustering near the walls is present to resolve the thin boundary layer that is created. The clustering near the inlet assists in the capture of the “leading edge” created by the simulated bleed and the rapid change in flow properties at the throat.

The wall boundary conditions need to account for the bleed in the BAM6QT since the bleed is not being included in the numerical simulations. This is done because the bleed would require a significant increase in the number of grid points to accurately capture the flow around the bleed and would thus increase the required computational resources and time. Since the flow around the bleed is not the portion of interest in the current work, increasing the grid points there was not a viable option. The bleed was therefore simulated by changing the wall boundary conditions at the location of the bleed. Upstream of the bleed, inviscid slip boundary conditions were used. The slip boundary conditions impose no flow through at the wall while extrapolating the tangential velocity from the interior of the domain. Downstream of the bleed, no-slip boundary conditions are used.

A temperature profile, which is fixed in time, is imposed on the entire wall. The temperature profile comes from Schneider et al. [103]. Schneider et al. provide the wall temperature profile from the throat to about half way to the end of the test section. The profile was computed from a commercial finite-element heat-transfer program. Figure 8.4 shows the profile provided by Schneider et al. and the profile used in the current simulations. There was a small portion from the throat of the nozzle to the bleed which was not included in the results of Schneider et al. The temperature for this small portion was extrapolated from the data of Schneider et al. The temperature upstream of the bleed is constant and set to the

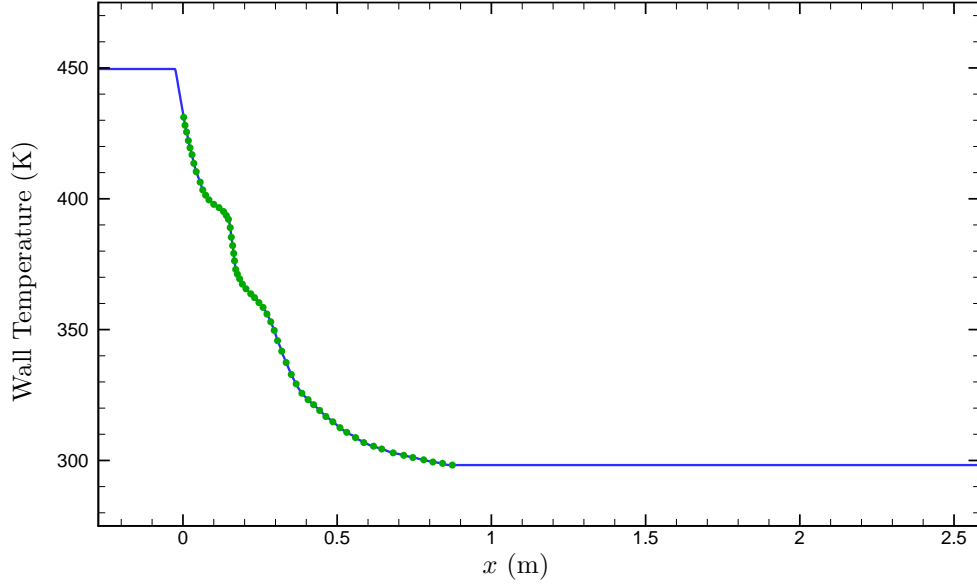


Figure 8.4: Wall temperature profile for the Boeing/AFOSR Mach 6 Quiet Tunnel. Profile used in simulations —; Profile from Schneider et al. [103] •.

extrapolated temperature at the bleed. The downstream half of the nozzle was not included in the simulations of Schneider et al. The simulations showed that the nozzle had reached the ambient temperature half way down the nozzle. Therefore, the downstream half of the wall was set to the ambient temperature.

A symmetry boundary condition is implemented on the conservative variables along the centerline of the nozzle. Since the flow at the inlet is subsonic and the walls have inviscid slip boundary conditions, inviscid characteristic boundary conditions [104] are applied at the inlet. The stagnation pressure and stagnation temperature are specified and the non-streamwise velocity components are set to zero. The inviscid characteristic equations are solved to calculate the streamwise velocity. From these, the isentropic relationships are used to calculate the static pressure and static temperature. At the outlet, the flow is mostly supersonic and therefore the conservative variables are extrapolated from the interior of the domain.

8.2 Results

Boundary-layer profiles are measured in the BAM6QT with a Pitot probe. During one run, the Pitot probe is moved perpendicular to the wall through the boundary layer. While the Pitot probe is being moved, the stagnation conditions are changing due to the nature of the BAM6QT. As a result, direct comparison with the results from one run of the BAM6QT to one numerical simulation is incorrect since the profile from the simulation is at constant stagnation conditions. Instead, two simulations were performed with the stagnation conditions at the time of the first and last measurement during one run in the BAM6QT.

Wheaton [105] provides a boundary-layer profile measured with a Pitot probe from a run with an initial stagnation pressure of 620 kPa. The stagnation pressure and temperature were 606.7 kPa and 430 K when the first Pitot probe measurement was taken and 530.9 kPa and 415 K when the last measurement was taken. Two simulations were performed using these stagnation values. When a Pitot probe is placed inside supersonic flow, a bow shock will form in front of the probe. Therefore, the pressure measured by the probe is the stagnation pressure downstream of the shock. However, the pressure measured in the simulation is the static pressure of the flow. Therefore the pressure from the simulation must be converted into post shock stagnation values before comparing to the measure values of Wheaton. The compressible isentropic flow and normal shock relations were used to convert the static pressure from the simulations to the stagnation pressure downstream of a shock at the probe location.

Figure 8.5 shows a comparison between the measured Pitot pressures from the BAM6QT and the computed values from the two simulations. The results were checked for grid convergence by performing the same simulations on a grid with 400×400 points. There was no noticeable difference in the results for the different grids. The simulation results show good agreement with the experimental measurements.

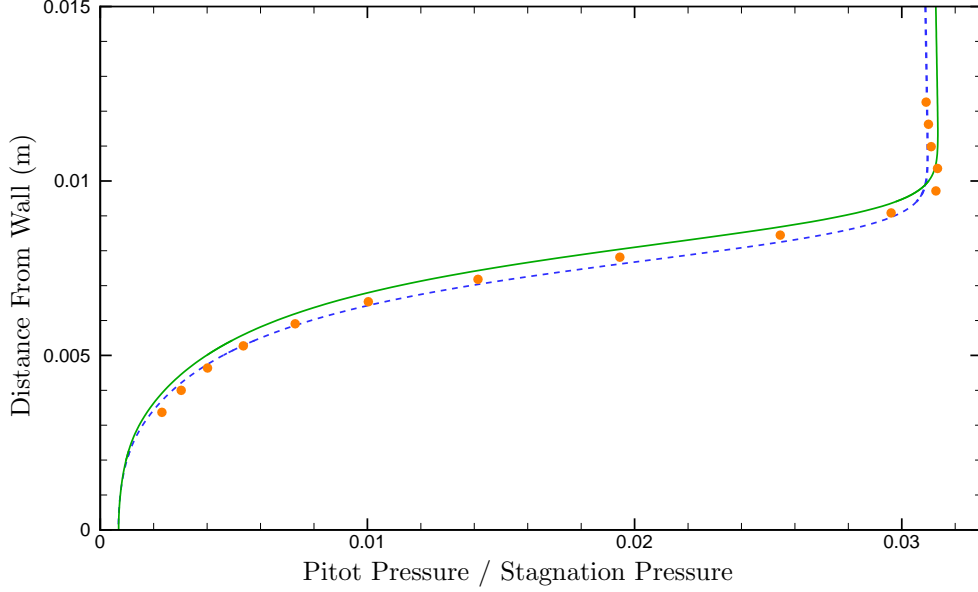


Figure 8.5: Comparison of Pitot pressure profile from the Boeing/AFOSR Mach 6 Quiet Tunnel at $x = 1.93$ m. Simulation results with $p_0 = 606.7$ kPa ---; Simulation results with $p_0 = 530.9$ kPa —; Measurements from Wheaton [105] •.

8.3 Geometry Change

A number of difficulties arose while attempting to simulate flow in the BAM6QT. Many of these difficulties have been mentioned in the previous sections of this chapter. These include the changing stagnation conditions in the BAM6QT during one run, modeling the bleed, the missing portions of the wall temperature profile, and the use of a Pitot probe to measure boundary-layer profiles. Changing the boundary conditions along the wall to model the bleed does start the boundary layer at the correct location, but any changes that the bleed causes to the stagnation conditions or the mass flow rate are not being accounted for. The method used to generate the missing portions of the wall temperature profile introduces kinks in the profile at the bleed and where the wall temperature is set to ambient. The shock in front of the Pitot probe will be a finite distance away from the probe and will change as the Mach number increases in the boundary layer. This distance cannot be accounted for in the Pitot pressures calculated from the simulation results. These issues bring uncertainties into the results of the simulations. Consequently, the source of any differences between the numerical

and experimental results would be difficult to identify.

In addition to these problems, more arose when attempting to simulate flow over a roughness element in the BAM6QT. The major issue was in the generation of a smooth grid. At the time that these simulations were performed, only a curvilinear code was available so smooth body-fitted grids were required to perform simulations. Both the commercial grid generation software Gridgen and an elliptical grid generation code written by the author were not able to generate a sufficiently smooth grid. Both had difficulties with the combination of the axisymmetric geometry and an isolated roughness.

Due to the uncertainties of the simulations without a roughness element and the difficulties in generating a smooth grid with a roughness element, it was decided to switch to a flat plate with an isolated roughness element. Flat plates have been numerically simulated for an extensive period of time so there are many sources that can be used to validate the code. In addition, the elliptical grid generation code had no difficulties in generating smooth grids for flat plates with isolated roughness elements. Although the geometry has been changed to a flat plate, the flow conditions and roughness parameters for the simulations will still come from the experiments done in the BAM6QT at Purdue University. This will still allow for comparison to their experiments.

CHAPTER 9

Isolated Roughness Element

This chapter presents results from simulations of Mach 6 flow over isolated roughness elements. As mentioned in Chapter 8, the original intention of this work was to simulate flow inside the BAM6QT at Purdue University. However, due to the previously mentioned difficulties, the BAM6QT geometry was abandoned. A simple flat plate is used to model the wind tunnel. Although the geometry has changed, the same flow conditions as the Purdue wind tunnel have been used.

Wheaton and Schneider [27] have recently measured a 21 kHz instability in Pitot probe and hot-wire measurements for Mach 6 flow over a cylindrical roughness element mounted to the wall of the wind tunnel's nozzle. The cylinder had a height of 10.2 mm and a diameter of 5.97 mm. The undisturbed boundary layer had a thickness of approximately 9.23 mm. This gives a roughness height to boundary-layer thickness ratio (k/δ) of 1.10 and a diameter to boundary-layer thickness ratio (D/δ) of 0.647. Ward et al. [106] measured the same instability with flush-mounted pressure transducers on the nozzle wall. The goal of the simulations in this Chapter was to reproduce these results and to investigate the source of the instability. The flow conditions used in the simulations came from Wheaton and Schneider and are given in Table 9.1. The wall temperature is the only value that did not come from Wheaton and Schneider. The value is the wall temperature in the test section from Figure 8.4. Two roughness geometries were used in the simulations: Gaussian and cylindrical.

