

The Characterization of High Aspect Ratio Rectangular Jet Flows

by

Carlos Christopher Alfredo Mendez

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering
University of Alberta

© Carlos Christopher Alfredo Mendez, 2020

Abstract

Steam assisted gravity drainage (SAGD) is an in-situ process where an injection well and a production well are used to recover bitumen from oil sands that are past the point of viability for open mining techniques. During operation, sand control technologies are used on the wells to ensure the longevity of the SAGD system. Slotted liners are the most commonly used sand control technology for SAGD due to their low manufacturing costs and their superior mechanical integrity. The design of a slotted liner includes many narrow slots about the circumference. With the placement of the slotted liner on the injection well, the injected steam's flow field is rendered complex. To approach such a complex flow field, a fundamental understanding of the flow exiting a single slot is first necessary.

This thesis aimed to investigate the flow field of a jet exiting a single slot. The aspect ratio, AR , and Reynolds number, Re , of the slot were chosen for study. In order to approach such a task, particle image velocimetry (PIV) techniques were applied to measure the flow field of the emerging jet. Streamwise and cross-sectional measurements were taken using two-dimensional (2D-PIV) and stereoscopic PIV (stereo-PIV) based setups, respectively. The single slot was made to have high- AR configurations of $AR = 100, 150, 200,$ and 210 at a constant area of $A_{\text{rec}} = 12 \text{ mm}^2$, which resulted in an equivalent diameter of $D_e = 3.91 \text{ mm}$. The flow rate of the jet exiting the slot was varied so that six Reynolds numbers were observed, at values of $Re = 10800, 14400, 18000, 21600, 25200,$ and 28800 . The streamwise measurements were taken within the X - Y plane, at $Z = 0$, and observed a downstream range spanning from $X = 0$ to $X = 7D_e$. In turn, The cross-sectional measurements were taken downstream from the slot exit, within the Y - Z plane at $X = 3D_e, 5D_e,$ and $7D_e$.

The first study to be conducted was with regards to high- AR effects. When analyzing the streamwise measurements, many dissimilarities between low- AR and high- AR jets were observed when the velocity decay, streamwise turbulence intensity, and lateral velocity half-width were considered. Further dissimilarities were observed in the cross-sectional measurements when velocities, turbulence intensities, shear stresses, turbulent kinetic energy, and streamwise vorticity were considered. Each jet was found to display a high number

of vortex pairs, so it was concluded that AR has a significant influence on the formation of the streamwise vorticity. It was also observed that small-scale disturbances at the slot exit led to large asymmetries in the distribution of vortex pairs, which in turn led to the jets displaying notable asymmetry. From these two observations on the vortex pairs, it was concluded that the initial distribution of the streamwise vorticity led to the overall differences that were perceived between low- AR and high- AR jets. Thus, it was concluded that a significant change in the downstream flow field occurs when high- AR configurations are utilized.

The second study to be conducted was concerned with the influence of Re . Clear Re -based trends were observed to exist when the velocity decay, streamwise turbulence intensity, lateral velocity half-width, and cross-sectional streamwise velocity were considered. Based on the observations, it was concluded that fully turbulent flow had not been reached for the first four Re configurations. Few differences were observed between the results of $Re = 25200$, and 28800 , so it was concluded that the critical Reynolds number, Re_{critical} , at which flows are fully turbulent had been passed. Thus, Re_{critical} was observed to occur within the interval of $Re = 21600$ and 25200 . This range was found to be higher than the Re_{critical} range given within the literature. It was further observed that Re directly influences the formation, distribution, and evolution of streamwise vortices. It was found that the number of streamwise vortex pairs and the dissipation of the vortex pairs downstream both increase with increasing Re . An array of evenly spaced vortex pairs was found to exist for fully turbulent jets. Though it was not concluded, it was proposed that the formation of such an array is what allows for fully turbulent jets to behave similarly.

Preface

This thesis is an original work completed by Carlos C. A. Mendez. The work done for this thesis study was completed under the supervision of Dr. David S. Nobes of the Department of Mechanical Engineering at the University of Alberta.

The main flow facility (detailed in Chapter 3.2) was designed, and developed under the supervision of Dr. David S. Nobes. The development and design of the test piece were completed in collaboration with Yishak Yusuf. The test piece was manufactured at the University of Alberta's Machine Shop located within the Mechanical Engineering building. Aspects of the design of the flow facility were completed in collaborations with Dr. Yishak Yusuf, Jason Michaud, and Dr. Mouhammad El Hassan.

The experimentation for this thesis was completed in the Applied Thermofluids Laboratory at the University of Alberta. Key components of the particle imaging velocimetry setup were in part received through contributions made by Dr. Sina Ghaemi of the Department of Mechanical Engineering at the University of Alberta.

Part of the content contained in Chapter 4 of this thesis has been presented as, Mendez C., Nobes, D.S. (2019). Investigation of the Velocity Field of a Free Jet Issuing From a High Aspect-Ratio Slot. In *4th Thermal and Fluids Engineering Conference*. Las Vegas, NV.

This thesis is dedicated to my mother, who not only motivated me to pursue a career in Mechanical Engineering, but also motivated me to pursue my Master's degree.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dr. Nobes. His willingness to accept me under his supervision for this project was truly a blessing in itself. His leadership during the course of my research allowed for a unique experience that allowed me to not only reach my academic goals, but also develop my research abilities along with my design skills as an engineer.

I would also like to express my thanks to my research group members. Whether it was an ear or a hand to lend when issues arose in my work, or simply a friendly demeanor to joke around with, they were instrumental in creating a wonderful workplace. I would like to specifically thank my colleague, Yishak Yusuf, who went above and beyond when assisting me in my research, whether it was with regards to the understanding of theory, design ideas, or troubleshooting. I would also like to thank Rafat Jami, Daren Wilkinson, and Drew Gingras for their friendships and assistance through my degree.

I would also like to express my gratitude and thanks to those closest to me: my family members and friends. All of my family members have been supportive to me in my academic journey, and have never been short to express their pride in my accomplishments. I would like to specifically thank my mother, uncle Cesar, and sister Julia for the roles they've had in my success. I would like to thank my two closest friends, Eyuiael Mersha and Sushanth Sai, who are both true blessings in my life and have continuously been present during the best and worst of times. I would also like to thank some others who I am truly blessed to call my friends, like Nkosana Moyo, Alex Foster, Anthony Ho, Aaron Mitra, Daniel Jimenez, Emmanuel Dinarte, Kevin Herrera, Navid Israq, Bilal Mustafa, David Herrera, Shahrukh Parvez, Ethan Cairns, Haley and Brooke Jurak, Khadija Nasser, and Caitlane Tarun. There are many others who have supported me in this journey that I am lucky to call friends, and I give my deepest thanks to them as well.

Finally, funding support for this project was made available by RGL Reservoir Management Inc. and the Natural Sciences and Research Council (NSERC) of Canada.

Contents

Abstract	ii
Preface	iv
Dedication	v
Acknowledgements	vi
Contents	vii
List of Figures	xi
List of Tables	xvi
1 Introduction	1
1.1 Steam Assisted Gravity Drainage	1
1.2 Conclusions	5
2 Literature Review	7
2.1 Overview of Jets and Jet Applications	7
2.2 The Rectangular Jet	8
2.2.1 Exit Geometry	8
2.2.2 Definition of Aspect Ratio for a Rectangular Jet	9
2.2.3 Definition of Reynolds Number for a Rectangular Jet	9
2.2.4 Studies on the Effects of Aspect Ratio	11
2.2.5 Studies on the Effects of Reynolds Number	20
2.2.6 Further Concepts Related to the Study of Rectangular Jets	21
2.3 Measurement Techniques for Jet Experiments	22
2.4 Research Objectives	24

2.5	Thesis Outline	24
3	Experimental Setup	27
3.1	Introduction	27
3.2	Flow Facility Setup	27
3.2.1	Experiment Slot Assembly	27
3.2.2	Flow Conditioning Assembly	31
3.3	Particle Image Velocimetry	37
3.3.1	Stereoscopic PIV	38
3.4	Particle Image Velocimetry Setups	39
3.4.1	Two-Dimensional Measurements	39
3.4.2	Stereo-PIV Measurements	41
3.5	Flow Line Arrangement	44
3.5.1	Tracer Particles	44
3.5.2	Initial Flow Arrangement Design	45
3.5.3	Final Flow Arrangement Design and Development	47
4	Data Processing Methodology	51
4.1	Overview of Image Processing	51
4.2	Processing of 2D-PIV data	51
4.2.1	Image Processing Scheme for Near-Exit Region	51
4.2.2	Image Processing Scheme for Far-Exit Region	53
4.2.3	Stitching of the Two Vector Fields	54
4.3	Processing of Stereo-PIV Data	59
4.3.1	Image Processing Scheme	59
4.3.2	Definition and Determination of the Jet Boundary	60
4.4	Uncertainty of PIV Data	63
5	Influence of Aspect Ratio	65
5.1	Introduction	65
5.2	Streamwise Flow Field Observations	66
5.2.1	Centerline Decay	66
5.2.2	Streamwise Velocity Turbulence Intensity	69
5.2.3	Velocity Half-Width	70
5.3	Cross-sectional Flow Field Observations	71

5.3.1	Whole-Field Average Velocity Measurements	71
5.3.2	Whole-Field Average Turbulence Intensity Measurements	79
5.3.3	Streamwise Turbulence Intensity Measurements	79
5.3.4	Lateral and Spanwise Turbulence Intensity Measurements	79
5.3.5	Reynolds Shear Stresses	83
5.3.6	Lateral Reynolds Shear Stress	83
5.3.7	Spanwise Reynolds Shear Stress	83
5.3.8	Turbulent Kinetic Energy	86
5.3.9	Streamwise Vorticity	88
5.3.10	Cross-Sectional Boundary Deformation	90
5.4	Conclusions	92
6	Influence of Reynolds Number	93
6.1	Introduction	93
6.2	Streamwise Flow Field Observations	93
6.2.1	Centerline Decay	94
6.2.2	Streamwise Velocity Turbulence Intensity	94
6.2.3	Velocity Half-Width	95
6.3	Cross-sectional Flow Field Observations	96
6.3.1	Whole-Field Average Velocity Measurements	96
6.3.2	Whole-Field Average Turbulence Intensity Measurements	105
6.3.3	Streamwise Turbulence Intensity Measurements	105
6.3.4	Lateral and Spanwise Turbulence Intensity Measurements	105
6.3.5	Reynolds Shear Stresses	112
6.3.6	Lateral Reynolds Shear Stress	112
6.3.7	Spanwise Reynolds Shear Stress	112
6.3.8	Turbulent Kinetic Energy	117
6.3.9	Streamwise Vorticity	120
6.3.10	Cross-Sectional Boundary Deformation	123
6.4	Conclusions	125
7	Conclusions and Future Work	127
7.1	Conclusions	127
7.1.1	The Effects of Aspect Ratio on a Free, Rectangular Jet	127
7.1.2	The Effects of Reynolds Number on a Free, Rectangular Jet	128

7.2 Recommendations for Future Work	129
Works Cited	131
Appendices	135
A Flow Conditioning Assembly Drawing Package	135
B MATLAB Codes for 2D-PIV Post-Processing	156
C MATLAB Codes for Stereo-PIV Post-Processing	232
D Two-Dimensional PIV Figures	258
E Stereo-PIV Figures	292

List of Figures

1.1	A schematic of the in reservoir components of the SAGD process	2
1.2	Simplified cross-sectional views of the slotted liner with a single slot, which includes the (a) straight, (b) keystone, and (c) rolled top profiles (Bennion, Gupta, Gittins, & Hollies, 2009; Xie et al., 2007)	4
1.3	The three slot configuration patterns used for slotted liners, which includes the (a) staggered, (b) overlapping, and (c) gang configurations	4
2.1	Schematic representation of a rectangular exit with the origin located at its centre, and the orientation of the X , Y , and Z axes specified	8
2.2	Velocity decay of a rectangular jet showing the (a) potential core region, (b) characteristic decay region, and (c) the axisymmetric-type decay region	13
3.1	Images of a coupon showing (a) an unlabelled front side with a large AR slot, and (b) a labelled back side with a reduced AR slot	28
3.2	Three-dimensional models of the slot assembly (a) in an exploded isometric view, and (b) mated together in an isometric view	30
3.3	Cross-sectional view of the flow conditioning assembly with the test slot installed	32
3.4	Drawings of the vaned diffuser showing its (a) exit face, (b) cross-sectional profile, and (c) entrance face	33
3.5	Exploded view of the screen holder assembly	34
3.6	Drawing of the screen showing (a) the front face and (b) a detailed view of the screen threads	35
3.7	Drawing of the honeycomb showing (a) the front face and (b) a detailed view of the honeycomb grid	35
3.8	Drawings of the vaned diffuser showing its (a) end face and (b) cross-sectional profile	36

3.9	Conceptual schematic of a PIV arrangement showing the use of a laser to illuminate the tracer particles, whose movements are captured on two successive images	37
3.10	Conceptual schematic of a stereo-PIV arrangement showing the fulfillment of the Scheimpflug condition	38
3.11	A simplified three-dimensional model of the 2D-PIV setup relative to the flow conditioning assembly	40
3.12	A schematic showcasing the two FOVs used for the 2D-PIV setup	40
3.13	A simplified three-dimensional model of the stereo-PIV setup relative to the flow conditioning assembly	42
3.14	A schematic showcasing the three FOVs used for the stereo-PIV setup . . .	42
3.15	A solid model of the custom-made three-dimensional calibration target . . .	43
3.16	A schematic of the initial experimental flow arrangement setup	46
3.17	A schematic of the final experimental flow arrangement setup	47
4.1	Sample images of the near-exit jet flow showing examples of (a) the raw image, and (b) the pre-processed image with the geometric mask shown . .	52
4.2	Simplified schematic showing (a) determination of U_{\max} values in the first quarter of U data, and (b) the determination of the flow field's overall U_{\max}	55
4.3	A simplified schematic showing the determination of the downstream U_{cl} , and the straightening of the centerline's line of best fit about U_{\max}	57
4.4	A streamwise velocity whole-field map of a jet after the stitching process . .	59
4.5	Annotated screen capture to exemplify the horizontal test employed to determine the jet boundary	63
4.6	A streamwise velocity whole-field map of a jet plotted with its calculated jet boundary	63
5.1	Velocity decay at $Re = 18000$ with the X-axis nondimensionalized by (a) equivalent diameter and (b) slot width. Every 5 th data point has been plotted for clarity	68
5.2	Normalized turbulence intensity against X/D_e at $Re = 18000$. Every 5 th data point has been plotted for clarity	69
5.3	Normalized lateral velocity half-width against X/D_e at $Re = 18000$. Every 5 th data point has been plotted for clarity	70

5.4	A plot showing the variation of the streamwise velocity profile along the spanwise direction for $AR = 210$. All data points plotted	74
5.5	Average streamwise velocity maps at each in-plane location for each AR configuration at $Re = 18000$	75
5.6	Average lateral velocity maps at each in-plane location for each AR configuration at $Re = 18000$	77
5.7	Average spanwise velocity maps at each in-plane location for each AR configuration at $Re = 18000$	78
5.8	Average streamwise turbulence intensity maps at each in-plane location for each AR configuration at $Re = 18000$	80
5.9	Average lateral turbulence intensity maps at each in-plane location for each AR configuration at $Re = 18000$	81
5.10	Average spanwise turbulence intensity maps at each in-plane location for each AR configuration at $Re = 18000$	82
5.11	Average lateral Reynolds shear stress maps at each in-plane location for each AR configuration at $Re = 18000$	84
5.12	Average spanwise Reynolds shear stress maps at each in-plane location for each AR configuration at $Re = 18000$	85
5.13	Average TKE maps at each in-plane location for each AR configuration at $Re = 18000$	87
5.14	Average vorticity maps at each in-plane location for each AR configuration at $Re = 18000$	89
5.15	Jet boundary evolution at $Re = 18000$ at (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$	91
6.1	Velocity decay at $AR = 150$ with the X -axis nondimensionalized by equivalent diameter. Every 5 th data point has been plotted for clarity	94
6.2	Normalized turbulence intensity against X/D_e at $AR = 150$. Every 5 th data point has been plotted for clarity	95
6.3	Normalized lateral velocity half-width against X/D_e at $AR = 150$. Every 5 th data point has been plotted for clarity	96
6.4	Average streamwise velocity maps at each in-plane location for $Re = 18000$, 14400, and 18000 at $AR = 150$	98

6.5	Average streamwise velocity maps at each in-plane location for $Re = 21600$, 25200, and 28800 at $AR = 150$	99
6.6	Average lateral velocity maps at each in-plane location for $Re = 18000$, 14400, and 18000 at $AR = 150$	101
6.7	Average lateral velocity maps at each in-plane location for $Re = 21600$, 25200, and 28800 at $AR = 150$	102
6.8	Average spanwise velocity maps at each in-plane location for $Re = 18000$, 14400, and 18000 at $AR = 150$	103
6.9	Average spanwise velocity maps at each in-plane location for $Re = 21600$, 25200, and 28800 at $AR = 150$	104
6.10	Average streamwise turbulence intensity maps at each in-plane location for $Re = 18000$, 14400, and 18000 at $AR = 150$	106
6.11	Average streamwise turbulence intensity maps at each in-plane location for $Re = 21600$, 25200, and 28800 at $AR = 150$	107
6.12	Average lateral turbulence intensity maps at each in-plane location for $Re =$ 18000, 14400, and 18000 at $AR = 150$	108
6.13	Average lateral turbulence intensity maps at each in-plane location for $Re =$ 21600, 25200, and 28800 at $AR = 150$	109
6.14	Average spanwise turbulence intensity maps at each in-plane location for Re $= 18000$, 14400, and 18000 at $AR = 150$	110
6.15	Average spanwise turbulence intensity maps at each in-plane location for Re $= 21600$, 25200, and 28800 at $AR = 150$	111
6.16	Average lateral Reynolds shear stress maps at each in-plane location for Re $= 18000$, 14400, and 18000 at $AR = 150$	113
6.17	Average lateral Reynolds shear stress maps at each in-plane location for Re $= 21600$, 25200, and 28800 at $AR = 150$	114
6.18	Average spanwise Reynolds shear stress maps at each in-plane location for $Re = 18000$, 14400, and 18000 at $AR = 150$	115
6.19	Average spanwise Reynolds shear stress maps at each in-plane location for $Re = 21600$, 25200, and 28800 at $AR = 150$	116
6.20	Average TKE maps at each in-plane location for $Re = 18000$, 14400, and 18000 at $AR = 150$	118
6.21	Average TKE maps at each in-plane location for $Re = 21600$, 25200, and 28800 at $AR = 150$	119

6.22	Average vorticity maps at each in-plane location for $Re = 18000, 14400,$ and 18000 at $AR = 150$	121
6.23	Average vorticity maps at each in-plane location for $Re = 21600, 25200,$ and 28800 at $AR = 150$	122
6.24	Jet boundary evolution at $AR = 150$ (a) $X = 3D_e,$ (b) $X = 5D_e,$ and (c) $X = 7D_e$	124

List of Tables

2.1	Summary of the aspect ratio, Reynolds number, measurement range, and cross-sectional exit for rectangular jet studies	12
2.2	Summary of the measurement techniques used for the experimental investigations on the effects of aspect ratio on a rectangular jet	23
3.1	Dimensions, desired aspect ratio, actual aspect ratio, actual area, and actual D_e for the slot configurations considered in this study	31
3.2	Properties used in the calculation of ν_s via Equation 3.2	45
3.3	Flow rates, average exit velocities, and corresponding Reynolds numbers considered in this study	48
3.4	Flow rates and relative error for each aspect ratio configuration	49

Chapter 1

Introduction

1.1 Steam Assisted Gravity Drainage

In terms of proven oil reserves, which are reserves that are known to exist and are recoverable under current technological and economic conditions, Canada has an estimated 171.0 billion barrels available (Natural Resources Canada, 2017). This accounts for approximately 10% of the world's proven oil reserves (Natural Resources Canada, 2017). Of the 171.0 billion barrels of oil, about 166.3 billion barrels are found in Alberta's oil sands. The Albertan oil sands underlie 142,200 km² of land located in three areas of northern Alberta, Athabasca, Cold Lake, and Peace River (Alberta Energy, 2017). If the oil sand reserves are shallow enough - specifically up to a depth of approximately 75 meters - then the bitumen located within the oil sands may be extracted through open pit mining (Alberta Energy, 2017). However, shallow reserves only exist in the Athabasca oil sands area and encompass about 4800 km² of land (Natural Resources Canada, 2017). The shallow reserves allow for approximately 20% of the oil sands' bitumen to be extracted through open pit mining (RAMP, 2008). In order to extract the remaining 80% of bitumen within the oil sands, in-situ recovery techniques must be used instead (CAPP, 2018).