Freestream Mach number	M_∞	5.93
Freestream velocity	u_∞	873 m/s
Freestream pressure	p_∞	0.37 kPa
Freestream temperature	T_∞	53.9 K
Stagnation pressure	p_0	551.58 kPa
Stagnation temperature	T_0	433 K
Wall temperature	T_{wall}	298.2 K
Freestream unit Reynolds number	Re_∞	$5.94 \times 10^6 \text{ m}^{-1}$
Undisturbed boundary-layer thickness	δ	9.23 mm

Table 9.1: Flow conditions for isolated roughness simulations.

9.1 Gaussian Roughness

The first set of results to be presented are for a three-dimensional Gaussian shaped roughness.

The shape of the roughness is given as

$$h(x, z) = h_0 \exp\left(-\frac{x^2 + z^2}{r_0^2}\right) \quad (9.1)$$

where h_0 is the roughness height and r_0 controls the roughness radius. Four different roughness heights were simulated. The tallest element has a roughness height to boundary-layer thickness ratio (k/δ) of 1.10 and is the same height as the roughness from Wheaton and Schneider. The remaining roughness elements have k/δ ratios of 0.75, 0.5 and 0.25. The shortest roughness element is completely submerged in the subsonic portion of the undisturbed boundary layer. r_0 is set to 0.498δ for all the roughnesses. Figure 9.1 shows a comparison of the roughness elements used in the simulations. Also shown is the cylindrical roughness used by Wheaton and Schneider. The height of the undisturbed boundary layer and the location where the Mach number is one (sonic line) are also shown. The Reynolds number based on the roughness height and the local conditions in the undisturbed boundary layer ($Re_k = \rho u k / \mu$) for each roughness are listed in Table 9.2.

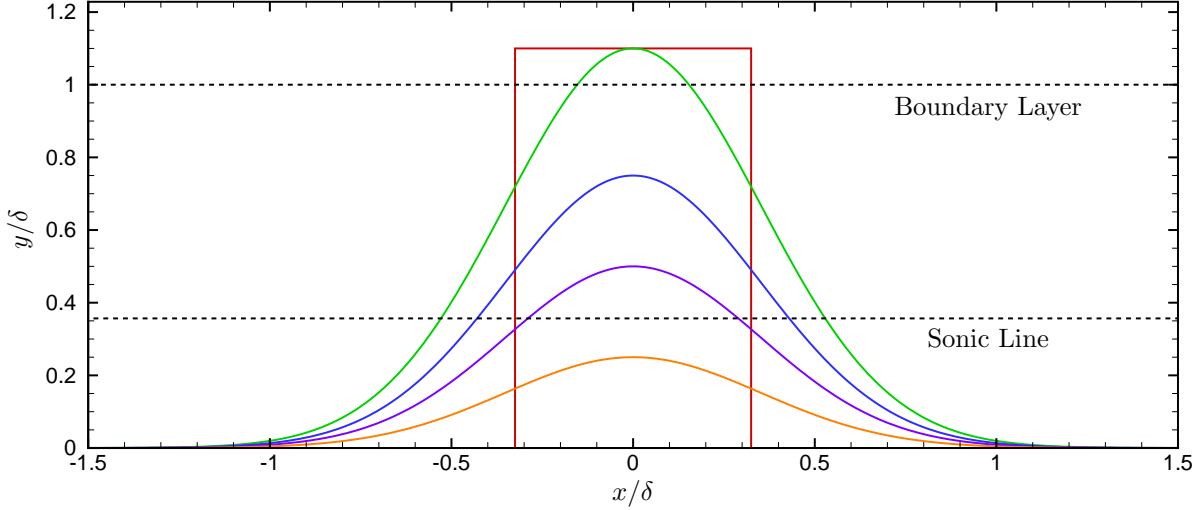


Figure 9.1: Comparison of roughness elements used in simulations. Profiles are shown at $z/\delta = 0.0$. $k/\delta = 1.10$ Gaussian —; $k/\delta = 0.75$ Gaussian —; $k/\delta = 0.50$ Gaussian —; $k/\delta = 0.25$ Gaussian —; $k/\delta = 1.10$ cylinder —.

Roughness height (k/δ)	Roughness Reynolds Number (Re_k)
0.25	130
0.50	680
0.75	3,330
1.10	57,600

Table 9.2: Reynolds number based on roughness height and local undisturbed conditions.

9.1.1 Simulation Setup

The setup for the simulations done here is very similar to the three-dimensional roughness element from Chapter 6. An isolated roughness element is placed on a flat-plate with boundary-layer flow over it. The origin is located at the center of the roughness element. The x axis is in the streamwise direction, the y axis is in the wall normal direction, and the z axis is in the spanwise direction. The numerical domain starts 20δ upstream of the roughness and ends 25δ downstream of the roughness. The domain has a height and half width of 5δ each. The same sponge layer and boundary conditions as the three-dimensional roughness element from Chapter 6 are used here except the wall is now isothermal. An isothermal wall was used in these simulations because the run time of the BAM6QT is only a few seconds.

It is unlikely that the wall will reach an adiabatic temperature in such a short time.

These simulations were performed before the development of the multi-zone cut-cell code. The simulations were performed with a single-grid curvilinear code. Two different grid generation techniques were used in the simulations. For the three shorter roughness elements, the grids were generated algebraically. Figure 9.2 shows the grid in the vicinity of the roughness with $k/\delta = 0.75$. The grid is clustered around the roughness element in the x and z directions and near the wall in the y direction. Only half of the domain in the z direction is simulated. The expressions used to generate the grid are

$$x = (x_c - x_{in}) \left(1 + \frac{\sinh(a_1)}{\sinh(a_2)} \right) + x_{in}, \quad (9.2)$$

$$y = (H - h(x, z)) \frac{\sinh(S_\eta \eta)}{\sinh(\eta)} + h(x, z), \quad (9.3)$$

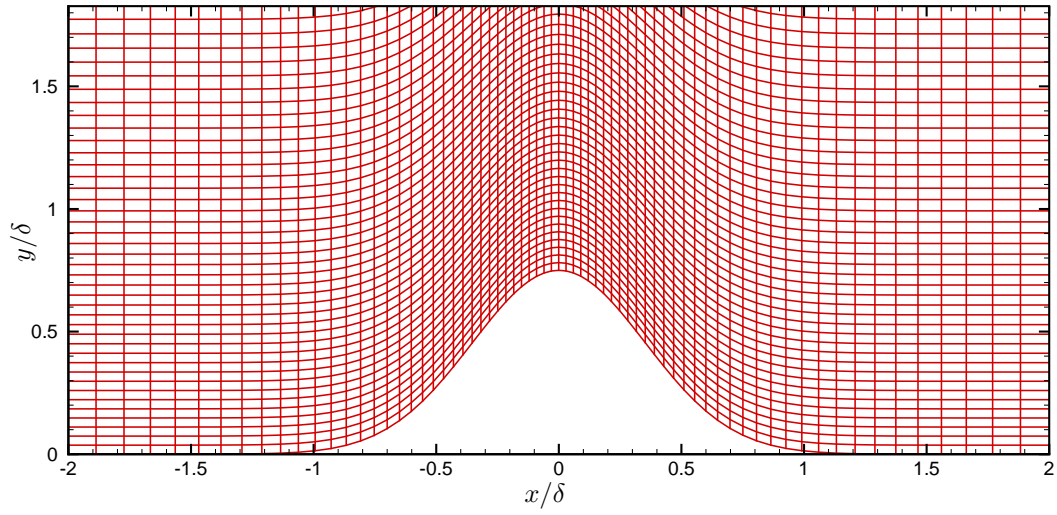
$$z = W \frac{\sinh(S_\zeta \zeta)}{\sinh(\zeta)}, \quad (9.4)$$

where

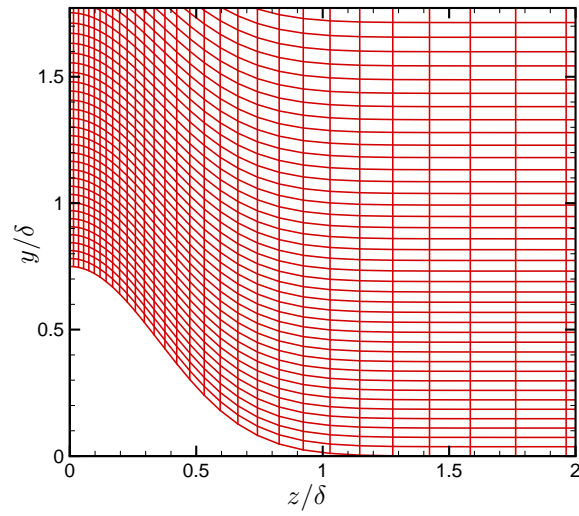
$$\begin{aligned} a_1 &= S_\xi \xi - \frac{1}{2} \log \left(\frac{b_+}{b_-} \right), \\ a_2 &= \frac{1}{2} \log \left(\frac{b_+}{b_-} \right), \\ b_\pm &= 1 - \left(\exp(\pm S_\xi) - 1 \right) \frac{x_{in}}{x_{out} - x_{in}}. \end{aligned} \quad (9.5)$$

x_{in} and x_{out} are the x locations of the domain inlet and outlet, respectively. H is the domain height and W is the domain half width. S_ξ , S_η , and S_ζ control the amount of clustering in the x , y and z directions, respectively. The same expressions are used for the shorter roughnesses except the roughness height is changed in $h(x, z)$. For all the simulations $S_\xi = 9$, $S_\eta = 2$, and $S_\zeta = 4$.

The grid for the $k/\delta = 1.10$ roughness element was first generating using the same equations. When that grid was used, the simulation became numerically unstable. The $k/\delta = 1.10$ roughness element caused the equations to generate very skewed grid cells which

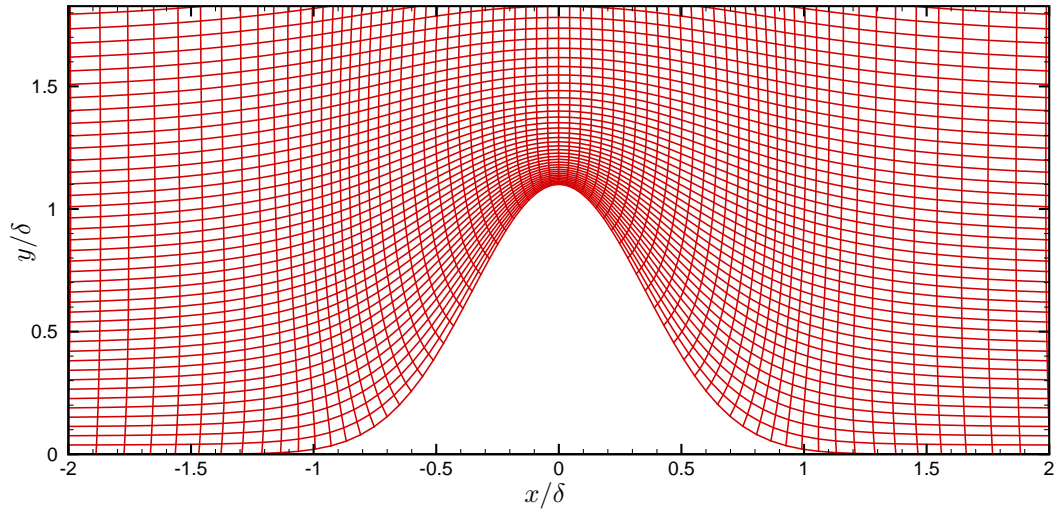


(a) $x - y$ plane at $z/\delta = 0$.