In-situ recovery techniques refer to the use of advanced technology to extract bitumen, which is located past the point of viability for open mining techniques, in place. Thermal in-situ recovery techniques are in-situ recovery techniques where the advanced technology is used to inject steam, combustion productions, or other heat sources to the oil sand reserves (CAPP, 2018). There are many thermal in-situ techniques that exist, which include Cyclic Steam Stimulation (CSS) (Haan & Lookeren, 1969; Martin, 1967; Willman, Valleroy, Runberg, Cornelius, & Powers, 1961), Steam-Assisted Gravity Drainage (SAGD) (Butler, 1994; Butler, McNab, & Lo, 1981; Butler & Stephens, 1981), the Vapor Extraction Process (VAPEX) (Das & Butler, 1998), and Toe to Heel Air Injection (THAI) (Greaves, Saghr,

Xia, Turtar, & Ayasse, 2001). Of these thermal in-situ techniques, the latter two are not currently used commercially, and SAGD is the most commonly used extraction technique, as it accounts for 38% of Alberta's total bitumen production (Alberta Government, 2017; Natural Resources Canada, 2017; RAMP, 2008).

For the SAGD process, two wells - an injection well and a production well - are used to recover bitumen (Butler, 1994; Butler et al., 1981; Butler & Stephens, 1981). As shown in Figure 1.1, the two wells are drilled to be horizontal and parallel to each other within the oil sands reservoir; the wells are placed approximately 5 m from each other, and their lengths typically range from 500 m to 1000 m. The injection well injects steam into the surrounding, cool oil sands, and heats up the bitumen so that there is a reduction in the bitumen's viscosity. As the bitumen heats up, the steam condenses and turns to water. Both the bitumen and water are acted upon by gravity and flow downwards to the production well, which transports the liquids above ground so that the bitumen may be processed. The bitumen-free space surrounding the two wells, the steam chamber, grows in size as the SAGD system operates; the growth of the steam chamber allows for a continuous interaction between the injected steam and the bitumen within the reservoir.

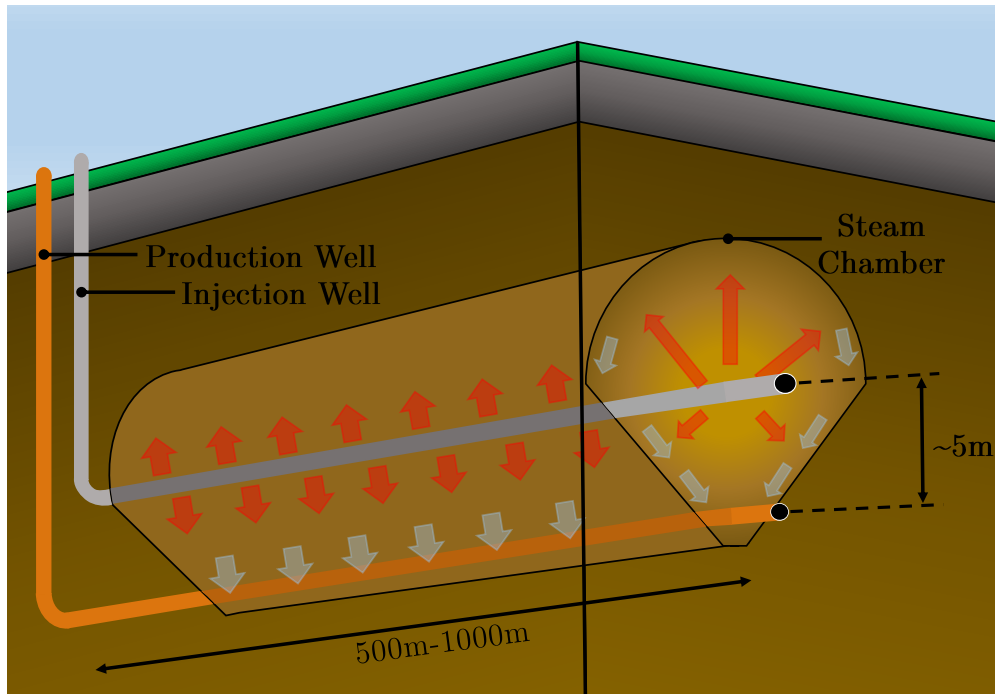


Figure 1.1: A schematic of the in reservoir components of the SAGD process

As a SAGD system operates, sand production in both wells is an issue that must be addressed so that the overall operation performance is not significantly inhibited (Ott &

Woods, 2003). Many factors affect the rate of sand production, which include: the degree of consolidation of the oil sands formation, pore pressure reduction during the operation's lifetime, reservoir fluid production rate, reservoir fluid viscosity, and the increase in water production during the operation's lifetime (Ott & Woods, 2003). Sand control methods are used to moderate sand production rate so that the harmful effects of sand production to the SAGD system are reduced. Without such precautions, the SAGD system is exposed to a higher degree of dangers, such as: sand accumulation in down-hole and surface equipment, erosion and corrosion of down-hole and surface equipment, and the collapse of the surrounding oil sands formation. Under such conditions, the constant inspection, maintenance, and replacement of down-hole and surface equipment, along with the removal of and disposal of solid waste, adds further financial burdens to the SAGD operation (Chalaturnyk, Wagg, & Dusseault, 1992; Mahmud, Leong, & Lestariono, 2019; Ott & Woods, 2003). Many sand control methods exist, such as, chemical consolidation, screens, slotted liners, special filters, inside-spacing and open-hole gravel packing, propped fracturing, selective and oriented perforating, frac packs, and production rate control (Carlson, Gurley, King, Price-Smith, & Waters, 1992; Matanović, Čikeš, & Moslavac, 2012; Ott & Woods, 2003). Of these sand control options, slotted liners are the most commonly used for SAGD sand control applications due to their low manufacturing costs and their superior mechanical integrity (Bennion et al., 2009; Kaiser, Wilson, & Venning, 2002; Matanović et al., 2012). It is observed that the sand control design is important with respect to the effectiveness of the SAGD system. An initial circulation process involving both wells injecting steam runs for approximately 2-4 months to enhance the oil mobility in the steam chamber (Ahmed & Meehan, 2012). Thus, the sand control design dominates the ability to inject steam during this process, which in turn influences the ability to obtain proper oil mobility.

When employing slotted liners for use in the SAGD process, various design parameters must be specified. The design parameters include the cross-sectional slot profile, slot configuration pattern, slot geometry dimensions, and slot density (Bennion et al., 2009; Ott & Woods, 2003; Xie et al., 2007). As shown in Figure 1.2, slotted liners can be characterized as having one of three slot profiles: a straight cut, a keystone cut, and a rolled top profile. Different manufacturing methodologies are employed to create each of the three profiles. The straight cut profile is created by plunging a saw blade into the liner once, the keystone profile is created by plunging a saw blade into the liner twice at two different angles, and the rolled top profile is created by employing cold rolling to plastically deform a straight cut profile (Bennion et al., 2009).

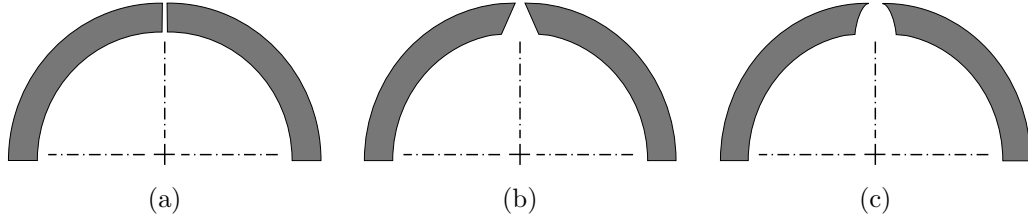


Figure 1.2: Simplified cross-sectional views of the slotted liner with a single slot, which includes the (a) straight, (b) keystone, and (c) rolled top profiles (Bennion et al., 2009; Xie et al., 2007)

The organization patterns used to order the slots along the liner fall within three main categories: staggered, overlapping, and gang (Bennion et al., 2009; Xie et al., 2007). Figure 1.3 provides solid models that showcase each of the three organizational patterns. The staggered pattern consists of slot columns that are evenly spaced about the liner’s circumference, with adjacent columns being offset about the circumference; the lack of an offset between adjacent columns is referred to as a straight pattern. An overlapping pattern consists of evenly spaced slot columns that overlap each other circumferentially. A gang pattern employs the use of multiple groups of slots, with the groups evenly spaced within a column, about the liner circumference. A gang pattern may also employ the use of an offset to form a staggered pattern (Xie et al., 2007).

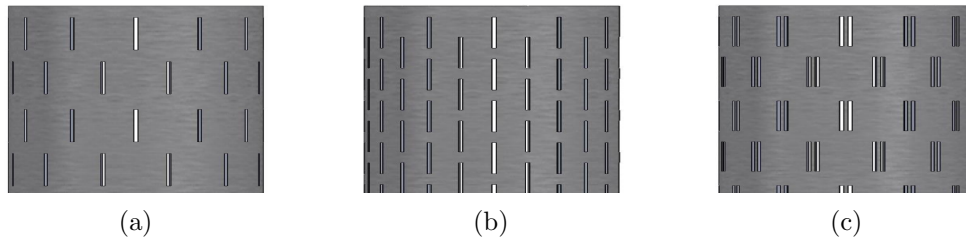


Figure 1.3: The three slot configuration patterns used for slotted liners, which includes the (a) staggered, (b) overlapping, and (c) gang configurations

In terms of slot geometry, the length and width of a slot may be adjusted through the adjustment of the saw blade penetration depth and the saw blade thickness, respectively. The length and width dimensions employed for a given SAGD operation is dependent upon the sand size distribution found at the SAGD operation site; the slot geometry is selected so that the slot is smaller than 90% of the onsite sands (Renpu, 2011). As such, typical slot length and width ranges are 50-70 mm and 0.31-6.4 mm, respectively (Ott & Woods, 2003; Xie et al., 2007). Along with the design parameters, the total number of slots along a liner is a parameter that may be adjusted so that both the slot density and total open area may

be adjusted. Overall, careful consideration of the design parameters results in a product that can appropriately address the sand control needs of a SAGD operation. However, it is clear by observation that the resulting product will also affect the way in which fluids flow in the nearby, surrounding region.

The near-well region of the injection well involves the formation of a complex flow field that is subject to change during the SAGD system's operation. The growth of the steam chamber, the transportation of the bitumen and water to the production well, and the movement of oil sands are observed to be factors that lead to changes in the near-well flow field. As such, the near-well flow field is dynamic with time. That is, the flow changes as the conditions within the SAGD operation site change. Furthermore, if slotted liners are employed for sand control purposes, then the flow field is rendered more complex, and any changes to the slotted liner itself over time will also factor into how the flow field changes. The slotted liner itself will likely change, as it is common for slots to fall victim to erosion and corrosion (Matanović et al., 2012; Ott & Woods, 2003). In spite of such a complex flow arising, it is observed that the most desirable outcome is one where the most bitumen is obtained during the lifetime of the SAGD operation. In order to do so, it is also observed that a large interface between the injected steam and the surrounding oil sands needs to be promoted. Under such conditions there is a greater likelihood of bitumen being available for collection at the production well. Thus, the promotion of favorable flow conditions at the exit of a slotted liner is vital for optimal SAGD operation.

1.2 Conclusions

This discussion highlights the complexity of the near-well flow field in a SAGD operation. In order to model and predict the flow, simplifications to the SAGD system must be made. Fundamentally, it is important to first characterize the flow exiting a single slot. By doing so, the ways in which manipulated variables affect the exiting flow can be best observed and quantified. Following such a fundamental study, further studies in which the conditions are slowly changed to resemble those of the SAGD process can then be performed; doing so will allow for a confident understanding in the nature of the flow exiting a slotted liner. In turn, the information can be used to make recommendations towards the optimization of the SAGD process.

The flow exiting a single slot can be best simplified and modeled as a free jet flow. Specifically, the flow exiting a straight slot profile can be represented by a free, rectangular jet configuration. Thus, this thesis aims to obtain a fundamental understanding of the flow

exiting from such a configuration. The following chapter provides a literature review of the ideas that are relevant to the study of rectangular jets, puts forth the objectives of this study with respect to the presented ideas, and provides an overview of the remainder of the thesis.

Chapter 2

Literature Review

2.1 Overview of Jets and Jet Applications

Many studies have been conducted to understand jet flows with the aim of applying such knowledge to different engineering systems. Such applications include, but are not limited to the following: propulsion technology for aircrafts, pollutant effluent dispersion, ventilation and air conditioning, drying of textiles, annealing of glass, processing of metals and glass, gas turbine and combustor cooling, electronic equipment cooling, and tissue freezing in cryosurgery (Malmström, Kirkpatrick., Christensen, & Knappmiller, 1997; Quinn, 1992b; Viskanta, 1993). Popular jet configurations that have been investigated for such applications include free jets, planar jets, wall jets, and impinging jets (Rajaratnam, 1976; Tesar, 2015). For these configurations, the flow regimes have ranged from creeping flow to supersonic flow, as necessary for the application purpose. When observing such configurations, particular focus has been placed on analyzing how the jet evolves downstream when subjected to various exit conditions. It has been recognized that the downstream flow field is affected by the upstream exit conditions, with exit shape (Mi, Nathan, & Luxton, 2000), exit cross-sectional profile (Nathan, Mi, Alwahabi, Newbold, & Nobes, 2006; Tsuchiya & Horikoshi, 1986), and exit velocity characteristics (Xu & Antonia, 2002) being identified as contributors to the flow's evolution.

When specifically considering the exit from which the jet emerges, many classifications for the shape may be specified. In particular, the exit's shape may be classified as being circular or non-circular; the bulk of exit shapes fall within the latter category. Non-circular shapes that have been investigated include but are not limited to ellipses, rectangles, triangles, and cruciform-type shapes (Gutmark & Grinstein, 1999). When dealing with non-circular shapes, two sub-classifications may also be considered: smooth-edged and sharp-edged. The sharp-edged designation is for profiles that contain corners. Thus,

elliptical configurations fall within the first category, while the rest of the aforementioned shapes fall within the latter. As displayed by the preceding overview, rectangular jet configurations account for a small sub-category in the overall study of jets. However, the study of rectangular jets is in itself a large field of study.

2.2 The Rectangular Jet

2.2.1 Exit Geometry

Rectangular jets emerge from non-circular, sharp-edged exits. Figure 2.1 shows a schematic of the rectangular exit in relation to the three-dimensional coordinate system adopted for this study. The length of slot exit (l) corresponds to the exit's long end, which is orientated in the spanwise direction, while the width corresponds to the short end of the slot exit (h), which is oriented in the lateral direction. The origin point is located at the centre of the slot exit plane, with the X , Y , and Z axes corresponding to the streamwise, lateral, and spanwise directions, respectively. When X , Y , or Z are rendered dimensionless, their nondimensional counterparts may be referred to as X^* , Y^* , and Z^* , respectively.

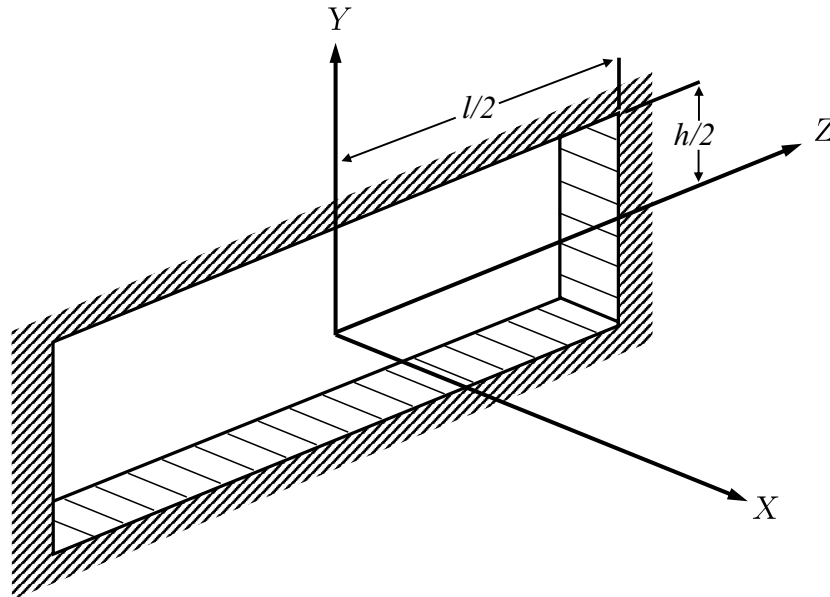


Figure 2.1: Schematic representation of a rectangular exit with the origin located at its centre, and the orientation of the X , Y , and Z axes specified

Although Figure 2.1 conveys a rectangular exit as having an upstream cross-sectional profile similar to that of an orifice plate, it is important to note that other profiles may be adopted for study. Both smooth contractions (Quinn, 1992b) and narrow channels (Sfeir,

1979) constitute other cross-sectional profiles that may be employed for use. With these cross-sectional profiles it is also important to note that their design specifications may vary. An orifice plate may be made to have a bevel at its exit (Sfeir, 1979). The profile of a smooth contraction may be based upon a polynomial function (Quinn, 1992b) or a contraction ratio (Krothapalli, Baganoff, & Karamcheti, 1981). A narrow channel may be manufactured to have a radial contour at its entrance (Sfeir, 1979). Regardless of the cross-sectional geometry employed for use, it must be held constant when investigating the influence of other exit conditions, as any changes in the cross-sectional profile will influence the downstream flow's evolution (Sfeir, 1979; Tsuchiya & Horikoshi, 1986).

2.2.2 Definition of Aspect Ratio for a Rectangular Jet

When studying rectangular jets, the aspect ratio is a dimensionless number that is used to characterize the changes in the exit geometry. By considering the exit's width and length, the aspect ratio can be expressed as:

$$AR = \frac{l}{h} \quad (2.1)$$

In their discussion of three-dimensional jets, Rajaratnam (1976) explains that the AR of a jet defines whether or not the jet is classified as being slender or bluff. The AR of a bluff jet is defined as being near unity - that is, the length and width are near equal to each other. Comparatively, the AR of a slender jet is defined as being far from unity. Based on the work conducted by Yevjevich (1965), the cut-off point for a jet to be considered slender is not definitive, but is estimated to be at an $AR \approx 5$. The importance in the classification of these two jets lies in the observation that slender jets exhibit the three decay regions described by Sforza, Steiger, and Trentacoste (1966), which is discussed later in Section 2.2.4, whereas bluff jets only exhibit two of the regions.

2.2.3 Definition of Reynolds Number for a Rectangular Jet

Along with AR , Reynolds number is another dimensionless number that is used in the study of rectangular jets to characterize changes in the flow conditions. In general terms, the Reynolds number can be expressed as:

$$Re_L = \frac{\rho UL}{\mu} \quad (2.2)$$

where ρ is the working fluid density, U corresponds to the working fluid velocity, L is a characteristic length relevant to the flow under investigation, and μ is the dynamic viscosity

of the working fluid. When considering free jet flows, the working fluid is the jet itself, and its properties are based on those at the slot exit. As a result, the velocity in Equation 2.2 can be specified as being the jet’s exit velocity, U_0 .

Considerations Towards the Jet Characteristic Length

Unlike the other three variables appearing in Equation 2.2, the choice of the lengthscale relevant to rectangular jet flows has varied among researchers. Two popular lengthscales have been used by researchers for rectangular jets: the width of the exit (h) and the equivalent diameter of the exit (D_e). The equivalent diameter refers to the diameter of a round slot having the same exit area as a rectangular slot. Thus, the equivalent diameter can be expressed mathematically as:

$$D_e = \sqrt{\frac{4A_{\text{rec}}}{\pi}} \quad (2.3)$$

where A_{rec} is the exit area of the rectangular slot. Early in the study of rectangular jets, researchers used h as the characteristic length (Krothapalli et al., 1981; Marsters & Fotheringham, 1980; Sforza et al., 1966; Sforza & Trentacoste, 1967; Tsuchiya & Horikoshi, 1986). As a result, Equation 2.2 may be modified so that the Reynolds number for their studies is defined as:

$$Re_h = \frac{\rho U_0 h}{\mu} \quad (2.4)$$

Conversely, in more recent years, some researchers have instead opted to use D_e as the characteristic length for their studies (Barber, Saediamiri, & Birouk, 2017; Lozanova & Stankov, 1998; Mi, Deo, & Nathan, 2005; Quinn, 1992b). Thus, their studies modified the Reynolds number in Equation 2.2 so that it was instead defined to be:

$$Re_{D_e} = \frac{\rho U_0 D_e}{\mu} \quad (2.5)$$

It is observed that the transition from using h to D_e for the relevant lengthscale was in response to the work completed by Hussain and Husain (1989). In their study, the authors used the mean velocity decay, centerline turbulence intensity, and equivalent jet width of elliptical jets to justify the adoption of the D_e as a lengthscale. It was found that using h resulted in the elliptical data displaying a known trend: the three velocity decay regions displayed by rectangular jets (discussed later in Section 2.2.4). With such trends being observed, it was in turn noticed that the data failed to collapse onto a single line downstream of the exit in the far-field. Conversely, by using D_e as the relevant lengthscale, the data plots for the velocity decay collapsed onto each other and also collapsed onto the data of an

axisymmetric configuration in the far-field. The same was also found to be the case in the far-field for the equivalent jet width, and the centerline turbulence intensity. Thus, it was found that the equivalent diameter was suitable for an AR range varying from 1.5 to 8. It is important to note that the authors argued that for higher AR s an elliptical jet is expected to behave more like a planar jet, in which case h proves to be a more suitable lengthscale than D_e . However, the aspect ratio cut-off was not investigated by the authors, nor was it investigated by other authors who adopted D_e as a lengthscale for their studies (Barber et al., 2017; Lozanova & Stankov, 1998; Quinn, 1992b).

2.2.4 Studies on the Effects of Aspect Ratio

Overview of the Aspect Ratio Ranges Previously Investigated

Of the works completed by other experimentalists, ten major studies have been completed where the influence of AR on free, rectangular jets was observed. An overview of these studies is provided in Table 2.1, where the AR range, Re range, downstream measurement range, and cross-sectional exit profile conditions are specified. With these studies, the adopted lengthscale is specified. It is important to note that the work of Sfeir (1979) opted to use the momentum thickness of the jet when considering the jet's velocity decay, but used the width of the jet for other jet parameters. When considering the previous investigations, it is clear that the AR ranges have remained relatively low. Of the ten studies, six studies have had a maximum AR of 20 or less, and only one study has had an AR greater than 100. It is observed that the lack of research on high AR configurations is likely due to a lack of clear application. This is best observed to be the case in the work of Mi et al. (2005), where relatively high aspect ratios were investigated, yet no indication of applicability for such configurations was given.

Table 2.1: Summary of the aspect ratio, Reynolds number, measurement range, and cross-sectional exit for rectangular jet studies

Author	AR	Re	Downstream Range	Exit Type
Sforza et al. (1966) [†]	1.00, 10.00, 25.00, 40.00	$2.60 \times 10^4 - 8.80 \times 10^4$	$0 \leq X^* \leq 350$	Orifice plate
Sforza and Trentacoste (1967) [†]	1.00, 2.50, 5.00, 10.00, 25.00, 40.00	2.60×10^4	$0 \leq X^* \leq 350$	Orifice plate
Sfeir (1979) [*]	10.00, 20.00, 30.00, 40.00, 50.00	$3.70 \times 10^3 - 1.22 \times 10^4$	$0 \leq X^* \leq 125$	Sharp-edged orifice plate, Radially-curved channel
Marsters and Fotheringham (1980) [†]	3.39, 6.44, 9.07, 11.88	$3.30 \times 10^4 - 5.77 \times 10^4$	$0 \leq X^* \leq 120$	Sharp-edged orifice plate
Krothapalli et al. (1981) [†]	5.50, 8.30, 12.50, 16.70	1.20×10^4	$0 \leq X^* \leq 115$	Radially-curved channel
Tsuchiya and Horikoshi (1986) [†]	1.00, 1.50, 2.00, 4.90, 5.00	$6.60 \times 10^3 - 4.00 \times 10^4$	$0 \leq X^* \leq 50$	Smooth nozzle contraction, Orifice plate, Pipe (channel)
Quinn (1992b) [‡]	2.00, 5.00, 10.00, 20.00	2.08×10^5	$0 \leq X^* \leq 63$	Sharp-edged orifice plate
Lozanova and Stankov (1998) [‡]	3.00, 4.76, 4.78, 6.67, 10.00	$1.00 \times 10^5 - 2.16 \times 10^5$	$0 \leq X^* \leq 33$	Smooth nozzle contraction, Orifice plate
Mi et al. (2005) [‡]	15.00, 30.00, 60.00, 120.00	1.00×10^5	$0 \leq X^* \leq 40$	Smooth nozzle contraction
Barber et al. (2017) [‡]	2.00, 3.00, 4.00, 5.00, 6.00, 7.00	$1.00 \times 10^4 - 5.00 \times 10^4$	$0 \leq X^* \leq 9.5$	Orifice plate

[†]: Denotes a study in which h was used as the characteristic length.

^{*}: Denotes a study in which momentum thickness was used as the characteristic length.

[‡]: Denotes a study in which D_e was used as the characteristic length.

Velocity Decay Regions

Collectively, the works of Sforza et al. (1966) and Sforza and Trentacoste (1967) provided the first key insights on the behavior of rectangular, free jet flows. In both studies, the authors observed how changing the ratio of the short edge to the long edge (eccentricity, e) influenced the downstream nature of the jet. The major observations made by the authors were made with regards to the downstream centerline velocity. As shown in Figure 2.2, the authors observed that the centerline velocity exhibits three distinct regions downstream of the exit point (0): the potential core region (PCR), characteristic decay region (CDR), and axisymmetric-type decay region (ATDR).

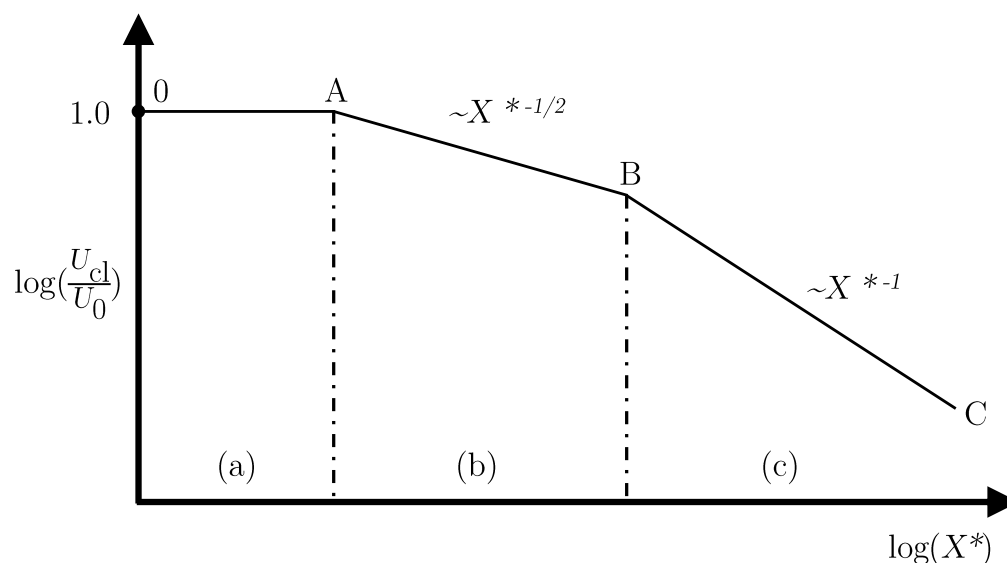


Figure 2.2: Velocity decay of a rectangular jet showing the (a) potential core region, (b) characteristic decay region, and (c) the axisymmetric-type decay region

The naming of the regions by the Sforza et al. (1966) was done with regards to the similarities that the regions exhibit in comparison to other well known jet decay regions. In the PCR, it was observed that the centerline velocity is at unity with the exit velocity, U_0 . Like the development regions of asymmetric and planar jet configurations, the PCR was observed to have a potential core, which is a wedge-like region surrounded by a mixing layer (Rajaratnam, 1976). Sforza et al. (1966) observed that the CDR begins once the jet's lateral shear layers penetrates the centerline and causes the centerline velocity to decay in a way similar to planar free jets. Within this region, it was observed that the jet exhibits similarity in the lateral plane, but does not do so in the spanwise plane. As such, the authors observed that the flow in the CDR is characteristic on the exit conditions.

Further downstream, the ATDR was found to have a velocity decay profile like that of axisymmetric jets, but similarity does not exist when comparing the velocity profiles in the lateral and spanwise planes. Krothapalli et al. (1981) observed that the start of the ATDR is signified by the merging of the spanwise shear layers. Sforza et al. (1966) noted that further downstream, a fully axisymmetric region exists, where the jet behaves completely like an axisymmetric jet, as similarity exists between the cross-sectional planes.

When increasing the aspect ratio of a slot, some changes to the three decay regions have been observed by various authors. It has been found that as aspect ratio nears being at unity, the transition from the jet being in the PCR to the ATDR decreases (Krothapalli et al., 1981; Sfeir, 1979; Sforza et al., 1966; Sforza & Trentacoste, 1967), with a square jet being shown to have a velocity decay almost similar to that of an axisymmetric jet (Sforza et al., 1966; Sforza & Trentacoste, 1967). Conversely, as the AR increases, an overall lengthening of the CDR and shortening of the ATDR has been observed. As a result of the lengthening and shortening of the CDR and ATDR, respectively, many of the previous works have shown the crossing-over of AR data plots at seemingly random points. Again this is a result of the use of h as the lengthscale by early authors, and also the relatively short downstream measurement range by the authors in comparison to Hussain and Husain (1989). Though some minor crossing-over of data occurs between $X = 6D_e$ and $X = 10D_e$, Hussain and Husain's work shows a consistent trend towards the collapsing of data in the far-field (at approximately $X = 50D_e$) rather than the crossing-over at random points seen by others. The work of Mi et al. (2005) also concluded that the same decay regions are exhibited for high aspect ratios, with the same lengthening and shortening of the CDR and ATDR, respectively, having been observed.

Lateral Velocity Profile

When considering the streamwise in-plane velocity profile in the lateral direction, it has been observed that it is characterized by self-similarity within the X - Y plane. By considering the nondimensional velocity profile, U/U_0 , against the dimensionless lateral dimension, Y/h , Sforza et al. (1966) showed that the dimensionless velocity profile expands in the lateral direction as it advances downstream. However, by considering the nondimensional local velocity profile, U/U_{cl} , along with using the the velocity half-width (discussed later) as a similarity parameter, the authors showed that similarity is in fact observable in the lateral plane in both the CDR and ATDR. Similar observations of self-similarity in the lateral direction were found and discussed upon by other authors (Krothapalli et al., 1981; Marsters

& Fotheringham, 1980; Sfeir, 1979). Though not directly discussed by the authors, the works of Tsuchiya and Horikoshi (1986) and Quinn (1992b) provide downstream similarity profiles that appear to further support the observed self-similarity in the CDR and ATDR. With regards to AR , Krothapalli et al. (1981) states that there are no observable changes in the similarity profiles for $AR \geq 10$. Quinn’s work adds to this statement, as it is seen that aspect ratio has a negligible effect somewhere between $AR = 2$ and $AR = 5$. In their work, a clear, observable difference is seen between similarity profiles of $AR = 2$ and the other AR s at $X = 3D_e$ and $X = 5D_e$. When considering the similarity profiles off of the X - Y plane, Marsters and Fotheringham (1980) have shown that there is a consistency in the similarity profiles along the spanwise direction. It was noted by the authors that this is expected to only occur near the centre of jet, as two-dimensional flow exists away from the edges of the jet. No indication, nor any attention was placed by the authors towards how increasing AR may influence the overall extent of the two-dimensional flow region in the spanwise direction, however.

Spanwise Velocity Profile

Unlike the velocity profiles in the lateral direction, the streamwise velocity profiles within the X - Z plane have found to not be characterized by self-similarity. Sforza et al. (1966) identified a "saddle-back" velocity profile exists in the spanwise direction. Though only shown for $AR = 10$ ($e = 0.1$), the profile was observed to have two off-center peaks that were higher in magnitude than the U_{cl} , and these peaks appeared to fade past $X = 30h$. The phenomena was further investigated in the follow-up work of Sforza and Trentacoste (1967), where the saddle-back profile was not found to be due to measurement errors nor orifice plate manufacturing errors. It was further observed that the profile did not exist until the onset of the CDR. It was also noted that the start of the ATDR marked the location where the saddle-back shape ceased to exist, and where the velocity profiles in the X - Y and X - Z planes began to share similarity to one another. In their examination of planar jets, der Hegge Zijnen (1958) proposed that the streamwise velocity was induced by vortex rings that formed about the slot exit perimeter. This induction of velocity was argued by der Hegge Zijnen (1958) to be the root cause for the saddle-back profile. Based upon restrictions in their set-up, Sforza and Trentacoste (1967) were unable to take velocity induction measurements to confirm whether der Hegge Zijnen’s proposition held for free, rectangular jets, or if another mechanism was also responsible for the saddle-back profile.

Following the two aforementioned works on rectangular jets, many other experimen-

talists proceeded in confirming the existence of the saddle-back profile for other rectangular configurations (Marsters & Fotheringham, 1980; Quinn, 1992b, 1994; Quinn, Pollard, & Marsters, 1983; Sfeir, 1979; Tsuchiya & Horikoshi, 1986; Vouros, Panidis, Pollard, & Schwab, 2015). Sfeir’s work established that the saddle-back profile is present for orifice plates, but the peaks are almost completely suppressed for channel configurations. Other authors established that this too is the case for orifices (Marsters & Fotheringham, 1980; Quinn, 1992b; Quinn et al., 1983; Tsuchiya & Horikoshi, 1986; Vouros et al., 2015), and for channel configurations (Krothapalli et al., 1981). It has also been established that smooth-nozzle contractions behave similarly to channel configurations in that the saddle-back peaks are significantly suppressed (Lozanova & Stankov, 1998).