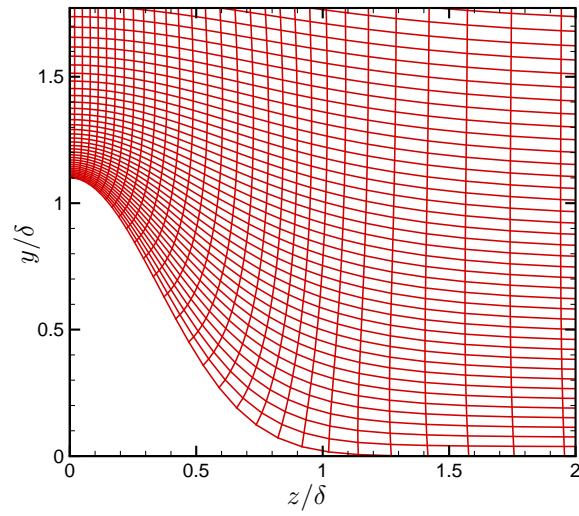


(b) $y - z$ plane at $x/\delta = 0$.

Figure 9.2: Grid in the vicinity of the three-dimensional Gaussian roughness element with $k/\delta = 0.75$. Every fourth point shown in all directions.



(a) $x - y$ plane at $z/\delta = 0$.



(b) $y - z$ plane at $x/\delta = 0$.

Figure 9.3: Grid in the vicinity of the three-dimensional Gaussian roughness element with $k/\delta = 1.10$. Every fourth point shown in all directions.

appeared to be the cause of the numerical instability. As a result, an orthogonal grid was generated using the method of Spekrijse [107]. Spekrijse outlined an elliptical grid generation method for creating smooth orthogonal grids. The only necessary input is the grid point distribution along the two-dimensional sides of the three-dimensional domain. These distributions came from Equations 9.2 to 9.4 with the same values for S_ξ , S_η , and S_ζ . Both sides of the roughness element were generated and initially used in the simulation. This was done because the instability measured by Wheaton and Schneider may generate an asymmetric flow over the roughness element. After running the simulation for sometime, the flow remained symmetric about the $z = 0$ plane. Therefore, to save computational time, half of the domain was removed for the remainder of the simulation. Figure 9.3 shows the grid in the vicinity of the roughness element. All the grids used 600 points in the streamwise direction, 300 points in the wall normal direction, and 150 points in the spanwise direction. Away from the roughness element, the grids used for each of the roughness elements match. The results were not confirmed to be grid converged. A grid convergence study would be too computationally expensive to perform.

9.1.2 Results

The simulation for each of the roughness heights reached a steady-state solution. Figure 9.4 shows density contours at the $z = 0$ symmetry plane for the four roughness elements. The $k/\delta = 1.10$ roughness clearly shows a shock generated slightly upstream of the roughness. Just downstream of the shock is an expansion fan which is followed by a weaker shock. The same structures are also visible in the shorter roughness heights, but they become weaker as the roughness height decreases. The contours for the $k/\delta = 1.10$, 0.75 and 0.50 roughnesses show that the boundary layer has been significantly altered downstream of the roughness element. The boundary layer downstream of the $k/\delta = 0.25$ roughness element was not as greatly affected by the roughness.

Figure 9.5 shows a three-dimensional isosurface of the vorticity magnitude, $|\omega|$, for each roughness. The $|\omega|\delta/u_\infty = 1.5$ surface is shown. The surface is colored by the streamwise

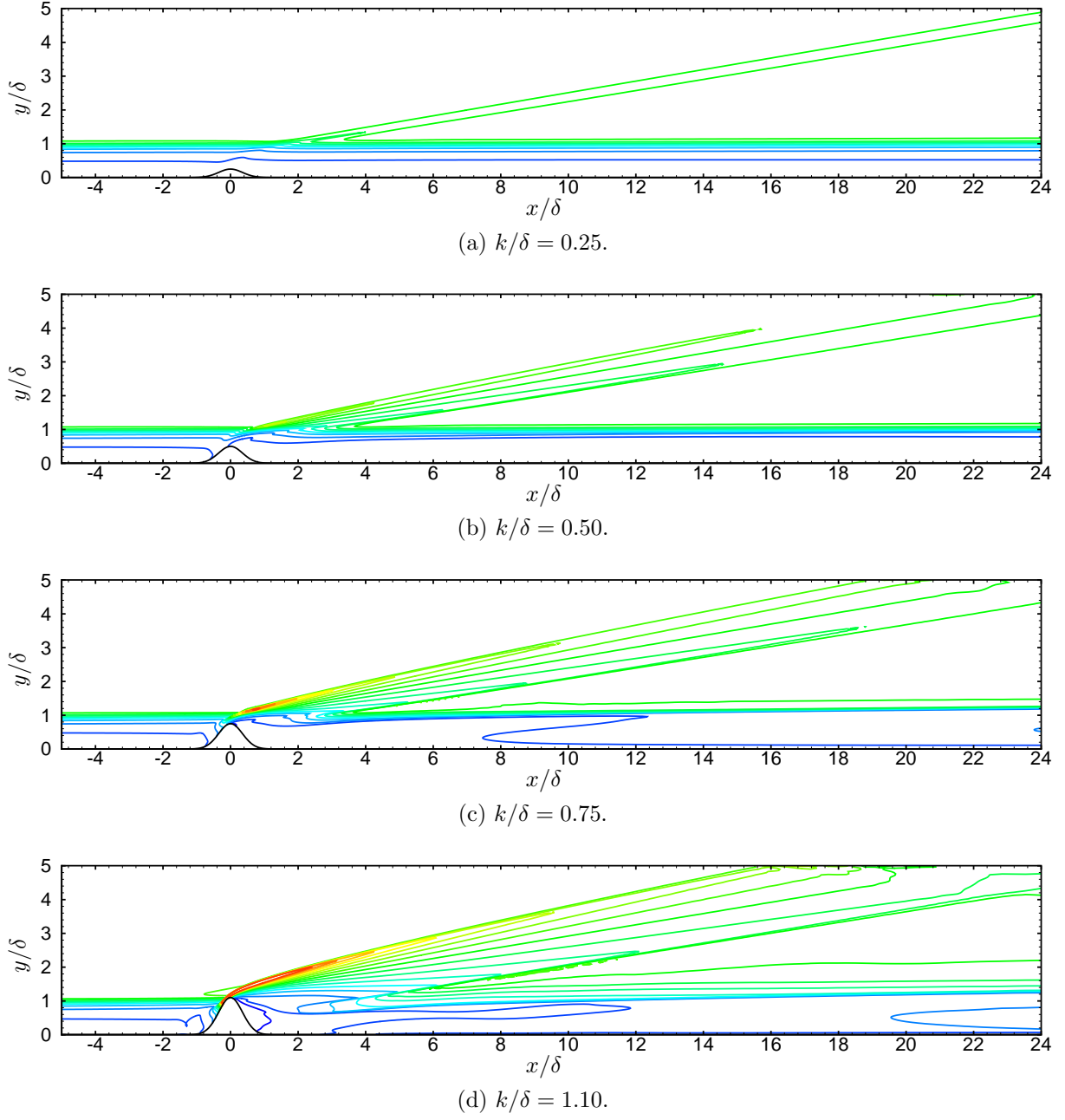
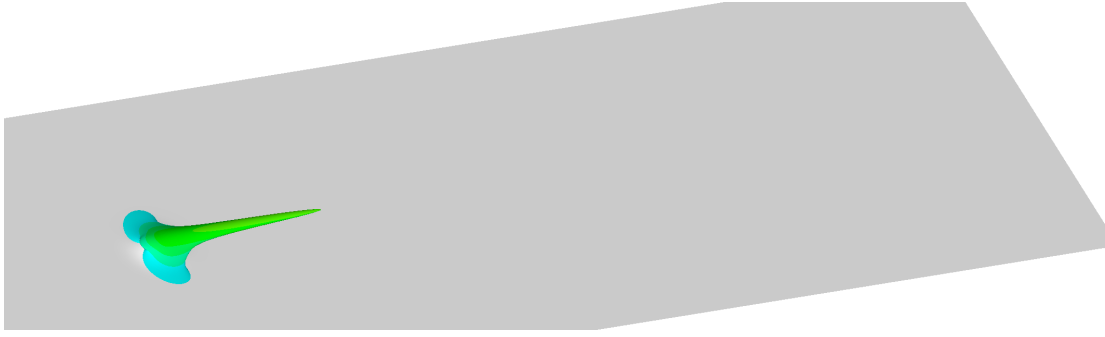


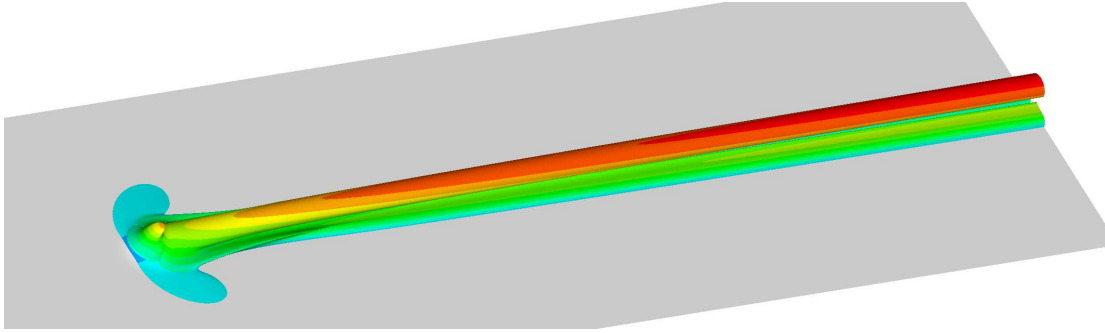
Figure 9.4: Density contours at $z = 0$ plane for Gaussian roughness simulations. Non-dimensionalized as ρ/ρ_∞ . 16 contour levels from 0.1 to 1.75.

velocity. The main features for the $k/\delta = 1.10$ roughness are the horse-shoe vortex and the wake vortex. The horse-shoe vortex is the two outer streaks seen in the figure. It wraps around the front of the roughness element and then aligns itself with the flow direction downstream of the roughness element. The wake vortex is formed directly behind the roughness element. In the figure, it is the long streak located between the legs of the horse-shoe vortex. There is also a high vorticity region along the wall underneath the wake vortex. All the structures are straight and persist until the exit of the numerical domain. For the $k/\delta = 0.75$ roughness, the same structures are present but the legs of the horse-shoe vortex are smaller. For the $k/\delta = 0.50$ roughness, the horse-shoe vortex is no longer visible in the figure. The wake vortex and the high vorticity region near the wall are still present. For the $k/\delta = 0.25$ roughness, only a small portion of the wake vortex is visible.