Specific researchers have gone on to attempt to explain the formation of the saddle-back profile. Marsters and Fotheringham (1980) postulated that the induction of fluid from the center to the ends of the jet was partly due to der Hegge Zijnen’s proposed vortex rings, and also the formation of streamwise vortices at the corners of the exit due to nonuniform vorticity shedding upstream of the exit. Though Marsters and Fotheringham could not prove this at the time, the existence of vorticity pairs at the corners was indeed found to be present for sharp-corner geometries (Quinn, 1992a). It was further postulated by Marsters and Fotheringham that the continued influence of these streamwise vortices allows for the two off-center velocity peaks to move towards each other and merge together near the end of the CDR. Such merging tendencies had also been observed by others (Tsuchiya & Horikoshi, 1986), but without explanation of the underlying mechanisms behind it. Others works (Quinn, 1992a; Quinn et al., 1983) have put forth that the vena-contracta leads to the skewing of the shear layer, which in turns leads to four vortex pairs forming at the corners. The formation of these vortices has been explained to be inline with the mechanisms of Prandtl’s secondary flows (Oertel, 2010).

Though not explicitly discussed by the authors, some changes to the saddle-back velocity profile occur with respect to changes with AR . When transitioning from $AR = 3.39$ to $AR = 6.44$ it was found by Marsters and Fotheringham (1980) that the jet went from not exhibiting a saddle-back profile to exhibiting one. Furthermore, it was found in other works (Quinn, 1992b; Tsuchiya & Horikoshi, 1986) that the saddle-back was present for $AR = 5$. Past these observations, the exact cut-off value has not been looked into. When increasing AR , the relative height of the peaks to the centerline velocity are increased (Marsters & Fotheringham, 1980; Quinn, 1992b). With regards to the peaks, the relative value of the peaks have been observed to range from about $1.1U_{cl}$ to $1.2U_{cl}$ (Marsters & Fotheringham,

1980; Quinn, 1992b; Tsuchiya & Horikoshi, 1986).

Lateral and Spanwise Velocity Observations

Unlike the proceeding discussions examining the changes in the streamwise velocity profiles, few studies have gone on to examine the changes in the other velocity components, V and W . Specifically, only three of the ten major studies that have examined the influence of aspect ratio have made such observations (Krothapalli et al., 1981; Lozanova & Stankov, 1998; Quinn, 1992b). Briefly, V exhibits similarity in its profiles far enough downstream, with the amount of induced velocity at the jet boundary decreasing downstream (Krothapalli et al., 1981). The observations of Lozanova and Stankov (1998) are in agreement, with it also being noted that the velocity profiles are higher than those of axisymmetric jets. With regards to W , no similarity is seen downstream of the exit (Krothapalli et al., 1981; Lozanova & Stankov, 1998). The velocity goes from being symmetrically induced towards the centerline at the exit (Quinn, 1992b) to having the degree of induction reducing, and eventually changing sign downstream (Krothapalli et al., 1981; Lozanova & Stankov, 1998), which Lozanova and Stankov (1998) describe as being due to large jet deformations in the spanwise direction.

Centerline Turbulence Intensities

Along with analyzing the average velocity profiles of the flow, researchers have investigated the flows turbulence intensities to further understand the underlying mechanisms of the perceived changes in the average jet flow. Specific attention has been placed on understanding the downstream profile of the streamwise turbulence intensity, u' , on the centerline. Sfeir (1979) was the first researcher to discuss the downstream centerline profile of u' , where they discussed its variations with respect to the three velocity decay regions. In the PCR, there is an initial region where there is a rapid increase in u' as the shear layers near the centerline. Next, upon entering the CDR, u' immediately decreases before beginning to increase again. In the ATDR, u' appears to approach a value around 0.25. Overall, similar trends were observed by other authors (Krothapalli et al., 1981; Quinn, 1992b; Tsuchiya & Horikoshi, 1986), along with insight on how u' changes with respect to cross-sectional exit shape, boundary layer state at the exit, and AR . When decreasing the AR , it has been observed that the magnitudes of the initial u' peak and following drop both decrease, with the peak and drop disappearing somewhere between $AR = 5$ and $AR = 2$ (Quinn, 1992b). When instead increasing the AR , it has been observed that the location of peak appears to

move closer to the jet exit (Lozanova & Stankov, 1998). When considering the downstream lateral and spanwise turbulence intensities (v' and w' , respectively), it was found that they follow similar trends to their u' counterparts, albeit at a lower magnitude, and cross-over each other at one point downstream (Lozanova & Stankov, 1998).

Lateral Turbulence Velocity Profiles

Unlike the in-plane velocity components, rigorous investigations on the turbulence velocity profiles have not been made. However, key observations have been made on the nature of the turbulence intensities in the lateral direction. The shape of u' in the lateral direction tends towards a saddle-back shape, with the peaks occurring where the velocity gradient is highest. Both Sfeir (1979) and Krothapalli et al. (1981) have noted that within the CDR, u' exhibits geometrical similarity. Further downstream, the saddle-back distribution of u' subsides as the jet approaches axisymmetry. With regards to v' and w' , it has been observed that both exhibit profiles like those seen in plane jet flows (Krothapalli et al., 1981).

Spanwise Turbulence Velocity Profiles

It was been observed that the shape of u' in the spanwise direction differs appreciable when compared to u' in the lateral direction. Peaks are present at the edges where the velocity gradient is highest, however, in the middle the magnitude of u' rises slightly (Sfeir, 1979; Tsuchiya & Horikoshi, 1986). Further downstream, the u' evolves to have the same saddle-back profile exhibit by u' in the lateral direction. Krothapalli et al. (1981) argues that the emergence of this shape is indicative of the merging of the two shear layers separated by the long dimension of the exit. The authors also observed that v' and w' , have approximately the same values as their lateral direction counterparts.

Velocity Half-Width

Velocity half-width is a measurement length used to describe the spread of the rectangular jet in the X - Y plane and the X - Z plane. In either plane, it is defined as the distance measure from the centerline of the jet where the local mean velocity is equal to half of the local centerline mean velocity. Based on this definition, the velocity half-width in the lateral direction (X - Y plane) may be expressed as:

$$Y_{\frac{1}{2}} = U(X, Y, 0) = \frac{1}{2}U_{cl}(X, 0, 0) \quad (2.6)$$

Similarly, the velocity half-width in the spanwise direction (X - Z plane) is expressed as:

$$Z_{\frac{1}{2}} = U(X, 0, Z) = \frac{1}{2}U_{cl}(X, 0, 0) \quad (2.7)$$

By considering the velocity half-widths in both planes, a simple understanding of the jet's downstream deformation is obtained. Sforza et al. (1966) observed that $Y_{\frac{1}{2}}$ rapidly increases, whereas $Z_{\frac{1}{2}}$ initially decreases and then increases at a rate similar to $Y_{\frac{1}{2}}$. The authors further found that there is a crossing-over of $Y_{\frac{1}{2}}$ and $Z_{\frac{1}{2}}$, which occurs at the onset of the ATDR. The aforementioned trends for the half-widths have also been observed in all of the other studies that appear within Table 3.1. It has been observed that as aspect ratio increases, $Y_{\frac{1}{2}}$ exhibits linear variation, but the slopes of the lines are different for different AR s. In contrast, it was also observed that the variation of $Z_{\frac{1}{2}}$ remains relatively unchanged as AR increases (Krothapalli et al., 1981). Using their experimental data, along with data from the works of Sforza et al. (1966) and Sfeir (1979), Krothapalli et al. (1981) found that the crossing-over point of $Y_{\frac{1}{2}}$ and $Z_{\frac{1}{2}}$ increases linearly as AR increases. Similar trends were observed in the works of Tsuchiya and Horikoshi (1986) and Barber et al. (2017).

Reynolds Shear Stresses

Of the works listed within Table 2.1, five have partially investigated the Reynolds shear stresses of the flow (Krothapalli et al., 1981; Lozanova & Stankov, 1998; Marsters & Fotheringham, 1980; Quinn, 1992b; Sfeir, 1979), with attention being placed towards observing $\overline{u'v'}$ and $\overline{u'w'}$. Sfeir (1979) presented $\overline{u'v'}$ within the X - Y plane and observed good agreement between the measured values and calculated values. The authors noted the peak value $\overline{u'v'}$ to be 2-3% lower than that of a plane jet. Marsters and Fotheringham (1980) presented in-plane measurements of $\overline{u'v'}$ and $\overline{u'w'}$, which were deemed to be convoluted such that the authors could not link the Reynolds stresses to the features of U . Krothapalli et al. (1981) presented $\overline{u'v'}$ and $\overline{u'w'}$ profiles on the X - Y and X - Z planes, respectively. It was established that for $X \geq 30h$, $\overline{u'v'}$ is geometrically similar, with a peak located where the velocity gradient ($\partial U/\partial y$) is highest. Likewise, in the X - Z plane it was found that $\overline{u'w'}$ formed a peak located where the velocity gradient ($\partial U/\partial z$) is highest. The authors also noted that at any X location within the ATDR, $\overline{u'w'}$ in the X - Z plane was higher than $\overline{u'v'}$ in the X - Y plane, but $\overline{u'w'}$ was significantly lower than $\overline{u'v'}$ in the X - Y plane.

Quinn (1992b) also presented $\overline{u'v'}$ and $\overline{u'w'}$ profiles on the X - Y and X - Z planes, respectively, at three downstream locations ($X = 2D_e$, $5D_e$, and $10D_e$). The author showed that the magnitude of $\overline{u'v'}$ and $\overline{u'w'}$ both increase as AR increases. Furthermore, the author

observed that the production due to mean streamwise velocity gradients was the largest contributor to the production of the Reynolds shear stresses. This observation was shared by Lozanova and Stankov (1998), who presented in-plane measurements of $\overline{u'v'}$ and $\overline{u'w'}$.

Turbulent Kinetic Energy

The turbulent kinetic energy is defined to be half the sum of the normal Reynolds stresses. Based on this definition, the turbulent kinetic energy may be expressed as:

$$TKE = \frac{1}{2}(\overline{(u')^2} + \overline{(v')^2} + \overline{(w')^2}) \quad (2.8)$$

Quinn (1992b) presented TKE profiles on the X - Y and X - Z planes, respectively, at three downstream locations ($X = 2D_e, 5D_e,$ and $10D_e$). It was found that the high velocity gradients formed by the saddle-back velocity profile led to the TKE being higher with the X - Z plane than the X - Y plane. The authors also noted that the TKE profiles exhibited for $AR = 2$ were consistently higher than those exhibit for $AR = 5$. By considering the trends in u' , this was explained as being due to the merging of the shear-layers, which caused a decrease in turbulent activity on the centerline, as was the case for $AR = 5$. Like their observations on Reynolds shear stresses, the author observed that the production due to mean streamwise velocity gradients dominates the production of TKE in both the X - Y plane and X - Z plane.

2.2.5 Studies on the Effects of Reynolds Number

In contrast to the work done by previous experimentalists to observe the effect of AR , little work has been done to investigate the effects of Re on the rectangular jet. In their discussion on Re , Dimotakis (2000) states that fully-developed turbulent flows occur at a magnitude of $Re \approx 10^4$. The author further notes that under such a magnitude, the behaviour of a jet is expected to be much different. Considering the works presented within Table 2.1, it is clear that most of the studies are beyond this limit, with the exception of Sfeir (1979) and Tsuchiya and Horikoshi (1986). Clear Re -based differences are presented in the velocity decay plots and downstream centerline u' plots given by Sfeir (1979), but without discussion, nor further investigation towards how Re influences the other flow characteristics. Though recognizing the importance of Re as an initial condition, Tsuchiya and Horikoshi (1986) only show clear Re -based changes within their downstream centerline u' plots. Though the authors do provide some discussion on the affect of Re , further investigation on how Re affects other jet characteristics is nonexistent.

In contrast to the two aforementioned works, the work of Barber et al. (2017) presents a particle image velocimetry (PIV) based study that investigates the affects of both AR and Re . The range of Re was from 1.00×10^4 to 5.00×10^4 , however, which was on the order of magnitude discussed by Dimotakis (2000) needed for fully-developed turbulent flow. As such, after considering the velocity decay and half-widths, Barber et al. concluded that no appreciable changes with Re could be observed. Nonetheless, approaching such a study for high AR configurations is observed to be of interest, as at the very least it allows for the confirmation of Dimotakis' ideas.

2.2.6 Further Concepts Related to the Study of Rectangular Jets

Vorticity Dynamics

As exemplified by the preceding discussion, many authors have tried to explain observed jet phenomena as being the result of vorticity-based phenomena. The work of Zaman (1996) provides a thorough overview of two vorticity types that appear natural in rectangular jet flows: azimuthal vorticity rings and streamwise vortices. In their work, the authors refer to the azimuthal and streamwise vortices as ω_θ - and ω_x -dynamics, respectively. Based upon the work of Hussain and Husain (1989), Zaman explains that a rolling-up of the azimuthal vorticity near the jet exit is what forms the azimuthal vorticity rings. In agreement with the work of Quinn (1992a), Zaman also explains the streamwise vortices to be a form of Prandtl's first kind of secondary flow caused by lateral pressure gradients. Furthermore, Zaman states that these two vorticity dynamics are not necessarily independent of each other, with the ω_θ -dynamics being observed with helping the formation of the ω_x -dynamics at the jet exit. Furthermore, their continued interactions downstream are noted as being the reason for the jet's cross-sectional deformation, as discussed in the proceeding discussion.

Jet Cross-sectional Shape

As a result of the complex interaction of the ω_θ - and ω_x -dynamics, there are appreciable changes in the jet's cross-sectional shape. The velocity half-width has proven to be a useful tool for researchers to understand the change in the jet's shape. However, in-plane measurements have gone on to supplement the information obtained from streamwise velocity-half measurements. A crossing-over of velocity half-widths has been observed in all of the studies outlined within Table 2.1. Upon further investigation through in-plane measurements, (Sforza & Trentacoste, 1967) found that the jet's major and minor axes appeared to switch places with each other. Other cross-sectional measurements also revealed the apparent

switching of the two axes (Marsters & Fotheringham, 1980; Tsuchiya & Horikoshi, 1986). The apparent switching of the two axes of a jet is referred to as axis-switching, and has been observed for many non-circular jets, as discussed by Gutmark and Grinstein (1999).

The concept of axis-switching was thoroughly examined by Zaman (1996), and was explained to be due to both ω_θ - and ω_x -dynamics. Zaman described the self-inductive nature of the ω_θ -dynamics to be the main mechanism behind the axis-switching phenomenon. In comparison, it was noted that ω_x -dynamics could stand to either enhance or resist the likelihood of the occurrence of axis-switching. Zaman showed that the change in the natural position and number of vorticity pairs (via the use of tabs and acoustic excitation) was what would lead to the ω_x -dynamics either enhancing or resisting the ω_θ -dynamics' tendency towards axis-switching. Of note, in their work on a square configuration, Quinn (1992a) found that only a 45° turn occurred rather than the typical 90° turn observed by Zaman (1996) for an $AR = 3$. Also of note, Vouros et al. (2015) showed that the deformation of the jet via vorticity can lead to it having a cross-sectional "dumbbell" shape for an $AR = 10$. Based on the two aforementioned studies, it is observed that AR may also influence the overall deformation of the cross-sectional jet shape. As such, the acquisition of cross-sectional measurements is viewed to be necessary, as it allows for a deeper understanding of the downstream changes of a rectangular jet.

2.3 Measurement Techniques for Jet Experiments

Within fluid mechanics, there are many measurement techniques to determine flow velocity. Specifically, the measurements may be taken using pressure-based systems, thermal anemometry, or particle-based techniques (Tropea, Yarin, & Foss, 2007). With free jet flows, many authors have taken to using hot-wire anemometry (HWA) as a means to measure the proceeding jet flow. This has been the case for most of the aforementioned studies investigating the influence aspect ratio on rectangular jets. Apart from HWA, some of these studies have instead relied on the use of pressure-probe anemometry (PPA) in the form of pitot-tubes and manometers to obtain their measurements. The measurement techniques of the experiments presented in Table 2.1 are summarized within Table 2.2.

Though much insight has been made on the nature of rectangular jets using HWA and PPA techniques, some of their shortcomings are causes for concern in their application. One of the greatest setbacks of is the inability to take proper measurements where flow reversals are present. In such cases, flying HWA techniques have been proven to be suitable for application (Coles, Cantnell, & Wadcock, 1978; Thompson & Whitelaw, 1984; Watmuff,

Table 2.2: Summary of the measurement techniques used for the experimental investigations on the effects of aspect ratio on a rectangular jet

Author	Measurement Technique
Sforza et al. (1966)	PPA
Sforza and Trentacoste (1967)	PPA
Sfeir (1979)	HWA
Marsters and Fotheringham (1980)	PPA, HWA
Krothapalli et al. (1981)	HWA
Tsuchiya and Horikoshi (1986)	PPA, HWA
Quinn (1992b)	HWA
Mi et al. (2005)	HWA
Barber et al. (2017)	Two-dimensional PIV

Perry, & Chong, 1983). It is observed however, the drawback of this technique is the higher complexity in applying the technique. In agreement, Deo (2005) notes that PIV is simpler in use than when flying HWA techniques are necessary. Another setback of HWA and PPA studies is their need to rely on visualization techniques in order to help supplement their data and understand the underlying phenomenon. Such visualization techniques have included dye injection photography (Sforza & Trentacoste, 1967), Schlieren photography (Krothapalli et al., 1981), planar laser induced fluorescence (PLIF) imaging (Gutmark, Schadow, Parr, Hanson-Parr, & Wilson, 1989). The need for a secondary system is observed to be a hindrance as it calls for higher experimental costs, and longer setup and experimentation times. Furthermore, the intrusive, and timely point-source nature of HWA and PPA techniques are seen to be other drawbacks.

For the aforementioned and undermentioned reasons, attention was placed onto utilizing PIV techniques for this study. The largest observed benefit to using PIV was in it being a non-intrusive, whole-field measurement technique. By being a whole-field measurement, the entirety of the jet can be easily captured, unlike in previous studies where only portions

of the jet could be obtained due to the long collection times (Marsters & Fotheringham, 1980; Sforza & Trentacoste, 1967; Tsuchiya & Horikoshi, 1986). Initially, the ability to apply PIV techniques to jet flows was brought into question. However, it was observed that PIV techniques have been successfully applied to not only rectangular configurations, but also non-rectangular configurations (Barber et al., 2017; Hassan & Meslem, 2010; Mi, Kalt, Nathan, & Wong, 2007). Finally, as shown in Table 2.2, only one study on rectangular jet flow has been presented in the literature (Barber et al., 2017). However, the study was not conducted for high aspect ratios, nor was the analysis as rigorous as some of the work done by other experimentalists listed in Table 2.1. Thus, the overall application of PIV techniques for the study of high aspect ratio slots presents itself to be a novel study with novel measurement techniques in the field of rectangular, free jets.

2.4 Research Objectives

The objective of this thesis is to observe and provide insight on the development of high AR rectangular jets in the near-exit region. The preceding chapter shows that an understanding of such flows is necessary as they have applications to the SAGD process, and the preceding sections within this chapter highlight the lack of experimental work that has been accomplished for such flow configurations. With respect to the work accomplished by other experimentalists, it is of interest to determine if such configurations continue to have similar trends as those seen for lower aspect ratios. As little work has also been done on the influence of Re , it is also of interest to observe what changes occur with respect to Re , and whether the $Re_{\text{critical}} \approx 10^4$ proposed by Dimotakis (2000) holds for such configurations.

In order to observe the downstream jet flow, PIV techniques will be applied. In particular, both two-dimensional PIV (2D-PIV) and stereoscopic PIV (stereo-PIV) techniques will be used to observe the downstream flow in the X - Y plane and the Y - Z plane, respectively. Due to the unique testing conditions imposed by the ranges chosen for the manipulated variables, a minor objective will also be the successful application of each PIV technique.

2.5 Thesis Outline

Following the preceding discussion, the rest of the thesis contains five chapters. An overview of the chapters and a summary of their content is provided, as follows:

Chapter 3 details this study’s experimental setups. First, a discussion of the slot manufacturing methodology is presented; within this discussion, the motivation and consider-

ations towards developing a slot manufacturing methodology are provided, along with a description of the methodology and the chosen experimental slot parameters. Next, and description of the experimental flow plenum is provided, with specific details on the flow conditioning parts. A brief overview of PIV is then provided before the descriptions of the 2D-PIV and stereo-PIV setups are given. Finally, the flow line arrangement and the considerations towards its setup are discussed.

Chapter 4 discusses the data processing methodology employed for this study. Descriptions of the processing steps employed for the 2D- and stereo-PIV data are given. Following this, explanations of further post-processing steps performed in MATLAB are provided. The chapter finishes with a discussion of the sources of uncertainty within the experiments, along with an estimation of their values.

Chapter 5 is dedicated to the presentation of the work done to examine the effects of AR on the rectangular, free jet. First, the streamwise flow field observations are discussed, which includes velocity decay, turbulence intensity, and lateral velocity half-width. Afterwards, the cross-sectional observations are discussed, which includes the velocity components, turbulence intensities, shear stresses, turbulent kinetic energy, vorticity, and jet boundary deformation. The chapter closes with conclusions based on the experimental observations.

Chapter 6 is dedicated to the presentation of the work done to examine the effects of Re on the rectangular, free jet. The format of this chapter is similar to that of the preceding chapter. The streamwise flow field observations are discussed first, which includes velocity decay, turbulence intensity, and lateral velocity half-width. Then the cross-sectional observations are discussed, which includes the velocity components, turbulence intensities, shear stresses, turbulent kinetic energy, vorticity, and jet boundary deformation. The chapter closes with conclusions based on the experimental observations.

Chapter 7 serves to summarize the conclusions made based on the experiments of this study. Following these conclusions, recommendations for future work are given.

Along with the aforementioned chapters, appendices have been added at the end of the thesis to provide supplementary material. These appendices are:

Appendix A, which provides the drawing package of the 3D-printed components of the flow plenum that was used for the experimental setup.

Appendix B, which provides the MATLAB scripts used for further post-processing of the 2D-PIV data. This includes the scripts that were used to stitch the data sets together,

along with the scripts that were used to generate the figures presented for this study.

Appendix C, which provides the MATLAB scripts used for further post-processing of the stereo-PIV data. This includes the scripts that were used to find the jet's boundary, and the scripts that were used to generate the results figures.

Appendix D, which stands to provide all of the 2D-PIV figures generated for this study, which includes: velocity decay plots, turbulence intensity plots, lateral velocity half-width plots, whole-field velocity colormaps, and whole-field velocity profile maps.

Appendix E, which stands to provide all of the stereo-PIV figures generated for this study. This includes whole-field figures for: all three average velocity components, all three RMS velocity components, Reynolds shear stresses, turbulent kinetic energy, and vorticity.

Chapter 3

Experimental Setup

3.1 Introduction

This chapter presents the experimental setups used to complete this study. All aspects of the setup are discussed, which includes: the experimental slot assembly, flow conditioning assembly, PIV setups, and flow line arrangement. For each aspect of the setup, the design considerations are elaborated upon, as necessary. Two PIV setups were utilized to make observations on the rectangular jets of interest. A 2D-PIV setup was used to investigate the changes in the jet characteristics in the X - Y plane. A stereo-PIV configuration was in turn used to observe cross-sectional changes in the jet at three downstream plane locations, $X = 3D_e$, $5D_e$, and $7D_e$. After discussing the 2D- and stereo-PIV configurations, the chapter ends with a discussion on the flow line arrangement used to supply seeded air to the system.

3.2 Flow Facility Setup

This study's experiments were completed in the Applied Thermofluids Laboratory located within the Mechanical Engineering Building at the University of Alberta. The overall experimental setup used for this study can be broken down into two main setup types: the flow facility setup, and the PIV setup. The flow facility setup itself contained two main sub-assemblies: the experimental slot sub-assembly, and the flow conditioning sub-assembly. This section provides an overview of the two sub-assemblies, with discussions of the considerations that were taken towards the final designs presented herein.

3.2.1 Experiment Slot Assembly

The test section, which would simulate a single slot, was the first aspect of the entire setup design that was considered. The slot design was considered first as its design would affect

the overall design of the flow conditioning assembly, which would have the partial function of holding the slot assembly in place.

Initial Considerations

Initially, test pieces referred to as *coupons* were provided by RGL Reservoir Management Inc. As shown in Figure 3.1, a coupon is a circular piece of metal with a single slot cut into it. The slot manufacturing process is the same as for slotted liners, so it was originally of interest to use the coupons for experimentation. However, further inspection of the coupons revealed flaws in the coupon design that made them unsuitable for experimental use.

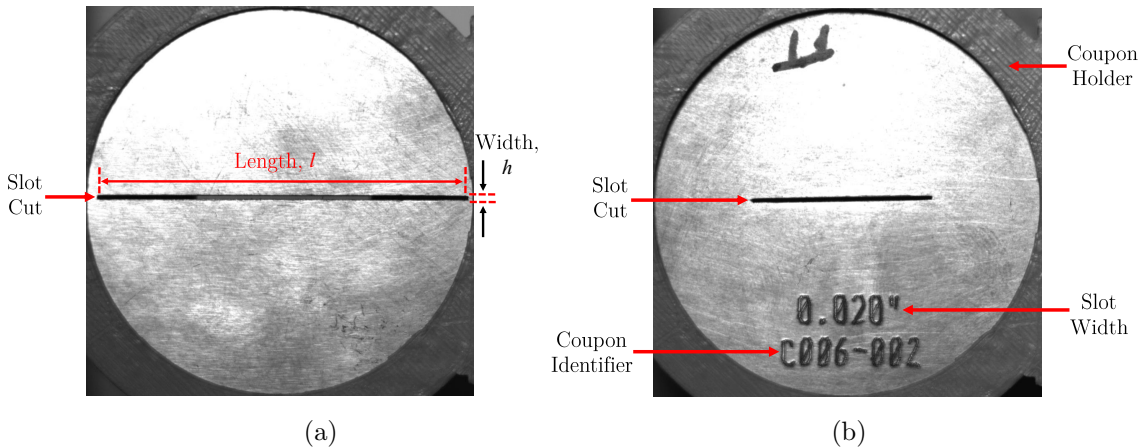


Figure 3.1: Images of a coupon showing (a) an unlabelled front side with a large AR slot, and (b) a labelled back side with a reduced AR slot

Four observations were made about the coupons that ultimately led to a desire for developing an in-house slot manufacturing methodology. First, it was observed that there were inconsistencies in the manufacturing process that resulted in many slots having appreciable deviations between their stated and actual dimensions. Secondly, it was observed that the saws made the coupons have slots with an elliptic shape rather than a rectangular shape. Thirdly, the saws were only plunged into one side of the coupons, so it was observed that each coupon's initial AR reduced across the coupon's cross-section, as shown in Figure 3.1. Finally, it was observed that the round design of the coupon posed an inherent difficulty in ensuring slot alignment. Based on these observations, it was of desire to develop a rapid manufacturing methodology that could:

- Obtain consistent length and width dimensions
- Form a proper, sharp-cornered rectangular exit

- Have a consistent AR across the test piece's cross-section
- Allow for proper alignment of the slot with the rest of the setup

Development Towards a Manufacturing Methodology

While developing a slot manufacturing methodology, both laser-cutting and 3D-printing technologies were considered for application for this research. The first slot manufacturing methodology that was developed involved the use of a commercial laser cutter (VersaLaser, Universal Laser Systems) to cut test pieces from acrylic sheets. When using the laser cutter, it was observed that the minimum possible slot size was dependent on the laser beam diameter. The minimum possible slot size was found to be on the order of a few millimeters, which was an order of magnitude larger than what is typical for SAGD slots. As this was found to be the case, it was decided that the laser cutter was ill-suited for slot manufacturing.

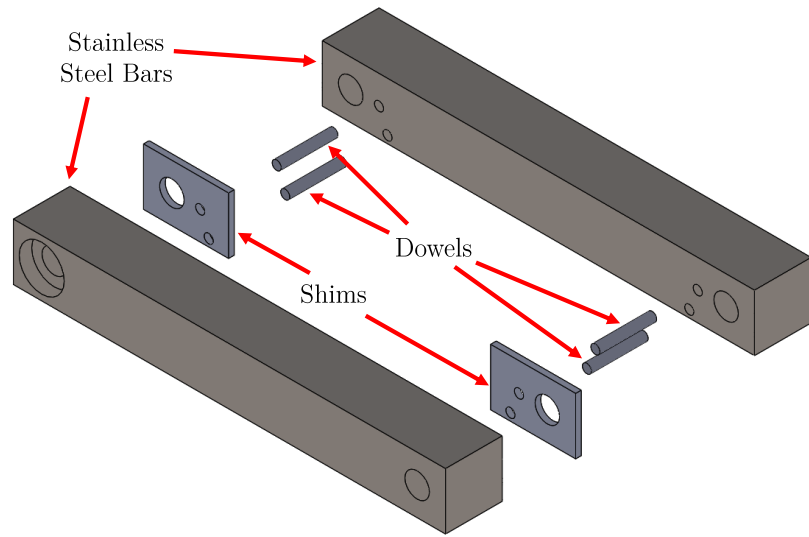
The second slot manufacturing methodology involved using a stereolithography-based (SLA) 3D-printer (Form2, Formlabs, Inc.) to print slots made of clear, photopolymer resin. As per the printer's specifications, a minimum opening size of 0.5 mm was stated to be possible for print jobs. Such a value is on the same order of magnitude that is typical for slot design. However, when slot dimensions below 1 mm were prescribed for printing, it was found that the printed slot dimensions had appreciable deviations from their prescribed values. It was determined that the majority of the dimension deviations were products of the print orientation. Through careful consideration of the print orientation, the dimension deviations could be reduced. Nevertheless, some dimension deviations were found to continue to exist. These dimensional deviations were found to be due to an inherent randomness in the printing process itself, and could not be minimized. The dimensional deviations that persisted were large enough that it was decided that the SLA 3D-printer was also unsuitable for slot manufacturing.

Manufacturing Methodology

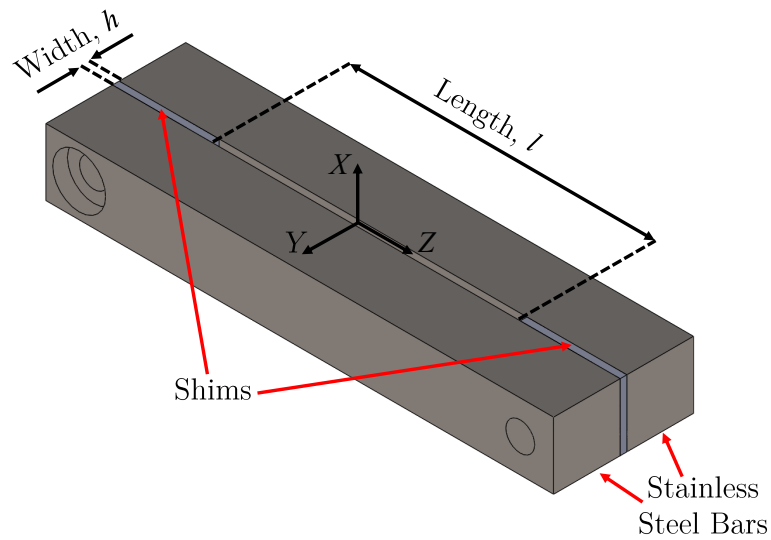
The methodology that was ultimately developed to make the slot employed a novel, shim-based design¹. As shown in Figure 3.2a, the slot was formed by placing plastic shims (Shim Stock, Artus Corp) of controlled thickness between two stainless steel bars. A shim cutter (Cricut Explore AirTM 2, Provo Craft and Novelty, Inc.) was used to cut the shim stock to the desired geometry. The shim stock geometry was designed using the shim cutter's

¹This methodology was developed by colleague Dr. Yishak Yusuf

company's online software, which was also used to prescribe the cut design to the cutter. By cutting shim stocks of various thicknesses and then stacking the cut shims, it was possible to obtain the narrow widths needed to make the aspect ratios of interest. Through this approach, a minimum slot width of 0.01 mm could be obtained, and the slot length could be ensured to within 0.01 mm of its prescribed value. Small stainless steel dowel pins were used to maintain the alignment of the shims with the stainless steel bars. Once the stainless steel bars, dowels, and shims were mated, as shown in Figure 3.2b, two screws were used to keep the entire coupon assembly together (not shown).



(a)



(b)

Figure 3.2: Three-dimensional models of the slot assembly (a) in an exploded isometric view, and (b) mated together in an isometric view

Experimental Slot Parameters

With a slot manufacturing methodology developed, attention was then placed towards determining appropriate experimental slot dimensions. In order to do so, slot aspect ratios and areas were calculated using typical slot dimensions provided by RGL Reservoir Management Inc. From the provided dimensions, minimum and maximum aspect ratios of 83 and 201 were calculated, respectively. Additionally, slot areas were calculated to be within a range spanning from 11 mm² to 27 mm². By using both of the calculated ranges as guidance, appropriate slot dimensions were chosen, and are shown within Table 3.1.

Table 3.1: Dimensions, desired aspect ratio, actual aspect ratio, actual area, and actual D_e for the slot configurations considered in this study

Length [mm]	Width [mm]	Desired AR	Actual AR	Actual Area [mm ²]	Actual D_e [mm]
34.64	0.35	100	98.97	12.12	3.93
42.43	0.28	150	151.54	11.88	3.89
48.99	0.25	200	195.96	12.25	3.95
50.20	0.24	210	209.17	12.05	3.92

The work of Deo, Nathan, and Mi (2010) showed that area affects downstream jet flow, so the slot area was held at 12 mm², which resulted in an D_e of approximately 3.91 mm. At this area, the lengths and widths corresponding to aspect ratios of 100, 150, 200, and 210 were determined. The slot manufacturing methodology ensured the length and width values to be within 0.01 mm of the desired value. Consequently, the final slot configurations had slight deviations from the desired values. Thus, Table 3.1 also details the actual aspect ratio, area, and equivalent diameter value for each slot configuration.

3.2.2 Flow Conditioning Assembly

With the slot assembly made, it was necessary to design and manufacture a plenum that would hold the slot assembly in place while also conditioning the upstream flow. The plenum, which is referred to as the flow conditioning assembly, was made to reduce swirl and turbulence fluctuations existing in the upstream flow. By conditioning the upstream flow, it is ensured that the observations on the downstream jet flow are a product of the manipulated variables (AR and Re) rather than the upstream flow conditions. Figure 3.3 shows a cross-sectional solid model schematic of the flow conditioning assembly that was ultimately designed, manufactured, and assembled.

As shown in Figure 3.3, the flow conditioning assembly was made to include five flow

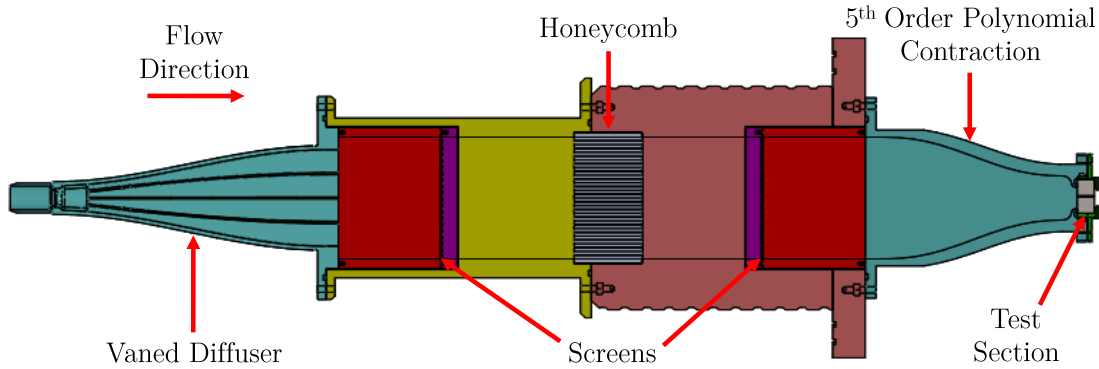


Figure 3.3: Cross-sectional view of the flow conditioning assembly with the test slot installed

conditioning components: a vaned diffuser, two screens, a honeycomb, and a 5th order polynomial contraction. In order to make the flow conditioning assembly, 3D printed parts were designed, fabricated, and assembled together. Each of the parts were printed from ABS material using a commercial 3D-printer (Ultimaker 2+, Ultimaker) that utilized fused filament fabrication (FFF) techniques. The design of the 3D printed parts took into account the flow conditioning needs, and the physical limitations of the 3D printers, which were printing volume and printability. The following sub-sections discuss the key features of the flow conditioning components. A full drawing package of all of the 3D-printed parts that the flow conditioning assembly was comprised of is provided in Appendix A.