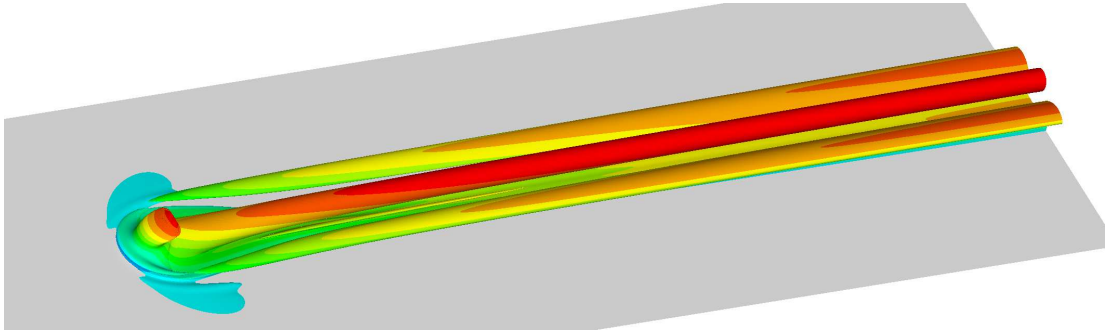
Figure 9.6 show contours of vorticity magnitude at various x/δ locations for the four roughness elements. The contours for $k/\delta = 1.10$ show a large circular vortex in the center of the domain and a kidney-shaped vortex off to the side of the roughness. The circular vortex is the wake vortex while the kidney-shaped vortex is the leg of a horse-shoe vortex. Both vortices persist until the end of the numerical domain. At $x/\delta = 2$, the top of the wake vortex is at approximately $y/\delta = 0.8$ which is less than the height of the roughness. The vortex moves vertically as it progresses downstream. At $x/\delta = 24$, the top of the vortex has reached $y/\delta = 1.3$. The horse-shoe vortex also moves vertically as it progresses downstream but the increase is much less than the wake vortex. However, the size of the horse-shoe vortex increases greatly. The vortex approximately doubles in length within the numerical domain. The two vortices are also present in the contours for the $k/\delta = 0.75$ roughness but both are much weaker in magnitude. At $x/\delta = 2$, the top of the wake vortex is approximately the same height as the roughness element. The vortex also moves vertically like the wake vortex from the $k/\delta = 1.10$ roughness but the increase is less. The horse-shoe vortex also increases in length but to a greater extent than the $k/\delta = 1.10$ roughness. For the contour levels used in the figure, only the wake vortex is visible for the $k/\delta = 0.50$ roughness. The top of the wake vortex is higher than the roughness element at $x/\delta = 2$ with an approximate



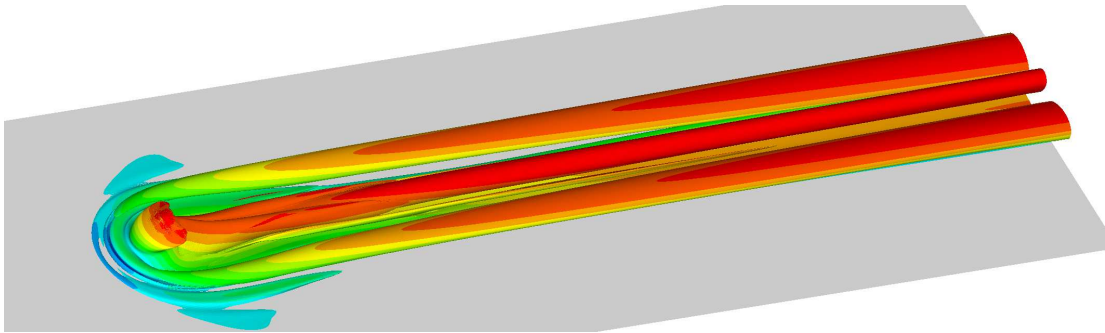
(a) $k/\delta = 0.25$



(b) $k/\delta = 0.50$



(c) $k/\delta = 0.75$



(d) $k/\delta = 1.10$

Figure 9.5: Vorticity magnitude isosurfaces for Gaussian roughness simulations. Surface for $|\omega|\delta/u_\infty = 1.5$ shown. Colored by u/u_∞ with 12 levels from -0.2 to 0.9.

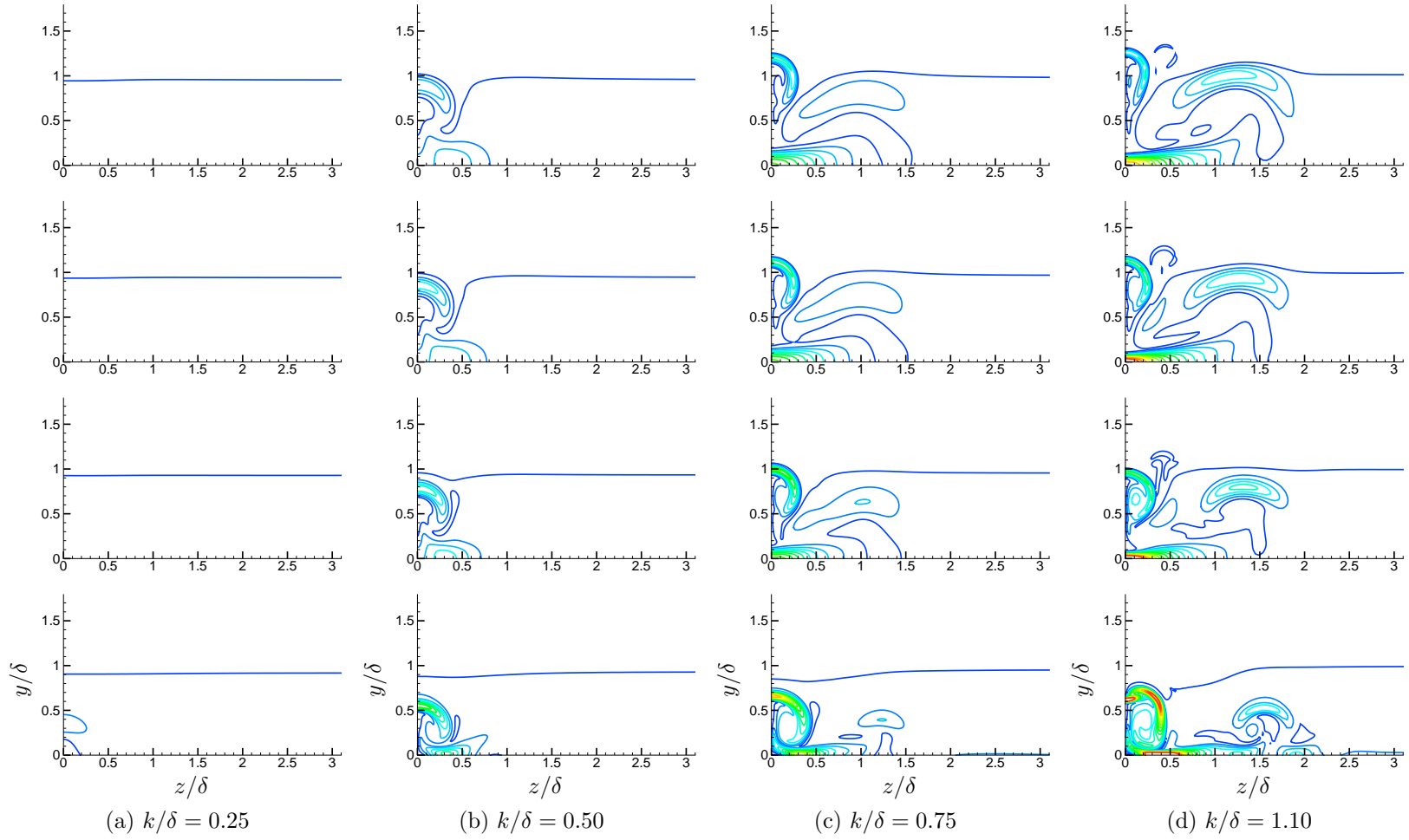


Figure 9.6: Vorticity magnitude contours for Gaussian roughness simulations at various x locations. Non-dimensionalized as $|\omega|\delta/u_\infty$. 16 contour levels from 0 to 10. Starting from the bottom and moving vertically, the contours are at $x/\delta = 2, 9.3, 16.6$, and 23.9 .

height of $y/\delta = 0.7$. The vortex moves vertically as it progresses downstream but appears to asymptote to approximately $y/\delta = 1.0$. For the $k/\delta = 0.25$ roughness, there is a very small wake vortex seen at $x/\delta = 2$ but it cannot be seen with these contour levels anywhere downstream.

9.2 Cylindrical Roughness

The multi-zone cut-cell code permits simulations with a cylindrical roughness element to be performed. Flow over a cylindrical roughness element with the same dimensions as the one used by Wheaton and Schneider was simulated. The roughness height to boundary-layer thickness ratio (k/δ) was 1.10 and the diameter to boundary-layer thickness ratio (D/δ) was 0.647. The roughness Reynolds number was the same as the $k/\delta = 1.10$ Gaussian ($Re_k = 57,600$).

9.2.1 Simulation Setup

The setup for this simulation is nearly identical to the Gaussian roughness simulations. The domain starts at $x/\delta = -20$ and ends at $x/\delta = 25$. The top of the domain is located at $y/\delta = 5$ and the domain half width is $W/\delta = 7.5$. The same boundary conditions and sponge layer are used in this simulation too. A coarse grid and two refinement zones were used for this simulation. The grid in the vicinity of the roughness is shown in Figure 9.7. The coarse grid has 200 points in the x direction, 140 points in the y direction, and 121 points in the z direction. The grid is clustered near the front of the roughness in the x direction, near the top of the roughness in the y direction, and near the center of the roughness in the z direction. Unlike the Gaussian roughness element simulations, both sides of the domain in

S_ξ	S_η	S_ζ	ℓ_η	P_η
9.2	4.099	3.5	0.71	1.07136061

Table 9.3: Grid constants for cylindrical roughness.

the z direction were simulated. The grid was generated with the following equations:

$$x = (x_c - x_{in}) \left(1 + \frac{\sinh(a_1)}{\sinh(a_2)} \right) + x_{in}, \quad (9.6)$$

$$y = H \left(A_\eta + |\eta|^{P_\eta} (B_\eta - A_\eta) \right), \quad (9.7)$$

$$z = W \frac{\sinh(S_\zeta(2\zeta - 1))}{\sinh(S_\zeta)}, \quad (9.8)$$

where

$$A_\eta = \ell_\eta \eta, \quad (9.9)$$

$$B_\eta = \frac{\sinh(S_\eta \eta)}{\sinh(S_\eta)}, \quad (9.10)$$

$$a_1 = S_\xi \xi - \frac{1}{2} \log \left(\frac{b_+}{b_-} \right), \quad (9.11)$$