Vaned Diffuser

The diffuser acted to smoothly allow for the flow to move from a circular geometry (Figure 3.4c) to a rectangular geometry (Figure 3.4a). As the air flowed from one geometric shape to another, there was a significant increase in the flow area. The area increase led to concerns towards the possibility of flow separation occurring. Such a scenario would lead to flow stall and the eventual formation of eddies, which would ultimately affect the downstream jet flow. To address the problem of flow separation, the diffuser was designed² in accordance to design recommendations outlined by Blevins (1984), and is shown in Figure 3.4.

It is important to note that some of the design recommendations were deviated from due to the limitations imposed by the manufacturing process. In particular, a design limitation on the diffuser angle was imposed by the 3D printer's maximum build height (205 mm) in combination with the necessary dimensions of the rectangular geometry. This resulted in a large change in area over a relatively short distance, which would likely cause flow

²Initial design of the diffuser was completed by Jason Michaud and modified for this experiment

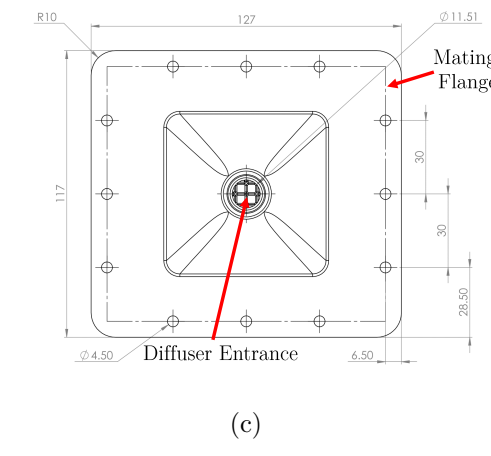
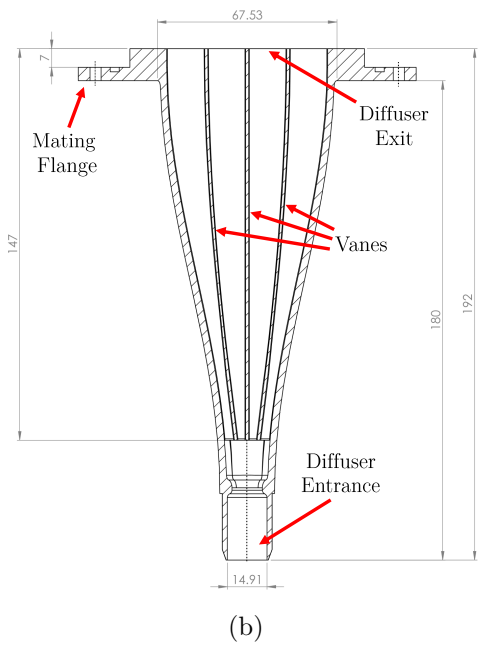
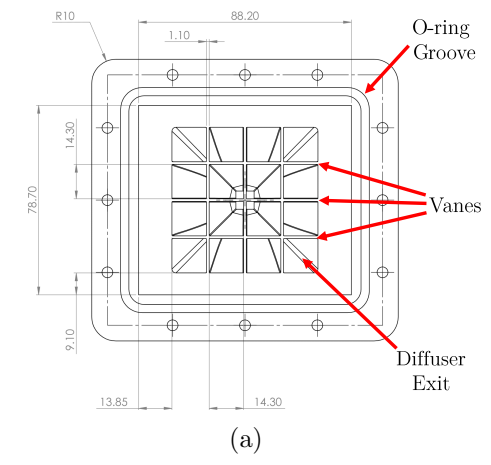


Figure 3.4: Drawings of the vaned diffuser showing its (a) exit face, (b) cross-sectional profile, and (c) entrance face

stall to occur. To mitigate this issue, the walls of the diffuser were made with a contoured profile in order to allow for a smoother transition from the diffuser’s inlet to its outlet. Additionally, three rows and three columns of vanes were added to the diffuser design, as shown in Figure 3.4a). The number of vane rows and columns was necessary to ensure that the angle between the vanes fell within Blevins’s recommended angle range of 7° to 10° . The trailing edges of the vanes could not be chamfered in accordance to the recommended 5:1 ratio as such a ratio resulted in vane thicknesses below the 3D printer’s resolution (0.4 mm, as per printer nozzle diameter). However, it was assumed that the issue was mitigated through the use of thin 0.8 mm vanes, which would allow for the flow to smoothly unify after exiting the diffuser sections.

Screen Holder Assemblies

Screens were used to help redistribute the velocity fields while also reducing the streamwise turbulence of the flow. To be able to use screens within the flow conditioning assembly, it was necessary to place the screens perpendicular to the flow and keep them taut during experimentation. As such, two screen holder assemblies were designed and manufactured. As shown in Figure 3.5, each screen holder comprised of three 3D-printed parts: a short insert with an external coupling, a long insert with both an internal and external coupling, and an internal insert piece.

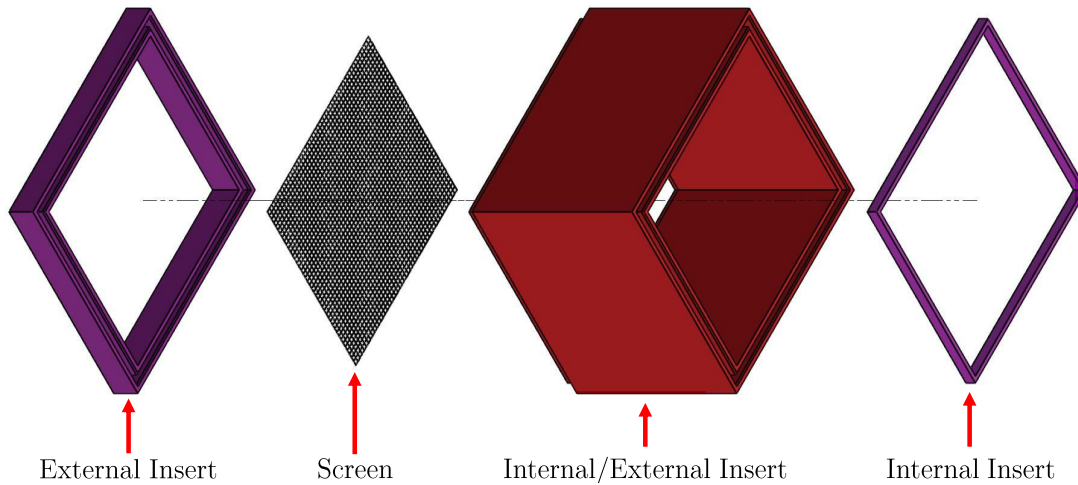


Figure 3.5: Exploded view of the screen holder assembly

Figure 3.6 shows a drawing of the screen, which provides details of the screen used in the screen assemblies. The assembly design allowed for two screens to be held in place, if necessary. Each of the two screen assemblies held one screen in place during experimen-

tation, as this was found to be sufficient in reducing the flow's turbulence. Both screens had the same specifications, which were: an opening size of 2.04 mm^2 , a wire diameter of 0.15 mm , and an open area percentage of 53% . One screen assembly was placed before the honeycomb, and the other screen assembly after the honeycomb.

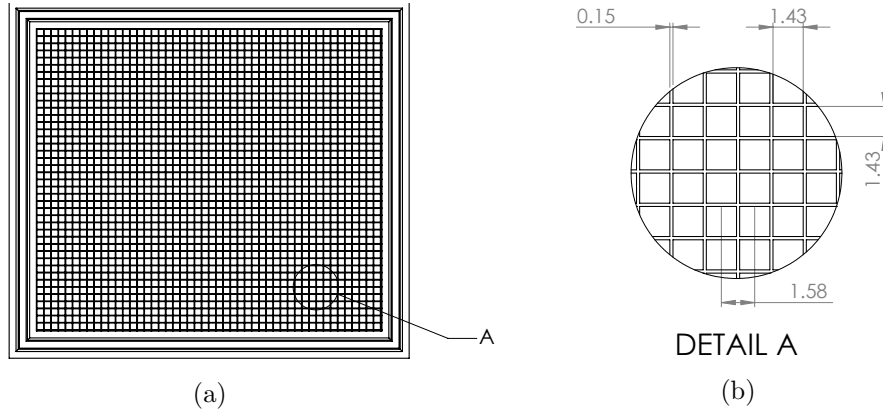


Figure 3.6: Drawing of the screen showing (a) the front face and (b) a detailed view of the screen threads

Honeycomb

The honeycomb stood to reduce both swirl and lateral turbulence in the bulk flow. A drawing showing the front face of the honeycomb grid is provided in Figure 3.7a, and details of the honeycomb's hexagons are shown in Figure 3.7b. Based on the hexagonal dimensions, the honeycomb had a cell diameter of 5 mm . Though not shown in Figure 3.7, the honeycomb's cell length was 40 mm . As a result, the cell diameter to cell length ratio was 8 , which was in the recommended ratio range of $6-8$ (Bell & Mehta, 1989).

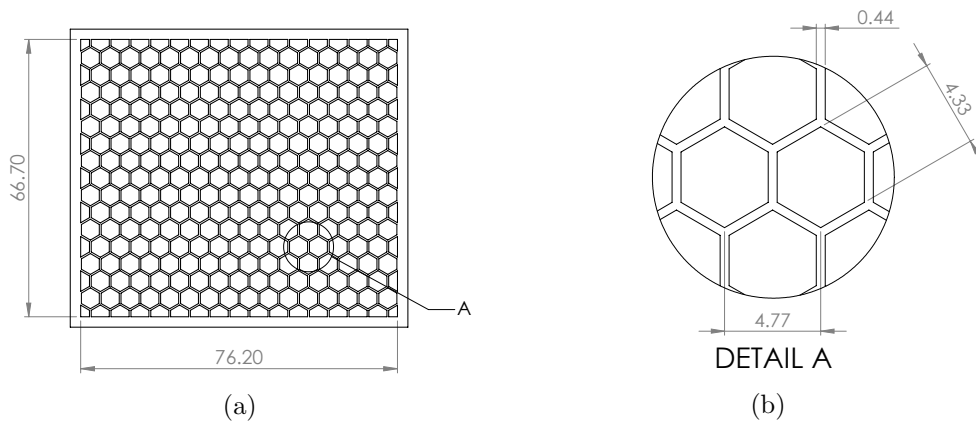


Figure 3.7: Drawing of the honeycomb showing (a) the front face and (b) a detailed view of the honeycomb grid

5th Order Polynomial Contraction

The 5th order polynomial contraction was made to hold the slot assembly, and also allow the flow to converge smoothly to the slot assembly. The polynomial³ that was used was as follows:

$$Y(\eta) = (-10\eta^3 + 15\eta^4 - 6\eta^5) * 1.21 + 1.75 \quad (3.1)$$

where $\eta = X/4.5$. As shown in the cross-sectional drawing of Figure 3.8, two O-rings existed where the slot was held in place in order to ensure that leakage would not be an issue near the slot exit. One O-ring existed underneath the slot, and the other existed along the slot perimeter. A lid was used to hold the slot in place, keep the slot properly aligned, and engage the two O-rings. Four pairs of nuts and bolts were used to fix the lid to the contraction. Clear silicone was applied in the crevasses between the contraction, lid, and test section as an extra means to prevent leakage.

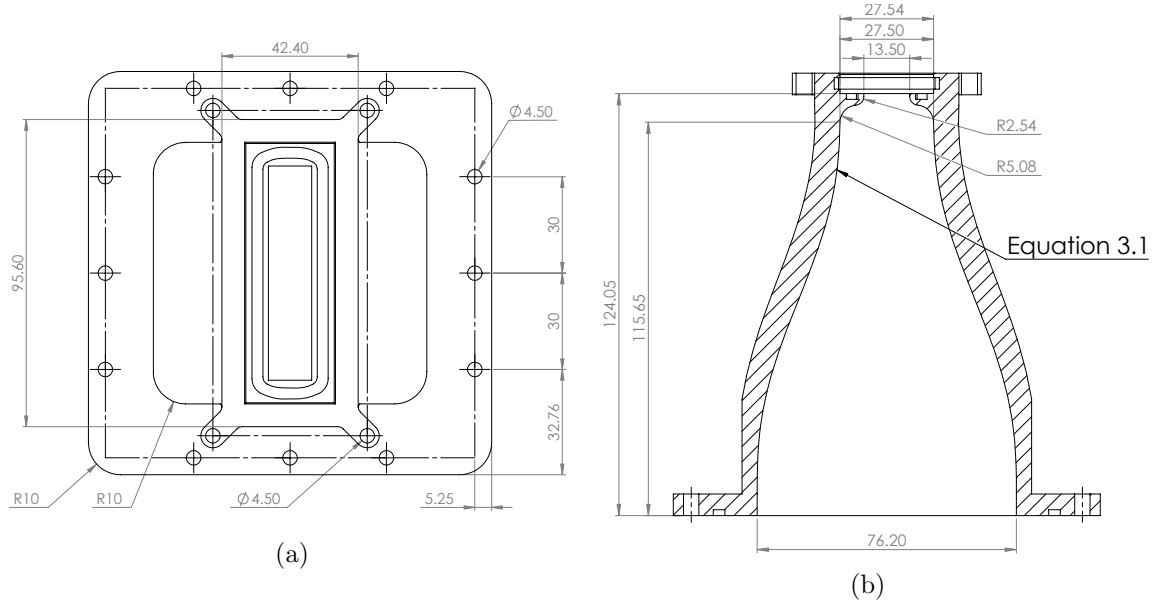


Figure 3.8: Drawings of the vaned diffuser showing its (a) end face and (b) cross-sectional profile

³The polynomial equation was provided by Dr. Mouhammad El Hassan

3.3 Particle Image Velocimetry

Particle Image Velocimetry (PIV) is a non-intrusive optical measurement technique that uses the displacement of tracer particles within a fluid to determine the fluid's velocity field. Over the years, PIV has proven to be popular within the field of experimental fluid dynamics as developments have been made on the technique itself (Adrian, 2005). As shown in Figure 3.9, the constituents of a simple 2D-PIV system are: a flow under investigation, tracer particles, a laser, light optics, and imaging optics consisting of a camera and a lens.

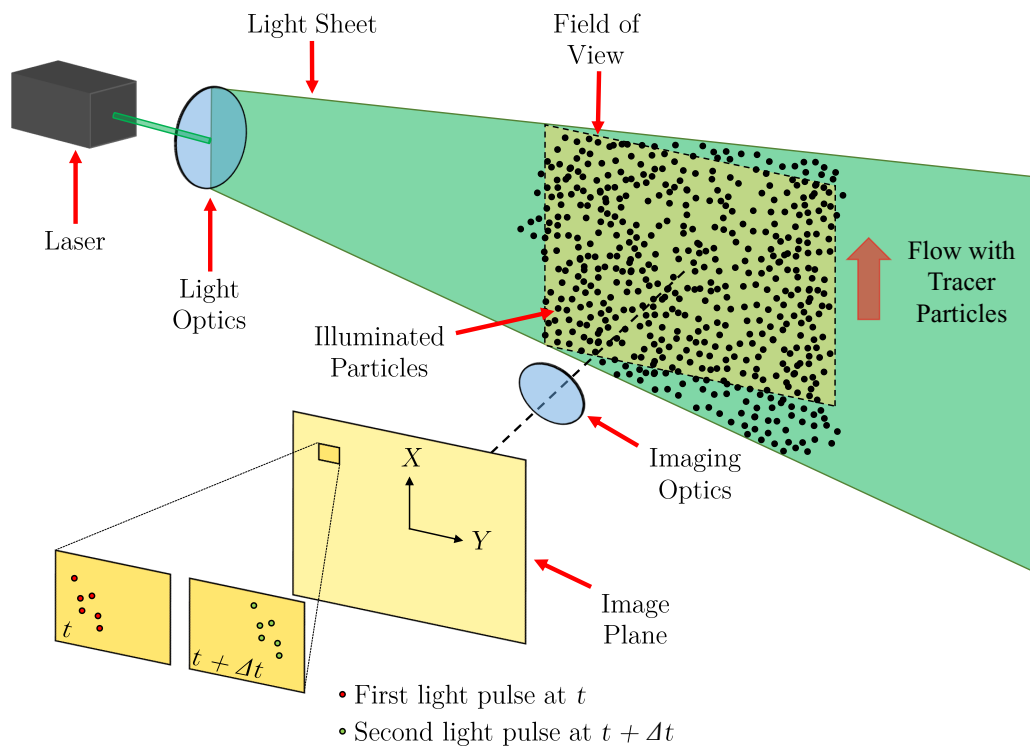


Figure 3.9: Conceptual schematic of a PIV arrangement showing the use of a laser to illuminate the tracer particles, whose movements are captured on two successive images

When performing a PIV-based experiment, a series of steps occur. As the working fluid flows, the laser illuminates the tracer particles twice - one laser pulse occurs at time t and another occurs at a later time $t + \Delta t$. As the laser emits pulses of light, the particles are illuminated and their location is recorded by the camera on either a single frame or a series of frames. After the particles are recorded, their movement is evaluated through statistical methods. In order to do so, a grid of small interrogation windows is prescribed over the entirety of the data. By using a cross-correlation, the displacement of the particles within each interrogation window is determined for successive frames. Once the displacement of particles is determined for the entire grid of interrogation windows, the time interval between

successive frames is used to obtain the velocity vectors of the particles. Each correlation window's resulting velocity vector is then mapped out and, as a result, the entire motion of the flow in the region of interest is made evident. As the preceding overview is only a summary of the pertinent ideas related to PIV, further reference to specific topics may be found in the discussions provided by Raffel et al. (2018).

3.3.1 Stereoscopic PIV

Stereo-PIV is an extension of 2D-PIV where the out-of-plane velocity component is also obtained. In order to obtain the out-of-plane component, a second camera is added to the conventional 2D-PIV system. The two cameras are placed at angles relative to the normal of the object plane. The two most common stereo-PIV configurations are the translational configuration and the rotational configuration (Prasad, 2000). The use of a rotational configuration is desirable as it allows for a larger angle relative to translational systems, which allows for a greater accuracy in the out-of-plane measurements. However, a caveat of the rotational configuration is the difficulty in obtaining well focused recordings on the image plane. To increase the focus quality, the Scheimpflug condition is typically fulfilled by making the object plane, lens plane, and image plane collinear through the use of Scheimpflug adapters, as shown in Figure 3.10.