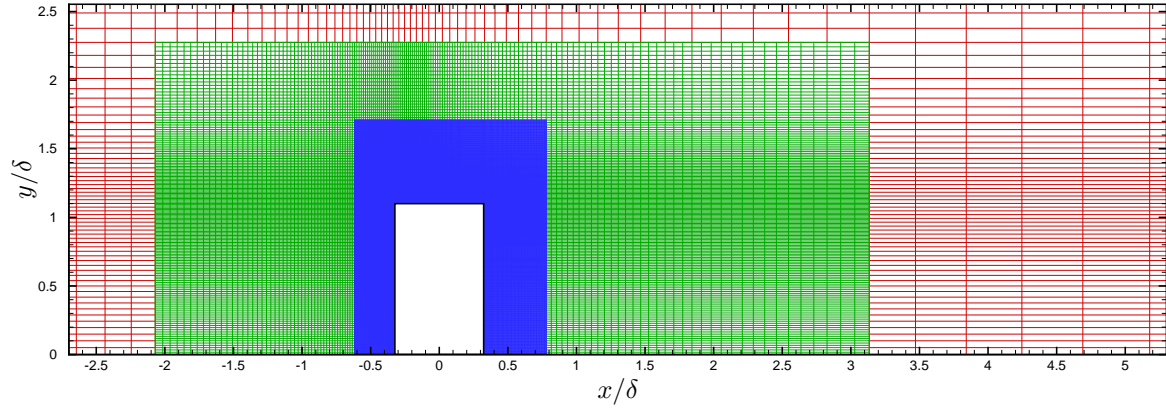
$$a_2 = \frac{1}{2} \log \left(\frac{b_+}{b_-} \right), \quad (9.12)$$

$$b_\pm = 1 + (\exp(\pm S_\xi) - 1) \frac{x_c - x_{in}}{x_{out} - x_{in}}. \quad (9.13)$$

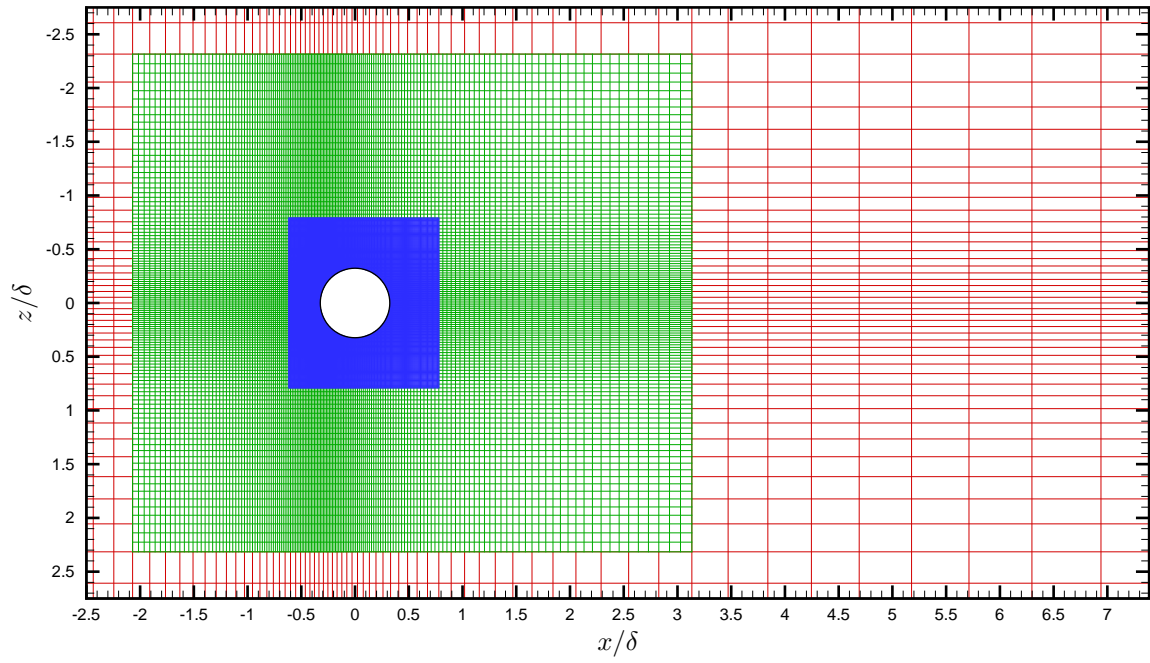
The value for the constants used in the expressions are given in Table 9.3.

9.2.2 Results

Figure 9.8 shows an isosurface of the Q-criterion. The surface is colored by the streamwise velocity. Unlike the Gaussian roughness elements, the flow has become unsteady. However, the flow field has not reached a statistically stationary state. The flow downstream of the roughness element is still changing significantly with time. Since the entire flow field has not reached a statistically stationary state, a frequency analysis of the instability has not been



(a) $x - y$ plane at $z/\delta = 0$.



(b) $x - z$ plane at $y/\delta = 0$.

Figure 9.7: Grid in the vicinity of the three-dimensional cylindrical roughness element. Points inside the roughness element are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue. Only every second point shown in all directions.

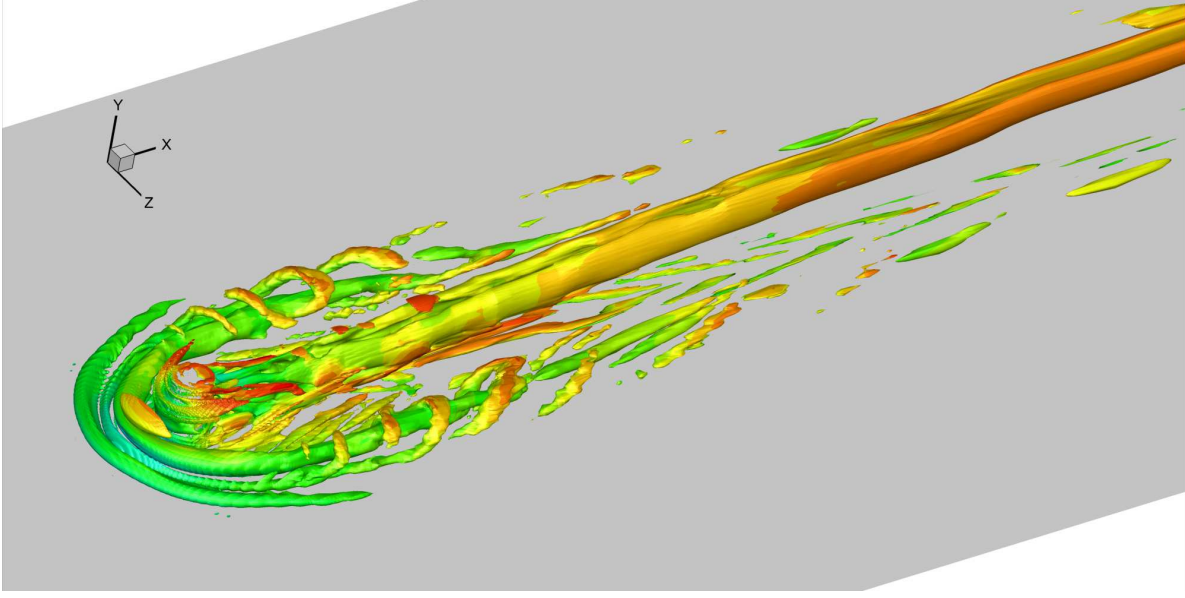
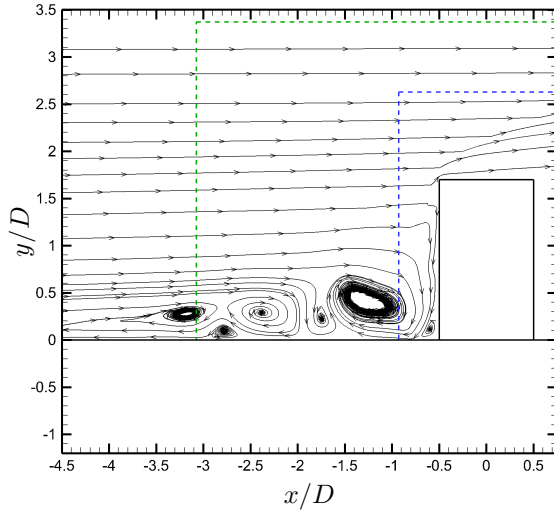


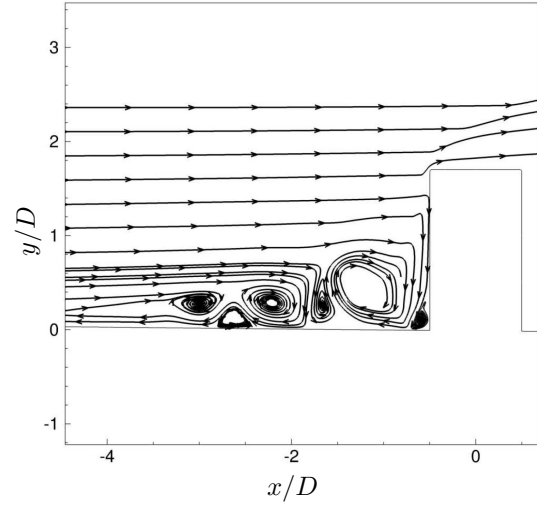
Figure 9.8: Isosurface of Q-criterion for cylindrical roughness simulation. Surface for $Q \delta^2 / u_\infty^2 = 0.2$ shown. Colored by u/u_∞ with 15 levels from -0.5 to 0.9.

performed. However, the flow near the roughness element is no longer undergoing large scale changes. This allows preliminary results to be compared with previously published work from other researchers.

Figure 9.9 shows a comparison of the streamlines in front of the roughness element for the current simulation and a simulation performed at similar condition from Wheaton et al. [108]. The vortical structures from the two simulations seem to match very well. Figure 9.10 shows the surface streamlines and wall pressure for the current simulation and a simulation performed by Bartkowicz et al. [26]. The streamlines show the separation and re-attachment lines created by the vortices shown in Figure 9.9. Upstream of the roughness, the results in Figure 9.10 appear very similar. To the sides and downstream of the roughness, the streamlines begin to show differences. This may be due to transient effects from the initial condition for the current simulation. Since the simulation has not reached a statistically stationary state, the initial transients are still present.

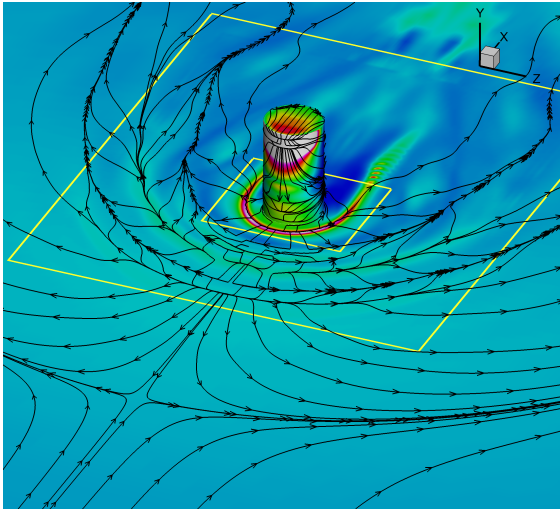


(a) Current simulation. First refinement zone edge ---; Second refinement zone edge ---.

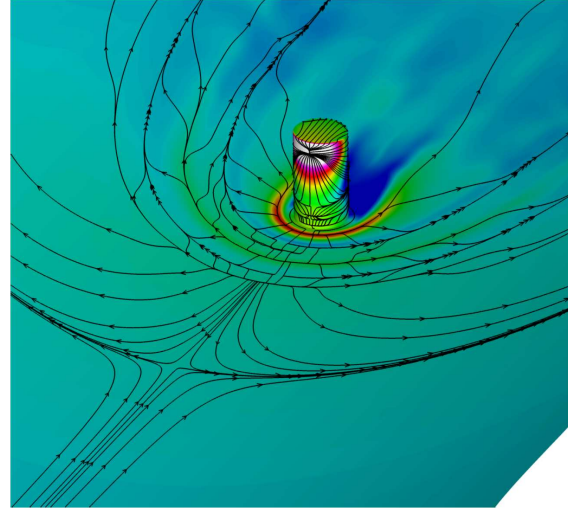


(b) Results from Wheaton et al. [108].

Figure 9.9: Comparison of streamlines in $z/\delta = 0$ plane for flow over cylindrical roughness.



(a) Current simulation. Pressure plotted as p/p_∞ with contour levels varying from 0.4 to 5.0. Refinement grid edges shown in yellow.



(b) Results from Bartkowicz et al. [26].

Figure 9.10: Comparison of surface streamlines and pressure contours for flow over cylindrical roughness.

CHAPTER 10

Conclusion

Results were presented for a new code designed to perform high-order simulations of high-speed flows over complex geometries. The code solves the compressible Navier-Stokes equations. To account for discontinuities in the solution, a hybrid WENO scheme was used for the inviscid terms in the Navier-Stokes equations. The hybrid scheme combines a fifth-order WENO with a low-dissipation finite-difference scheme. Simulations are carried out on Cartesian grids with the geometries imposed by the cut-cell method. A multi-zone refinement method is also utilized to provide extra control over the placement of grid points. The cut-cell method was confirmed to provide globally fourth-order convergence when combined with the codes fifth-order interior scheme. It was also shown that the multi-zone method does not reduce the order of the interior scheme. The combination of the cut-cell and multi-zone refinement methods should result in a globally fourth-order scheme.

The code was validated by simulating subsonic flow over a cylinder and supersonic flow over an isolated roughness element constructed from hyperbolic tangents. The cylinder simulation was an unsteady test for the code. The Reynolds number for the flow placed the solution in the periodic vortex shedding regime. The frequency of the vortex shedding matched very well with experimental results. The purpose of the isolated roughness simulation was to evaluate the code on a problem similar to the problem of interest in this work. The use of a hyperbolic tangent allowed the simulation to also be performed on a single body-fitted grid. The results from the body-fitted simulation and the multi-zone cut-cell simulation matched. The results from these test cases suggest that the code is working correctly. In addition, the multi-zone method provided a significant reduction in runtime for

the hyperbolic roughness case compared to the single body-fitted simulation.

To demonstrate the capabilities of the new code, two simulations of over flow over complex geometries were presented. The first was Mach 3 flow over the logo of the Ubuntu Linux distribution and the second was Mach 0.8 flow over the logo for the University of California, Los Angeles. There are no previous results to compare with for these simulations. However, the results for the simulations do not appear to have any obvious errors.

The motivation for the development of the code presented in this work was to study the effects of isolated roughness elements on the stability of hypersonic boundary layers. Specifically, recent results from experiments performed in the Mach 6 wind tunnel at Purdue University were to be reproduced and studied. The experiments measured a 21 kHz instability in the wake of a cylindrical roughness element placed on the wall of the wind tunnel. The goal was to reproduce this instability and identify its source.

The study of the Purdue wind tunnel began by simulating flow in the nozzle of the wind tunnel without a roughness element. These simulations were performed with an older version of the code which only used a single body-fitted grid. The results from this base flow simulation compared well with boundary-layer measurements made at Purdue University. Although the results compared well, the methods used to simulated the flow could bring uncertainties to future results. Consequently, the source of any future differences between the experimental and numerical results would be difficult to identify. To address this issue, the nozzle geometry was replaced with a flat plate. Boundary layer flows over flat plates have been extensively studied so the base flow can be confidently verified before adding a roughness element. The flow conditions used for the simulations were still based on the Purdue experiments.

Preliminary results for a study of the effects of isolated roughness elements on hypersonic boundary-layer flow was also presented. A set of simulations were performed for flow over a Gaussian shaped roughness element. The height of the roughness was changed for each simulation. The shortest roughness was only 25% of the boundary-layer thickness. The tallest was 110% of the boundary-layer thickness which is equal to the roughness height

used in the experiments at Purdue University. These simulations were performed on a single body-fitted grid using an older version of the code. All the simulations reached a steady-state solution. Results for flow over a cylindrical roughness matching the roughness used by Purdue University were presented next. The flow become unsteady but the solution has not reached a statistically stationary state, so the frequency of the instability could not be measured. However, the flow near the roughness element is no longer undergoing large scale changes so the solution could be compared to previously published results by other researchers. The flow near the roughness compared very well with the published results.

The results presented here have demonstrated the new code's ability to handle flows over complex geometries. The cut-cell method allows arbitrary geometries to be simulated on simple Cartesian grids while the multi-zone method can provide significant reductions in runtime. The combination is a promising tool for the study of roughness-induced transition in hypersonic boundary layers.

10.1 Future Work

Although the code has been able to simulate flow over a variety of geometries, there are still some issues with the solver. The primary issue is runtime. Even with the multi-zone method, some problems can require substantial computational time. One example is the boundary-layer flow over a cylindrical roughness presented in Chapter 9. The large amount of computational resources required by this simulation prevented it from being advanced to a statistically stationary state. There are two promising approaches for further reducing the runtime: a semi-implicit scheme and the adaptive mesh refinement method.

The Navier-Stokes solver currently uses an explicit Runge-Kutta scheme for time advancement. For unsteady problems, this is an appropriate time advancement scheme as long as the solver's time step is restricted by the convective time scale. When using the cut-cell method, this is sometimes not the case. The cut-cell method tends to require a high spatial resolution near boundaries to maintain numerical stability. This high resolution results in

small grid spacing which can cause the viscous time scale to restrict the solver’s time step size. This occurred in the cylindrical roughness simulation. The convective time step for that simulation was approximately four times larger than the viscous time step. This issue can be rectified through the use of a semi-implicit time advancement scheme. In semi-implicit schemes, the viscous terms in the Navier-Stokes equations are treated implicitly while the inviscid terms are treated explicitly. This should remove the viscous time step restriction. The semi-implicit scheme of Dong and Zhong [109] seems to be the best choice for the code. Dong and Zhong designed a high-order semi-implicit scheme for the unsteady compressible Navier-Stokes equations. Dong and Zhong derived their method on a curvilinear grid where the wall-normal direction was always aligned with the η direction in computational space. This meant that they only need to make the η direction derivatives implicit since the η direction had the smallest grid spacing. To implement this method in the current code, the viscous derivatives in all directions would need to be made implicit due to the cut-cell method.

The second possible approach to reducing the runtime is to use an adaptive mesh refinement method. Shen et al. [84] developed a high-order finite-difference adaptive mesh refinement method. The currently implemented multi-zone refinement method was based on the work of Shen et al. but instead of being adaptive, the refinement zones are fixed in time. For flow over isolated roughness elements, the areas requiring increased resolution tend to be fixed in time so the adaptive portion of the method was not needed. Keeping the grids fixed also made the implementation simpler. One downside is that the areas needing increased resolution must be known before the start of the simulation. Also, for the current implementation, there is only a single grid per refinement level. If there are two areas requiring the same level of refinement which are far apart, a single grid must be used to cover both area. This results in wasted grid points between the two areas. Implementing the entire adaptive mesh refinement method of Shen et al. would remove both of these issues.

Appendix A

A.1 Two-Dimensional Characteristic Transformation

The three-dimensional characteristic transformation is provided in section 3.1.1.3. The two-dimensional transformation cannot be easily reduced from the three-dimensional case and therefore, is presented here. For two-dimensional problems, the right eigenvectors for $\partial \mathbf{F}_\xi / \partial \mathbf{U}$ are the columns of

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ u - \tilde{\xi}_x c & u & \tilde{\xi}_y & u + \tilde{\xi}_x c \\ v - \tilde{\xi}_y c & v & -\tilde{\xi}_x & v + \tilde{\xi}_y c \\ H - \tilde{\theta}_\xi c & \frac{1}{2}(u^2 + v^2) & \tilde{\xi}_y u - \tilde{\xi}_x v & H + \tilde{\theta}_\xi c \end{pmatrix} \quad (\text{A.1})$$

and the left eigenvectors are the rows of

$$\mathbf{L} = \frac{1}{2} \begin{pmatrix} b_2 b_1 + \tilde{\theta}_\xi / c & -(b_1 u + \tilde{\xi}_x / c) & -(b_1 v + \tilde{\xi}_y / c) & b_1 \\ 2(1 - b_1 b_2) & 2b_1 u & 2b_1 v & -2b_1 \\ 2(\tilde{\xi}_x v - \tilde{\xi}_y u) & 2\tilde{\xi}_y & -2\tilde{\xi}_x & 0 \\ b_1 b_2 - \tilde{\theta}_\xi / c & -(b_1 u - \tilde{\xi}_x / c) & -(b_1 v - \tilde{\xi}_y / c) & b_1 \end{pmatrix} \quad (\text{A.2})$$

where

$$b_1 = \frac{\gamma - 1}{c^2}, \quad b_2 = \frac{1}{2}(u^2 + v^2), \quad (\text{A.3a})$$

$$H = \frac{e + p}{\rho}, \quad (\text{A.3b})$$

$$\tilde{\xi}_x = \frac{\xi_x}{\sqrt{\xi_x^2 + \xi_y^2}}, \quad \tilde{\xi}_y = \frac{\xi_y}{\sqrt{\xi_x^2 + \xi_y^2}}, \quad \tilde{\theta}_\xi = \tilde{\xi}_x u + \tilde{\xi}_y v. \quad (\text{A.3c})$$

ξ_x and ξ_y are the metrics ξ_1^1 and ξ_2^1 , respectively. The right and left eigenvectors for $\partial \mathbf{f}_\eta / \partial \mathbf{u}$ can be computed from Eqns. A.1 and A.2 by substituting η_x and η_y for ξ_x and ξ_y .