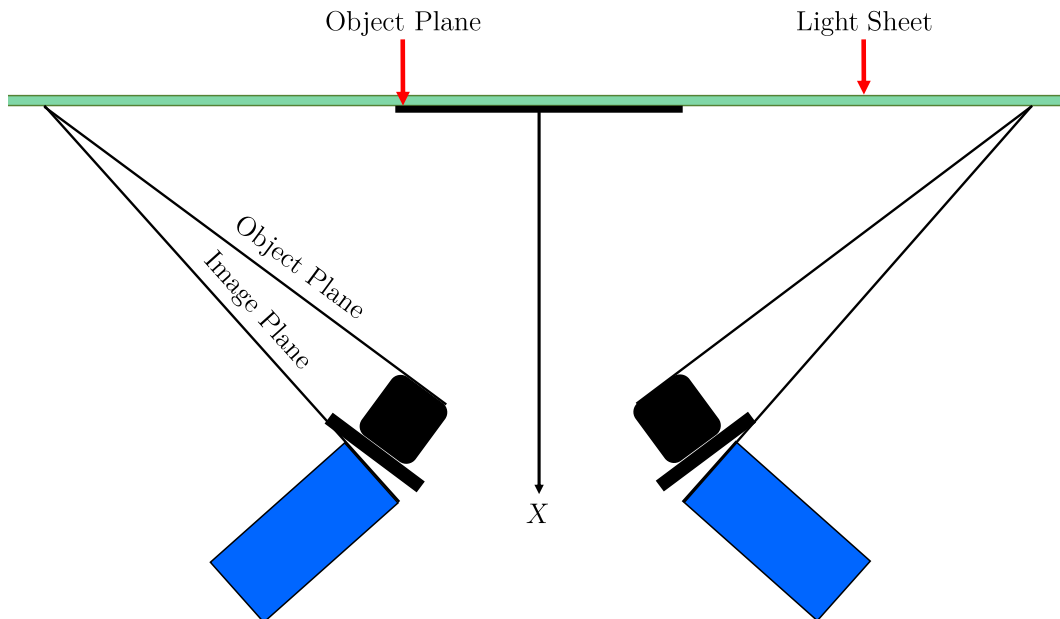


Figure 3.10: Conceptual schematic of a stereo-PIV arrangement showing the fulfillment of the Scheimpflug condition

With the addition of a secondary camera for recording, the calibration of a stereo-PIV system is rendered more complex than that for a 2D-PIV system. This is due to the requirement of accurately calibrating the cameras in all three dimensions. In order to do so, it is common to either translate a planar calibration target in the out-of-plane direction or use a 3D calibration plate equipped with two target levels (Soloff, Adrian, & Liu, 1997). Through either method, both cameras are mapped in three-dimensional space and their overlap is used to form the measurement plane. The latter calibration method has proven to be preferential as the work of Wieneke (2005) has allowed for an easier calibration procedure that results in any misalignment between the laser and 3D-target to be adjusted through disparity vector correction via self-calibration. Using the measurement plane, the images from each camera are warped, and then correlated using 2D-PIV processing methods. After the processing steps are completed for each camera's data, a three-dimensional reconstruction is used to obtain the third velocity component.

3.4 Particle Image Velocimetry Setups

3.4.1 Two-Dimensional Measurements

A two-camera, 2D-PIV system was used to measure the jet in the streamwise direction. Figure 3.11 provides a simplified solid model of the 2D-PIV setup. The overall field of view (FOV) of the two cameras encompassed the region immediately downstream of the slot. Both cameras were Imager ProX-4M cameras (LaVision, GmbH), which is a CCD-sensor camera with a 2048 x 2048-pixel sensor; the sensor's pixel size is $7.4 \times 7.4 \mu\text{m}^2$, and it has a 14-bit resolution. To illuminate the tracer particles within the region of interest, an Nd:YAG laser (Gemini PIV 15, New Wave Research Inc.) was used to provide 532 nm light at 120 mJ. The laser operated at a repetition rate of 15 Hz and had a pulse width of 3-5 ns. The laser beam was formed into a sheet by using commercial light sheet optics (LaVision, GmbH) that allowed for manipulation of the sheet's width and thickness.

As shown in Figure 3.12, the overall FOV designated in Figure 3.11 comprised of two overlapping FOVs. The region immediately proceeding the slot exit (designated as FOV 1) is the near-exit region. In this region, the jet's width was still comparable in size to the slot's width. Thus, a high spatial resolution was needed to resolve the velocity field of the jet when taking measurements in the near-exit region. In order to accomplish this, the camera observing the near-exit region was coupled with a high magnification lens system (Navitar, Inc.) capable of achieving a maximum $12\times$ zoom. A $2\times$ zoom lens was attached

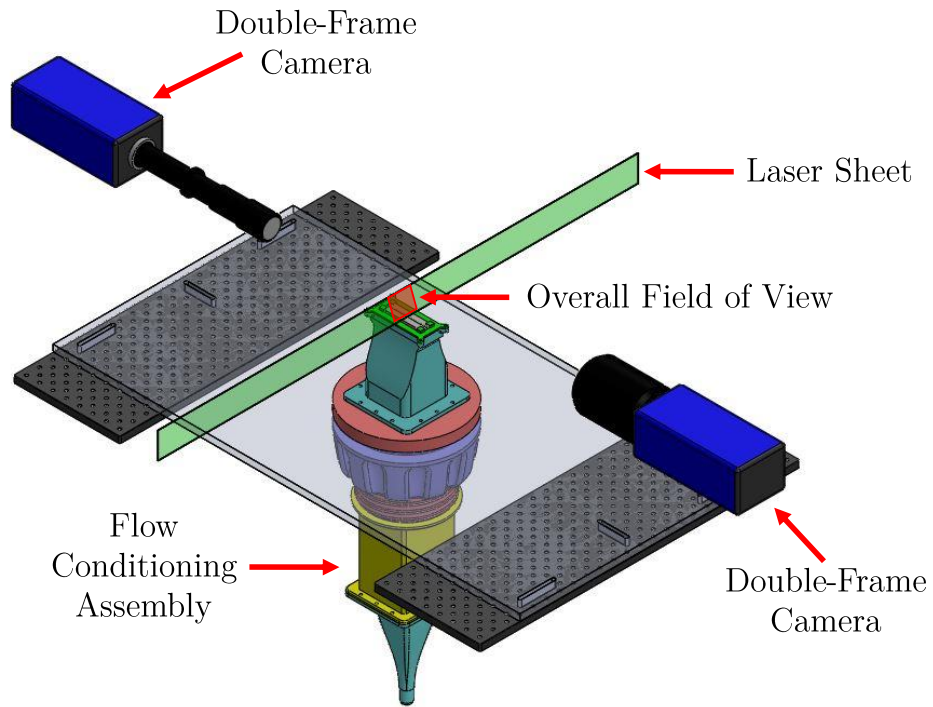


Figure 3.11: A simplified three-dimensional model of the 2D-PIV setup relative to the flow conditioning assembly

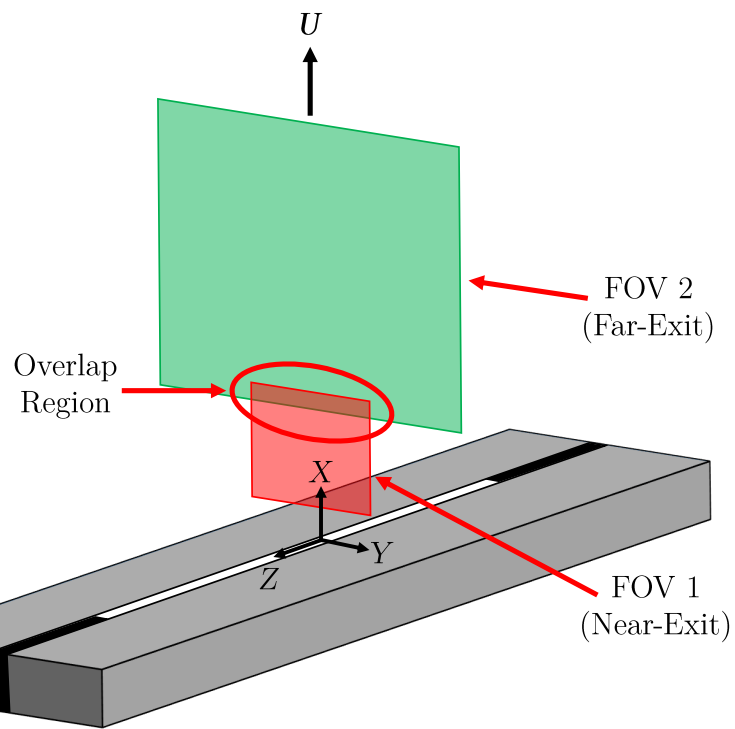


Figure 3.12: A schematic showcasing the two FOVs used for the 2D-PIV setup

to increase the overall zoom capabilities of the lens system. The working distance and zoom were both adjusted until the desired FOV for the near-exit region was obtained. The size of the near-exit region's FOV was 9.25×9.25 mm, which resulted in a spatial resolution of $4.5 \mu\text{m}/\text{pix}$.

The region further downstream in the overall FOV (designated as FOV 2) is the far-exit region. Close to end of the near-exit region, the jet rapidly expanded in size. Thus, a larger FOV was needed to properly observe the jet further downstream. As such, the camera used to observe the far-exit region was equipped with a 105 mm Nikkor lens at an aperture setting of $f/4$. The size of FOV 2 was $21.75 \text{ mm} \times 21.75 \text{ mm}$, which resulted in a spatial resolution of $10.6 \mu\text{m}/\text{pix}$. The near-exit and far-exit region had approximately 4.5 mm overlap with each other. Thus, the overall 2D-PIV FOV extended approximately 27.5 mm ($X = 7.03D_e$) downstream from the slot exit.

As conveyed in Figure 3.12, the 2D-PIV setup was only used to take measurements in the X - Y plane. The ability to accurately take measurements on the X - Z plane could not be ensured as the small widths of the slots proved laser alignment with the X - Z plane to be an arduous task. The two cameras recorded the jet flow at different instances. This was due to a higher Δt being needed downstream to compensate for the jet's decreased velocity. Though proper seeding was obtained in the ambient and jet, the formation of the shear layers in the near-exit led to a lower seeding density surrounding the jet. Thus, a higher amount of images were taken in the near-exit region than in the far-exit region. In the near-exit and far-exit regions, 5000 and 1000 double-frame images were collected at a frequency of 5.5 Hz for every experimental configuration, respectively.

3.4.2 Stereo-PIV Measurements

The stereo-PIV setup took cross-sectional measurements of the jet downstream of the slot exit. The stereo-PIV system utilized the same cameras and laser as the 2D-PIV system. However, for this setup, each camera was fitted with a 105 mm Nikkor lens and a Scheimpflug mount (Lavision, GmbH). Figure 3.13 a simplified solid model of the stereo-PIV setup up relative to the flow conditioning assembly.

The stereo-PIV setup was used to observe three downstream cross-section of the jet. As shown in Figure 3.14, the cross-sectional measurement planes were located at $X = 3D_e$ (11.73 mm), $X = 5D_e$ (19.54 mm), and $X = 7D_e$ (27.36 mm). These three planes were chosen as they were located within the overall FOV of the 2D-PIV experiments. The chosen planes were perceived to be near enough the slot exit such that the effects of AR would

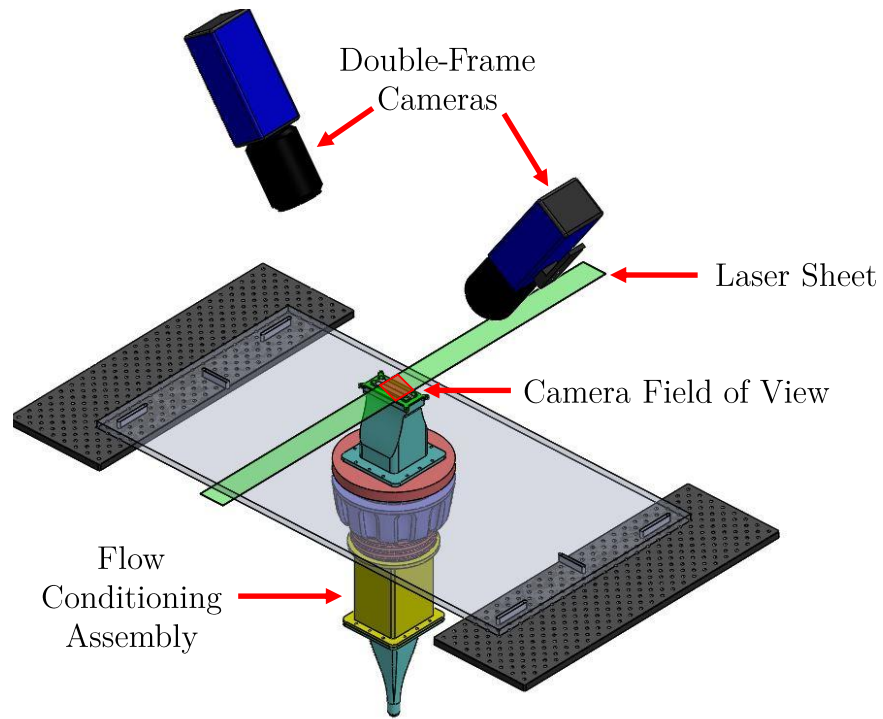


Figure 3.13: A simplified three-dimensional model of the stereo-PIV setup relative to the flow conditioning assembly

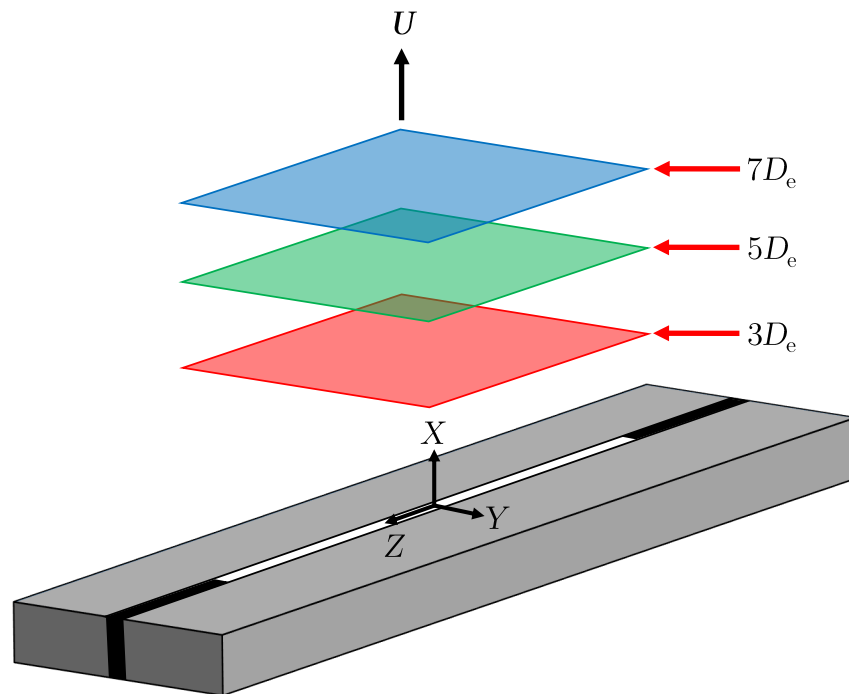


Figure 3.14: A schematic showcasing the three FOVs used for the stereo-PIV setup

be best observed. Planes closer to the coupon exit could not be observed; this was partly due to laser reflections being an issue near the exit. The thickness of the calibration target also inhibited the ability to examine planes closer to the coupon's exit. At each plane, the FOV of the stereo-PIV setup was $70 \text{ mm} \times 70 \text{ mm}$, which resulted in a spatial resolution of $35.2 \mu\text{m}/\text{pix}$. Both lenses were set to an aperture setting of $f/5.6$.

For this study's stereo-PIV experiments, a custom, two-level calibration plate was used to calibrate the cameras, as shown in Figure 3.15. The calibration plate has dimensions of $80 \times 80 \times 9.95 \text{ mm}$, and a plane-to-plane distance of 1.33 mm . The outer plane contains a 10×10 dot configuration while the inner plane contains a 9×9 dot configuration; the dots are 1.60 mm in diameter and have an 8 mm spacing between each other. To have the target in the proper location downstream of the coupon exit, 3D-printed stands were made for the target to rest upon. Once the target was in place, the cameras were both adjusted to obtain the proper FOV. An average deviation value of less than 1 was procured for each camera. Afterwards, self-calibration was then used on 50 images to lower the average disparity deviation value further.

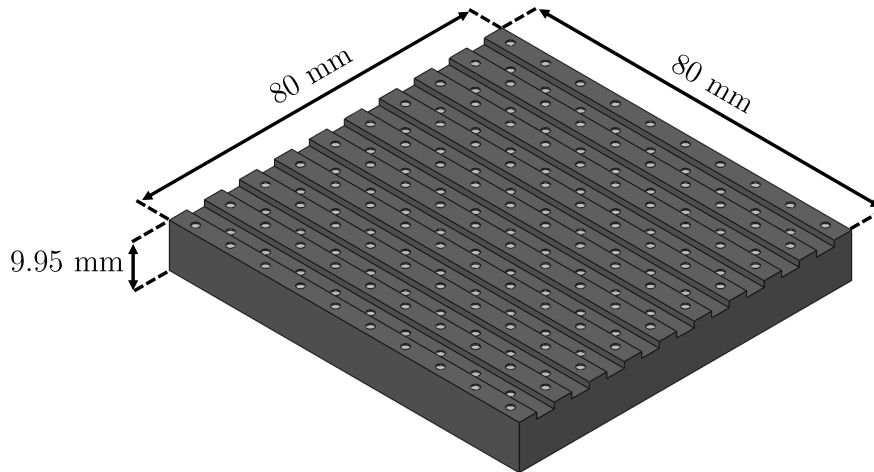


Figure 3.15: A solid model of the custom-made three-dimensional calibration target

After calibration, each of the two cameras recorded 1000 double-frame images at a frequency of 5.5 Hz for every experimental configuration. As the flow rate increased, the Δt between double-frame images needed to be decreased accordingly. At planes downstream, the Δt had to be increased for the same flow rates due to the jet velocity decreasing. Thus, the Δt varied with a range of $8\text{-}20 \mu\text{s}$.