A.2 Roe Average

The Roe averaged state between points i and $i + 1$ will be denoted with an overbar and the index $i + 1/2$ (e.g., $\bar{u}_{i+1/2}$). The first quantity computed is the ratio of densities

$$R = \sqrt{\frac{\rho_{i+1}}{\rho_i}}. \quad (\text{A.4})$$

With the density ratio known, the Roe average of the quantities can be computed from the following equations:

$$\bar{\rho}_{i+1/2} = R\rho_i, \quad (\text{A.5})$$

$$\bar{u}_{i+1/2} = \frac{Ru_{i+1} + u_i}{R + 1}, \quad (\text{A.6})$$

$$\bar{v}_{i+1/2} = \frac{Rv_{i+1} + v_i}{R + 1}, \quad (\text{A.7})$$

$$\bar{w}_{i+1/2} = \frac{Rw_{i+1} + w_i}{R + 1}, \quad (\text{A.8})$$

$$\bar{H}_{i+1/2} = \frac{RH_{i+1} + H_i}{R + 1}, \quad (\text{A.9})$$

$$\bar{c}_{i+1/2} = \sqrt{\left(\gamma - 1\right) \left(\bar{H}_{i+1/2} - \frac{\bar{u}_{i+1/2}^2 + \bar{v}_{i+1/2}^2 + \bar{w}_{i+1/2}^2}{2}\right)}, \quad (\text{A.10})$$

where H is the enthalpy (Equation A.3b) and c is the speed of sound. The Roe average of any other variable (f) can be found from

$$\bar{f}_{i+1/2} = \frac{Rf_{i+1} + f_i}{R + 1}. \quad (\text{A.11})$$

REFERENCES

- [1] J. D. Anderson, *Hypersonic and High Temperature Gas Dynamics*. American Institute of Aeronautics & Astronautics, 2 ed., 2006.
- [2] S. B. Pope, *Turbulent Flows*. Cambridge University Press, 2005.
- [3] S. Berry and T. Horvath, “Discrete roughness transition for hypersonic flight vehicles,” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2007-307, Jan 2007.
- [4] S. P. Schneider, “Effects of roughness on hypersonic boundary-layer transition,” *Journal of Spacecraft and Rockets*, vol. 45, pp. 193–209, Mar-Apr 2008.
- [5] E. Reshotko, “Transition issues for atmospheric entry,” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2007-304, Jan 2007.
- [6] L. Mack, “Boundary-layer linear stability theory,” in *Special Course on Stability and Transition of Laminar Flow (AGARD-R-709)*, pp. 3/1–81, AGARD, June 1984.
- [7] M. Malik, “Numerical methods for hypersonic boundary-layer stability,” *Journal of Computational Physics*, vol. 86, pp. 376–413, Feb 1990.
- [8] A. Fedorov, “Receptivity of a high-speed boundary layer to acoustic disturbances,” *Journal of Fluid Mechanics*, vol. 491, pp. 101–129, Sep 2003.
- [9] Y. Ma and X. Zhong, “Receptivity of a supersonic boundary layer over a flat plate. part 1. wave structures and interactions,” *Journal of Fluid Mechanics*, vol. 488, pp. 31–78, 2003.
- [10] A. Tumin and E. Reshotko, “Spatial theory of optimal disturbances in boundary layers,” *Physics of Fluids*, vol. 13, pp. 2097–2104, July 2001.
- [11] E. White and E. Reshotko, “Roughness-induced transient growth in a flat-plate boundary layer,” in *40th AIAA Aerospace Sciences Meeting & Exhibit*, AIAA Paper 2002-0138, Jan 2002.
- [12] E. White and F. Ergin, “Receptivity and transient growth of roughness-induced disturbances,” in *33rd AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2003-4243, June 2003.
- [13] E. B. White, J. M. Rice, and F. Gokhan Ergin, “Receptivity of stationary transient disturbances to surface roughness,” *Physics of Fluids*, vol. 17, p. 064109, June 2005.
- [14] P. Fischer and M. Choudhari, “Numerical simulation of roughness-induced transient growth in a laminar boundary layer,” in *34th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2004-2539, June 2004.

- [15] M. Choudhari and P. Fischer, “Roughness-induced transient growth,” in *35th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2005-4765, June 2005.
- [16] E. Reshotko and A. Tumin, “Role of transient growth in roughness-induced transition,” *AIAA Journal*, vol. 42, pp. 766–770, Apr 2004.
- [17] D. C. Reda, “Correlation of nosetip boundary-layer-transition data measured in ballistics-range experiments,” *AIAA Journal*, vol. 19, no. 3, pp. 329–339, 1981.
- [18] D. C. Reda, “Review and Synthesis of Roughness-Dominated Transition Correlations For Reentry Applications,” *Journal of Spacecraft and Rockets*, vol. 39, pp. 161–167, Mar 2002.
- [19] M. Acarlar and C. Smith, “A Study of Hairpin Vortices in a Laminar Boundary Layer. Part 1. Hairpin Vortices Generated by a Hemisphere Protuberance,” *Journal of Fluid Mechanics*, vol. 175, pp. 1–41, Feb 1987.
- [20] F. G. Ergin and E. B. White, “Unsteady and transitional flows behind roughness elements,” *AIAA Journal*, vol. 44, pp. 2504–2514, Nov 2006.
- [21] C.-L. Chang and M. Choudhari, “Hypersonic viscous flow over large roughness elements,” in *47th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2009-0173, 2009.
- [22] D. Papamoschou and A. Roshko, “The compressible turbulent shear-layer: An experimental study,” *Journal of Fluid Mechanics*, vol. 197, pp. 453–477, Dec 1988.
- [23] P. Danehy, B. Bathel, C. Ivey, J. Inman, and S. Jones, “NO PLIF Study of Hypersonic Transition Over a Discrete Hemispherical Roughness Element,” in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, AIAA Paper 2009-394, Jan 2009.
- [24] C.-L. Chang, M. Choudhari, and F. Li, “Numerical computations of hypersonic boundary-layer over surface irregularities,” in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2010-1572, Jan 2010.
- [25] M. Choudhari, F. Li, M. Wu, C.-L. Chang, J. Edwards, M. Kegerise, and R. King, “Laminar-turbulent transition behind discrete roughness elements in a high-speed boundary layer,” in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2010-1575, Jan 2010.
- [26] M. D. Bartkiewicz, P. Subbareddy, and G. Candler, “Numerical simulations of roughness induced instability in the purdue mach 6 wind tunnel,” in *40th Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-4723, 2010.

- [27] B. M. Wheaton and S. P. Schneider, “Roughness-induced instability in a laminar boundary layer at mach 6,” in *48th AIAA Aerospace Sciences Meeting*, AIAA Paper 2010-1574, 2010.
- [28] B. Wheaton and S. Schneider, “Roughness-Induced Instability in a Hypersonic Laminar Boundary Layer,” *AIAA Journal*, vol. 50, no. 6, pp. 1245–1256, 2012.
- [29] Z. Duan, Z. Xiao, and S. Fu, “Direct Numerical Simulation of Hypersonic Transition Induced by a Cylindrical Roughness Element,” in *43rd Fluid Dynamics Conference*, AIAA Paper 2013-3112, 2013.
- [30] R. Mittal and G. Iaccarino, “Immersed boundary methods,” *Annual Review Of Fluid Mechanics*, vol. 37, pp. 239–261, 2005.
- [31] C. S. Peskin, “Flow patterns around heart valves: A numerical method,” *Journal of Computational Physics*, vol. 10, no. 2, pp. 252 – 271, 1972.
- [32] C. S. Peskin, “Numerical analysis of blood flow in the heart,” *Journal of Computational Physics*, vol. 25, no. 3, pp. 220 – 252, 1977.
- [33] D. Goldstein, R. Handler, and L. Sirovich, “Modeling a no-slip flow boundary with an external force field,” *Journal of Computational Physics*, vol. 105, no. 2, pp. 354 – 366, 1993.
- [34] J. Mohd-Yusof, “Combined Immersed-Boundary/B-spline methods for simulations of flow in complex geometries,” annual research briefs, Center for Turbulence Research, 1997.
- [35] A. M. Roma, C. S. Peskin, and M. J. Berger, “An adaptive version of the immersed boundary method,” *Journal of Computational Physics*, vol. 153, no. 2, pp. 509 – 534, 1999.
- [36] M. Lai and C. Peskin, “An immersed boundary method with formal second-order accuracy and reduced numerical viscosity,” *Journal of Computational Physics*, vol. 160, pp. 705–719, May 20 2000.
- [37] J. Kim, D. Kim, and H. Choi, “An immersed-boundary finite-volume method for simulations of flow in complex geometries,” *Journal of Computational Physics*, vol. 171, pp. 132–150, Jul 20 2001.
- [38] M. Uhlmann, “An immersed boundary method with direct forcing for the simulation of particulate flows,” *Journal of Computational Physics*, vol. 209, no. 2, pp. 448 – 476, 2005.
- [39] B. Griffith and C. Peskin, “On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems,” *Journal of Computational Physics*, vol. 208, pp. 75–105, Sep 1 2005.

- [40] D. Kim and H. Choi, “Immersed boundary method for flow around an arbitrarily moving body,” *Journal of Computational Physics*, vol. 212, no. 2, pp. 662 – 680, 2006.
- [41] K. Taira and T. Colonius, “The immersed boundary method: A projection approach,” *Journal of Computational Physics*, vol. 225, no. 2, pp. 2118 – 2137, 2007.
- [42] W.-F. Hu, Y. Kim, and M.-C. Lai, “An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in Navier-Stokes flows,” *Journal of Computational Physics*, vol. 257, pp. 670–686, Jan 15 2014.
- [43] S. O. Unverdi and G. Tryggvason, “A front-tracking method for viscous, incompressible, multi-fluid flows,” *Journal of Computational Physics*, vol. 100, no. 1, pp. 25 – 37, 1992.
- [44] R. Beyer, “A computational model of the cochlea using the immersed boundary method,” *Journal of Computational Physics*, vol. 98, pp. 145–162, Jan 1992.
- [45] R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. Haworth, “Large eddy simulation in complex geometric configurations using boundary body forces,” *AIAA Journal*, vol. 38, pp. 427–433, Mar. 2000.
- [46] R. Verzicco, M. Fatica, G. Iaccarino, B. Khalighi, and P. Moin, “Large Eddy Simulation of a Road Vehicle with Drag-Reduction Devices,” *AIAA Journal*, vol. 40, pp. 2447–2455, Dec. 2002.
- [47] L. Zhu and C. Peskin, “Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method,” *Journal of Computational Physics*, vol. 179, pp. 452–468, JUL 1 2002.
- [48] E. Balaras, “Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations,” *Computers & Fluids*, vol. 33, no. 3, pp. 375 – 404, 2004.
- [49] L. Miller and C. Peskin, “A computational fluid dynamics of ‘clap and fling’ in the smallest insects,” *Journal of Experimental Biology*, vol. 208, pp. 195–212, Jan 2005.
- [50] O. Marxen and G. Iaccarino, “Numerical simulation of the effect of a roughness element on high-speed boundary-layer instability,” in *38th Fluid Dynamics Conference and Exhibit*, AIAA Paper 2008-4400, June 2008.
- [51] J. Teran and C. Peskin, “Tether force constraints in stokes flow by the immersed boundary method on a periodic domain,” *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3404–3416, 2009.
- [52] P. A. J. van Melick and B. J. Geurts, “Flow through a cylindrical pipe with a periodic array of fractal orifices,” *Fluid Dynamics Research*, vol. 45, Dec 2013.

- [53] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, “A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method),” *Journal of Computational Physics*, vol. 152, no. 2, pp. 457 – 492, 1999.
- [54] R. P. Fedkiw, T. Aslam, and S. Xu, “The ghost fluid method for deflagration and detonation discontinuities,” *Journal of Computational Physics*, vol. 154, no. 2, pp. 393 – 427, 1999.
- [55] S. Majumdar, G. Iaccarino, and P. Durbin, “RANS solvers with adaptive structured boundary non-conforming grids,” in *Center for Turbulence Research Annual Research Briefs*, 2001.
- [56] Y.-H. Tseng and J. H. Ferziger, “A ghost-cell immersed boundary method for flow in complex geometry,” *Journal of Computational Physics*, vol. 192, no. 2, pp. 593 – 623, 2003.
- [57] T. Liu, B. Khoo, and K. Yeo, “Ghost fluid method for strong shock impacting on material interface,” *Journal of Computational Physics*, vol. 190, no. 2, pp. 651 – 681, 2003.
- [58] R. Ghias, R. Mittal, and T. S. Lund, “A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries,” in *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2004-0080, 2004.
- [59] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke, “A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries,” *Journal of Computational Physics*, vol. 227, no. 10, pp. 4825 – 4852, 2008.
- [60] D. K. Clarke, H. A. Hassan, and M. D. Salas, “Euler calculations for multielement airfoils using cartesian grids,” *AIAA Journal*, vol. 24, pp. 353–358, Mar 1986.
- [61] H. Udaykumar, R. Mittal, and W. Shyy, “Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids,” *Journal of Computational Physics*, vol. 153, no. 2, pp. 535 – 574, 1999.
- [62] H. Udaykumar, R. Mittal, P. Rampungoon, and A. Khanna, “A sharp interface cartesian grid method for simulating flows with complex moving boundaries,” *Journal of Computational Physics*, vol. 174, no. 1, pp. 345 – 380, 2001.
- [63] H. S. Udaykumar, R. Mittal, and P. Rampungoon, “Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes,” *Communications in Numerical Methods in Engineering*, vol. 18, no. 2, pp. 89–97, 2002.
- [64] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy, “An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries,” *Journal of Computational Physics*, vol. 156, pp. 209–240, Dec 10 1999.

- [65] M. Kirkpatrick, S. Armfield, and J. Kent, “A representation of curved boundaries for the solution of the navier-stokes equations on a staggered three-dimensional cartesian grid,” *Journal of Computational Physics*, vol. 184, no. 1, pp. 1 – 36, 2003.
- [66] L. Duan, X. Wang, and X. Zhong, “A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness,” *Journal of Computational Physics*, vol. 229, no. 19, pp. 7207 – 7237, 2010.
- [67] F. Bouchon, T. Dubois, and N. James, “A second-order cut-cell method for the numerical simulation of 2d flows past obstacles,” *Computers & Fluids*, vol. 65, pp. 80 – 91, 2012.
- [68] M. Meinke, L. Schneiders, C. Günther, and W. Schröder, “A cut-cell method for sharp moving boundaries in cartesian grids,” *Computers & Fluids*, vol. 85, pp. 135 – 142, 2013.
- [69] M. W. Johnson, “A novel cartesian CFD cut cell approach,” *Computers & Fluids*, vol. 79, pp. 105 – 119, 2013.
- [70] M. J. Berger and R. J. LeVeque, “Stable boundary conditions for cartesian grid calculations,” *Computing Systems in Engineering*, vol. 1, no. 24, pp. 305 – 311, 1990.
- [71] H. Johansen and P. Colella, “A cartesian grid embedded boundary method for poisson’s equation on irregular domains,” *Journal of Computational Physics*, vol. 147, no. 1, pp. 60 – 85, 1998.
- [72] D. Causon, D. Ingram, C. Mingham, G. Yang, and R. Pearson, “Calculation of shallow water flows using a cartesian cut cell approach,” *Advances in Water Resources*, vol. 23, no. 5, pp. 545 – 562, 2000.
- [73] P. McCorquodale, P. Colella, and H. Johansen, “A cartesian grid embedded boundary method for the heat equation on irregular domains,” *Journal of Computational Physics*, vol. 173, no. 2, pp. 620 – 635, 2001.
- [74] R. Mittal, Y. Utturkar, and H. Udaykumar, “Computational modeling and analysis of biomimetic flight mechanisms,” in *Aerospace Sciences Meetings*, American Institute of Aeronautics and Astronautics, Jan 2002.
- [75] J. Zhang and M.-J. Ni, “A consistent and conservative scheme for mhd flows with complex boundaries on an unstructured cartesian adaptive system,” *Journal of Computational Physics*, vol. 256, pp. 520 – 542, 2014.
- [76] M. J. Berger and J. Oliger, “Adaptive mesh refinement for hyperbolic partial differential equations,” *Journal of Computational Physics*, vol. 53, no. 3, pp. 484 – 512, 1984.
- [77] M. Berger and P. Colella, “Local adaptive mesh refinement for shock hydrodynamics,” *Journal of Computational Physics*, vol. 82, no. 1, pp. 64 – 84, 1989.

- [78] M. J. Berger and R. J. Leveque, “Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems,” *SIAM J. Numer. Anal.*, vol. 35, pp. 2298–2316, 1998.
- [79] J. Bell, M. Berger, J. Saltzman, and M. Welcome, “Three-dimensional adaptive mesh refinement for hyperbolic conservation laws,” *SIAM Journal on Scientific Computing*, vol. 15, no. 1, pp. 127–138, 1994.
- [80] D. S. Balsara, “Divergence-free adaptive mesh refinement for magnetohydrodynamics,” *Journal of Computational Physics*, vol. 174, no. 2, pp. 614 – 648, 2001.
- [81] L. Jameson, “Amr vs high order schemes,” *Journal of Scientific Computing*, vol. 18, no. 1, pp. 1–24, 2003.
- [82] A. J. Cunningham, A. Frank, P. Varnière, S. Mitran, and T. W. Jones, “Simulating magnetohydrodynamical flow with constrained transport and adaptive mesh refinement: Algorithms and tests of the astrobear code,” *The Astrophysical Journal Supplement Series*, vol. 182, no. 2, p. 519, 2009.
- [83] D. A. Olivieri, M. Fairweather, and S. A. E. G. Falle, “Rans modelling of intermittent turbulent flows using adaptive mesh refinement methods,” *Journal of Turbulence*, vol. 11, no. 29, pp. 1–18, 2010.
- [84] C. Shen, J.-M. Qiu, and A. Christlieb, “Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations,” *Journal of Computational Physics*, vol. 230, no. 10, pp. 3780–3802, 2011.
- [85] M. J. Berger, D. L. George, R. J. LeVeque, and K. T. Mandli, “The geoclaw software for depth-averaged flows with adaptive refinement,” *Advances in Water Resources*, vol. 34, no. 9, pp. 1195 – 1206, 2011.
- [86] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, “An adaptive, formally second order accurate version of the immersed boundary method,” *Journal of Computational Physics*, vol. 223, no. 1, pp. 10 – 49, 2007.
- [87] E. Uzgoren, J. Sim, and W. Shyy, “Marker-Based, 3-D Adaptive Cartesian Grid Method for Multiphase Flow Around Irregular Geometries,” *Communications in Computational Physics*, vol. 5, pp. 1–41, Jan 2009.
- [88] S. K. Sambasivan and H. UdayKumar, “Sharp interface simulations with local mesh refinement for multi-material dynamics in strongly shocked flows,” *Computers & Fluids*, vol. 39, no. 9, pp. 1456 – 1479, 2010.
- [89] P. Lee, B. E. Griffith, and C. S. Peskin, “The immersed boundary method for advection-electrodiffusion with implicit timestepping and local mesh refinement,” *Journal of Computational Physics*, vol. 229, no. 13, pp. 5208 – 5227, 2010.

- [90] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis, 2nd ed., 1997.
- [91] Y.-X. Ren, M. Liu, and H. Zhang, “A characteristic-wise hybrid compact-weno scheme for solving hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 192, no. 2, pp. 365 – 386, 2003.
- [92] X. Zhong, “High-order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition,” *Journal of Computational Physics*, vol. 144, pp. 662–709, Aug 1998.
- [93] G. Jiang and C. Shu, “Efficient implementation of weighted eno schemes,” *Journal of Computational Physics*, vol. 126, no. 1, pp. 202–228, 1996.
- [94] A. K. Henrick, T. D. Aslam, and J. M. Powers, “Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points,” *Journal of Computational Physics*, vol. 207, no. 2, pp. 542 – 567, 2005.
- [95] M. H. Carpenter, D. Gottlieb, and S. Abarbanel, “The theoretical accuracy of runge-kutta time discretizations for the initial boundary value problem: A careful study of the boundary error,” tech. rep., Institute for Computer Applications in Science and Engineering, 1993.
- [96] O. Marxen and G. Iaccarino, “An immersed boundary method for numerical simulations of boundary layers with roughness,” in *Center for Turbulence Research Annual Research Briefs*, 2008.
- [97] D. A. von Terzi, M. N. Linnick, J. Seidel, and H. F. Fasel, “Immersed boundary techniques for high-order finite-difference methods,” in *15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2001-2918, 2001.
- [98] S. Xu and M. Martin, “Assessment of inflow boundary conditions for compressible turbulent boundary layers,” *Physics of Fluids*, vol. 16, pp. 2623–2639, July 2004.
- [99] T. Poinso and S. Lele, “Boundary conditions for direct simulations of compressible viscous flows,” *Journal of Computational Physics*, vol. 101, pp. 104 – 129, 1992.
- [100] C. Williamson, “Vortex dynamics in the cylinder wake,” *Annual Review Of Fluid Mechanics*, vol. 28, pp. 477–539, 1996.
- [101] Canonical Ltd, *Ubuntu*. <http://www.ubuntu.com>.
- [102] S. P. Schneider, “Fabrication and testing of the purdue mach-6 quiet-flow ludwig tube,” in *38th Aerospace Sciences Meeting & Exhibit*, AIAA Paper 2000-0295, Jan 2000.
- [103] S. P. Schneider, L. Rufer, L. Randall, and C. Skoch, “Shakedown of the purdue mach-6 quiet-flow ludwig tube,” in *39th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2001-0457, Jan 2001.

- [104] K. W. Thompson, “Time-dependent boundary-conditions for hyperbolic systems,” *Journal of Computational Physics*, vol. 68, pp. 1–24, Jan 1987.
- [105] B. M. Wheaton, “Roughness-induced instability in a laminar boundary layer at mach 6,” Master’s thesis, Purdue University, 2009.
- [106] C. Ward, B. Wheaton, A. Chou, P. Gilbert, L. Steen, and S. Schneider, “Boundary-layer transition measurements in a mach-6 quiet tunnel,” in *40th Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-4721, 2010.
- [107] S. P. Spekreijse, “Elliptic grid generation based on laplace equations and algebraic transformations,” *Journal of Computational Physics*, vol. 118, pp. 38–61, Apr 1995.
- [108] B. M. Wheaton, M. D. Bartkowicz, P. Subbareddy, S. Schneider, and G. Candler, “Roughness-induced instabilities at mach 6: A combined numerical and experimental study,” in *41st AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2011-3248, June 2011.
- [109] H. Dong and X. Zhong, “High-order semi-implicit schemes for unsteady compressible flow simulations,” *AIAA Journal*, vol. 40, pp. 869–878, May 2002.