3.5 Flow Line Arrangement

The final aspect of the experimental setup was the flow arrangement. It was necessary to not only supply the system with air to flow across the slot, but also supply the air itself with tracer particles. Of these two necessities, the latter involved much more consideration. The following subsections discuss such considerations. Also discussed is the choosing of experimental flow rates.

3.5.1 Tracer Particles

As briefly described in Section 3.3, PIV-based experiments track the motion of tracer particles to determine the whole-field motion of the working fluid in the region of interest. As such, it is important that proper tracer particles are chosen for PIV applications. In order to do so, consideration must be placed towards the particle's degree of adherence to the fluid's flow and how well the tracer particle scatters light. Further consideration must also be placed on what type of tracer particle is suitable for the working fluid being used, how it will be generated, and also how the tracer particles will be seeded into the fluid (Raffel et al., 2018). The suitability of the tracer particle will mostly depend on its fluid mechanical and light scattering properties, as previously mentioned. However, consideration also needs to be placed on the diameter of the seed with respect to the scale being dealt with. For a smaller field of view, or smaller scale experiment, a small tracer particle diameter will be needed. Based upon the discussion provided by Raffel et al. (2018), tracer particles of di(2-ethylhexyl) sebacate (DEHS) were used for this study, which had a mean particle diameter of approximately $1 \mu\text{m}$.

Fluid Mechanical Properties

To quantify how well a tracer particle follows the flow of a fluid, the lag in velocity of the tracer particles to that of the fluid is determined by observing the influence of gravitational forces on the particles and fluid. The lag that occurs is due to a difference in the densities of the working fluid and the tracer particle, which are ρ and ρ_p , respectively. The lag velocity of a particle, ν_s , is described as

$$\nu_s = d_p^2 \frac{\rho_p - \rho}{18\mu} g \quad (3.2)$$

where d_p is the particle diameter, μ is the dynamic viscosity of the fluid, and g is the

gravitational constant. If the value of ν_s is small enough, then the tracer particle is assumed to follow the fluid flow. Based on Equation 3.2, the relaxation time for a DEHS particle is 2.790×10^{-6} m/s. In comparison to the experimental velocity ranges given later within Table 3.4, the settling velocity is significantly smaller. Thus, it was concluded that the particles were suitable for use, as they would accurately follow the flow. The property values used to calculate this value are provided within Table 3.2.

Table 3.2: Properties used in the calculation of ν_s via Equation 3.2

Particle diameter [m]	1×10^{-6}
Particle density [kg/m ³]	912
Fluid density (air at 20°C) [kg/m ³]	1.204
Fluid dynamic viscosity (air at 20°C) [N·s/m ²]	18.13×10^{-6}

Tracer Particle Generation

Many aerosol generators can be bought rather than manufactured, which mitigates the task of generating particles. In particular, Laskin-nozzle aerosol generators can be used to generate oil tracer particles, as is commonly done for air flows (Melling, 1997). For this study, a commercial aerosol generator (PIVpart45, PIVTEC) was used to generate the DEHS tracer particles. According to the manufacturer, the mean particle size of the DEHS particles is approximately $1 \mu\text{m}$. Four adjustable valves allowed for the number of engaged Laskin nozzles to be adjusted at intervals of three (i.e., 3, 6, ... 45), which in turn controlled the particle density. With the tracer particle type chosen and the generation of the tracer particles taken care of, it was then necessary to determine how to seed the flow passing through the flow conditioning assembly. For open air systems, it is recommended that the particles are introduced far enough upstream so that the particle distribution within the flow may be homogeneous by the time the region of interest in the flow is approached (Melling, 1997). With the aerosol generator bought, two flow line arrangements were considered so that proper mixing of the tracers was obtained, as discussed in the following subsections.

3.5.2 Initial Flow Arrangement Design

Initially, the flow arrangement shown in Figure 3.16 was setup for experimentation. The configuration utilized one air supply line, which continuously supplied compressed air to the system at a pressure of approximately 621 kPa (90 psi). The air passed through a filter (SS-8TF-140, Swagelok) and a mass flow controller (RK-32907-79, Cole-Parmer), which was used to control the air's flow rate. After exiting the mass flow controller, the air passed

through two lines: the bulk flow line and the bypass line. The bypass line was fitted with a needle valve to finely control the amount of air bypassed to the aerosol generator. The two lines joined together at a tee-fitting before moving on to the flow conditioning assembly. This arrangement was initially desired as it allowed for the mass flow rate of the air at the slot exit to be directly controlled by the mass flow controller.

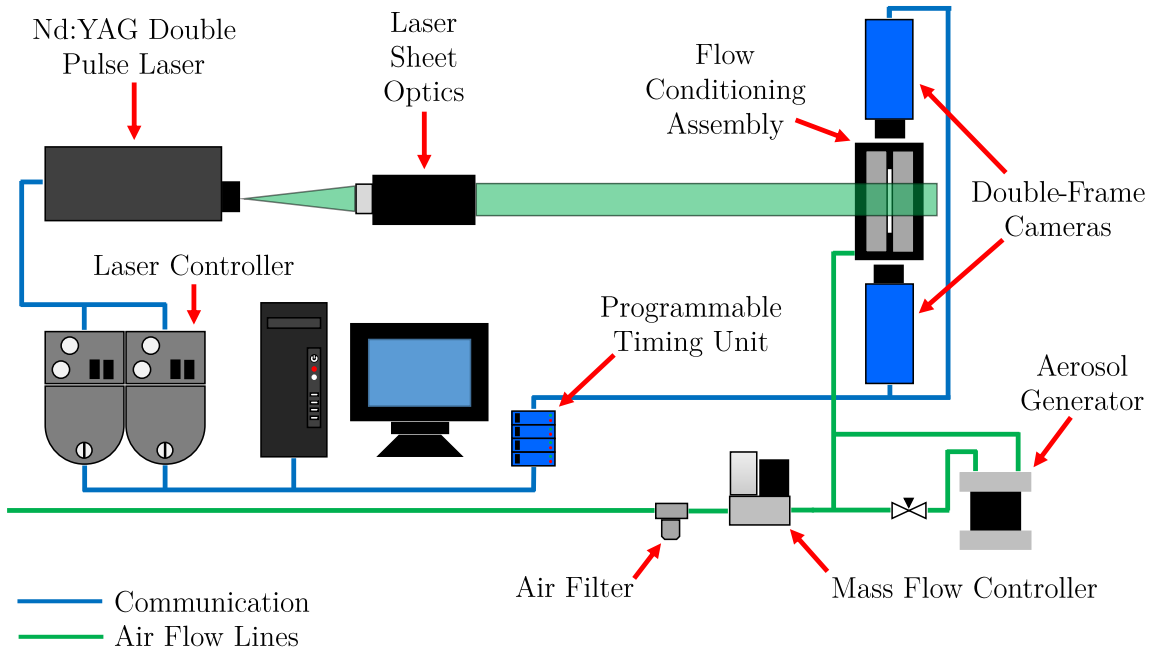


Figure 3.16: A schematic of the initial experimental flow arrangement setup

Many issues arose in the flow line arrangement's due to pressure imbalances in the lines. As the set point for the flow meter was increased, a pressure imbalance formed between the bulk flow line and the bypass line. This increase in pressure imbalance led to some air in the bulk flow line to enter the aerosol generator, which caused the aerosol generator's safety valve to open from over-pressurization. Also, the pressure regulator on the aerosol generator could not be used in such a configuration, thus the issue of over-pressurization could not be mitigated by such means. Such a configuration led to the needle valve having to be used to directly control the tracer particle density in the flow. To achieve lower particle densities in the flow, more air needed to be passed through the bulk flow line. However, a minimum amount of air was needed to engage the aerosol generator. These led to situations where a lower particle density could not be achieved. Another issue with the system was the inability to seed the ambient fluid. The ability to do so was desired as this would allow for the interactions between the jet and ambient fluid to be observed. Thus, with all of these issues existing in the original configuration design, a different air flow configuration

was developed for experimental use.

3.5.3 Final Flow Arrangement Design and Development

Figure 3.17 provides a schematic of the air flow arrangement that was ultimately used for this study. Rather than using one air supply line, this configuration instead utilized two air supply lines instead. One air supply line served as the main flow line, and air would pass through an air filter, a mass flow controller, a 3D-printed venturi vacuum pump, and then enter the flow conditioning assembly. The venturi vacuum pump was added to the system so that the issue of pressure imbalance would not occur as the flow rate was increased. As the flow rate of the main flow line increased, there was always a vacuum to promote the movement of the tracer particles away from the aerosol generator to the main flow line.

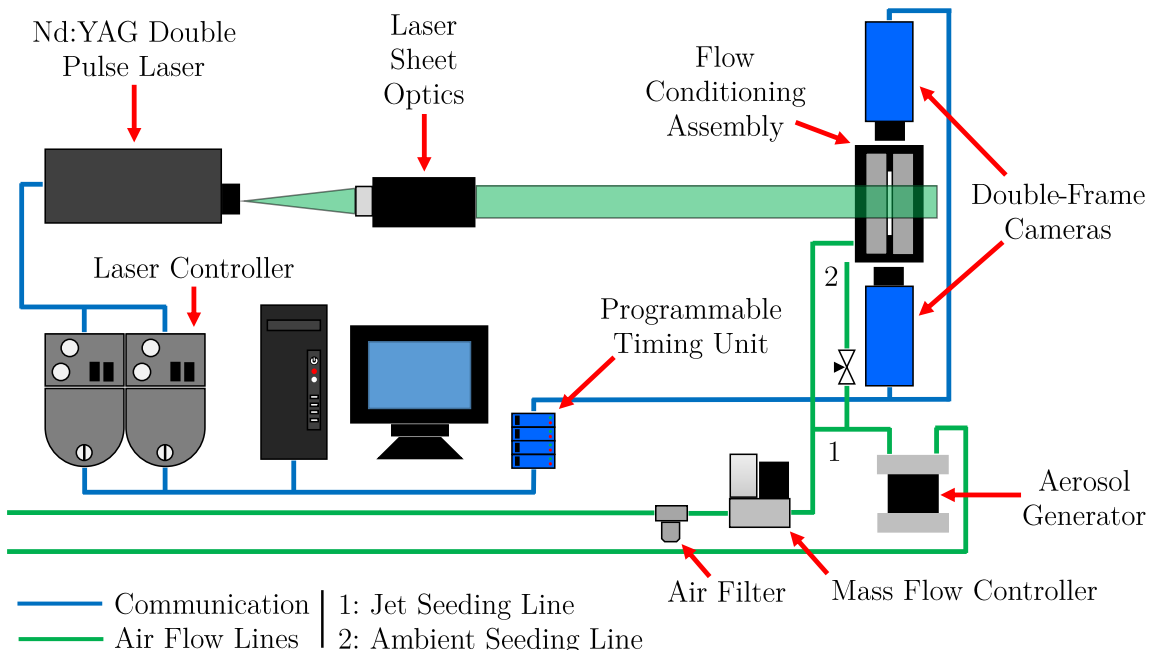


Figure 3.17: A schematic of the final experimental flow arrangement setup

The other compressed air line was used to supply air to the aerosol generator. Upon exiting the aerosol generator, the tracer particles were allowed to pass through either the jet seeding line or the ambient seeding line. The jet seeding line met the main flow line at the 3D-printed venturi vacuum pump and served to seed the bulk flow. On the other hand, the ambient seeding line allowed for excess tracer particles to seed the ambient fluid. As was done in the work completed by Mi et al. (2007), the ambient seeding line consisted of a hose with evenly spaced holes, which was coiled beneath the flow conditioning assembly. This allowed for an even distribution of low-momentum tracer particles to the ambient, which

did not generate a co-flow with the jet flow. A needle valve was added to the beginning of the ambient seeding line in order to control the amount of tracer particles passing to the ambient. Once the entire experimental system was assembled, curtains of black cloth were used to surround all of the system components. The cloths minimized the influence of room draughts on the jet flow and also served as a safety measure to contain the laser. Furthermore, a custom-made fume hood was placed approximately 1.5 m downstream of the slot exit to extract the tracer particles. The presence of the fume hood and its influence on the jet was observed to be minimal, as no changes in the flow were observed when the fume hood was turned on in comparison to when it was turned off.

Experimental Flow Rates

With the flow arrangement assembled, further attention was then placed towards determining appropriate experimental flow rates. Industry-based flow rates could not be procured and used as a reference. However, based on the CFD work of Li, Ma, Mahmoudi, Fattahpour, and Lange (2018), it was estimated that the average velocity of steam exiting a single slot on a slotted liner is approximately 12 m/s. With a slot area of 12 mm², this corresponds to a flow rate of approximately 9 L/min. An attempt was made to use flow rates near this value, but it was observed that reliable seeding densities could not be achieved for such a set point. It was also observed that for $AR = 100$, the minimum achievable flow rate was approximately 25 L/min. Conversely, it was observed that for $AR = 210$, the maximum flow rate was approximately 85 L/min. Using these two observations as guidance, it was decided that the experimental flow rates would range from $Q = 30$ L/min to $Q = 80$ L/min, at increments of 10 L/min, as shown in Table 3.3.

Table 3.3: Flow rates, average exit velocities, and corresponding Reynolds numbers considered in this study

Q [L/min]	U [m/s]	Re_{D_e}
30.00	41.67	10814
40.00	55.55	14419
50.00	69.44	18024
60.00	83.33	21629
70.00	97.22	25234
80.00	111.11	28839

System Limitation and Flow Rate Setting Methodology

Although many of the issues that existed in the first flow configuration were mitigated by the changes in the setup, it is important to note that one key issue continued to exist with the new configuration: the flow rate at the slot exit could not be set exactly. By using two supply lines, more mass was exiting the slot than what was being controlled by the mass flow controller for a given set point. Also, as the flow rate set point increased, the amount of engaged Laskin-nozzle on the aerosol generator needed to be increased in order to maintain the desired seeding density. Thus, for each experimental case, the actual flow rate was observed to change from the desired flow rate.

In an attempt to mitigate this issue, specific steps were taken when conducting experiments. First, the set point on the mass flow controller was always lower than desired total flow rate, by a value of approximately 5 L/min. Second, the experiments were always conducted in sequential order in terms of increasing flow rate. This ensured that the flow rate always increased as the experiments continued onwards, and that at the very minimum, an overall trend of increasing flow rate could be ensured. Third, the 2D-PIV experiments were conducted first, as the desired set-point could be more easily procured, and later used as reference for the stereo-PIV experiments. Based upon theory, the centerline velocity value remains approximately at unity with the exit velocity until the shear layers fully penetrate the jet. Prior to collecting an entire 2D-PIV data set, 50 to 100 images were first collected, processed, and averaged. Once the average velocity field was obtained, the approximate centerline values in the PCR were checked and compared to the average velocities listed within Table 3.3. At most a 10% deviation was allowed between the experimental velocity value and theoretical velocity value, and in every case but one ($Q = 30$ L/min, $AR = 210$), the procured deviation was smaller than 4%, as exemplified by the relative error values provided in Table 3.4.

Table 3.4: Flow rates and relative error for each aspect ratio configuration

Q [L/min]	Q_{100} [%]	Q_{150} [%]	Q_{200} [%]	Q_{210} [%]
30.00	1.29	0.49	0.15	7.72
40.00	2.72	0.44	1.11	1.06
50.00	3.59	1.17	1.67	2.12
60.00	-0.25	1.02	0.39	1.46
70.00	0.37	3.42	1.07	3.70
80.00	2.61	3.02	1.28	2.25

The ability to ensure that the flow rate was near the desired value was much more difficult when running the stereo-PIV experiments. As before, the set point on the mass flow controller was always lower than desired total flow rate, and the experiments proceeded in sequential order. However, the average velocity in the PCR region could not be checked. In order to mitigate this issue, the 2D-PIV data was fully processed, the average velocity fields obtained, and the profiles at $X = 3D_e$, $5D_e$, and $7D_e$ used as reference. As was done with the 2D-PIV experiments, 50 to 100 stereo-PIV images were first collected, processed, and averaged before a full experimental data set was collected. The 2D-PIV values and stereo-PIV values were then compared to each other, and minor adjustments were then made if major differences between the two data sets were found.

Chapter 4

Data Processing Methodology

This study's experiments utilized 2D- and stereo-PIV setups to observe the changes of rectangular jet configurations. For each experiment, the flow of the jet was captured in image sets, and then processed using commercial PIV software (DaVis Imaging Software, 8.4.0, LaVision GmbH). The processed data results were then further processed using in-house codes (Matlab, Mathworks, Inc.) so that the flow quantities of interest could be obtained and examined. This chapter outlines the techniques and considerations that were used to process the data from each PIV setup, and also provides descriptions of the processing steps that were performed to obtain the results presented in the preceding chapters.

4.1 Overview of Image Processing

When calculating a flow field's vector field, it is common to use some algorithm to process the recorded images. Briefly, Raffel et al. (2018) explains that the processing of the recorded images is done through the statistical evaluation of the intensity peaks of the tracer particles. Processing involves using correlation techniques to try to match the intensity signals of particles in order to determine their movement, and, in turn, their velocities. Pre-processing steps are typically applied to help enhance the signals being examined so that a stronger correlation between frames is achieved. After processing the data, further post-processing methods are used in order to enhance the quality of the final results.

4.2 Processing of 2D-PIV data

4.2.1 Image Processing Scheme for Near-Exit Region

Figure 4.1 shows a raw image of the jet exiting the slot in the near-field, and the image after pre-processing has been applied. Pre-processing of the near-exit region data from the long-range microscope was performed by subtracting a sliding background and then applying

particle intensity normalization. The scale length of the subtracted sliding background was 10 px. A scale length of 5 px was used for the particle intensity normalisation. Before moving on to process the images, a rectangular geometric mask was prescribed around the jet and some of the ambient fluid. The mask was prescribed to ensure extraneous ambient fluid was not evaluated and to reduce computational time. As shown in Figure 4.1b, the bottom of the rectangle was located beneath the position of the slot exit to ensure the resolution of PIV data at the exit itself. The length of the rectangle was extended to the end of the image, and its width was adjusted with respect to the spread of the jet to ensure the entirety of the jet was enclosed.

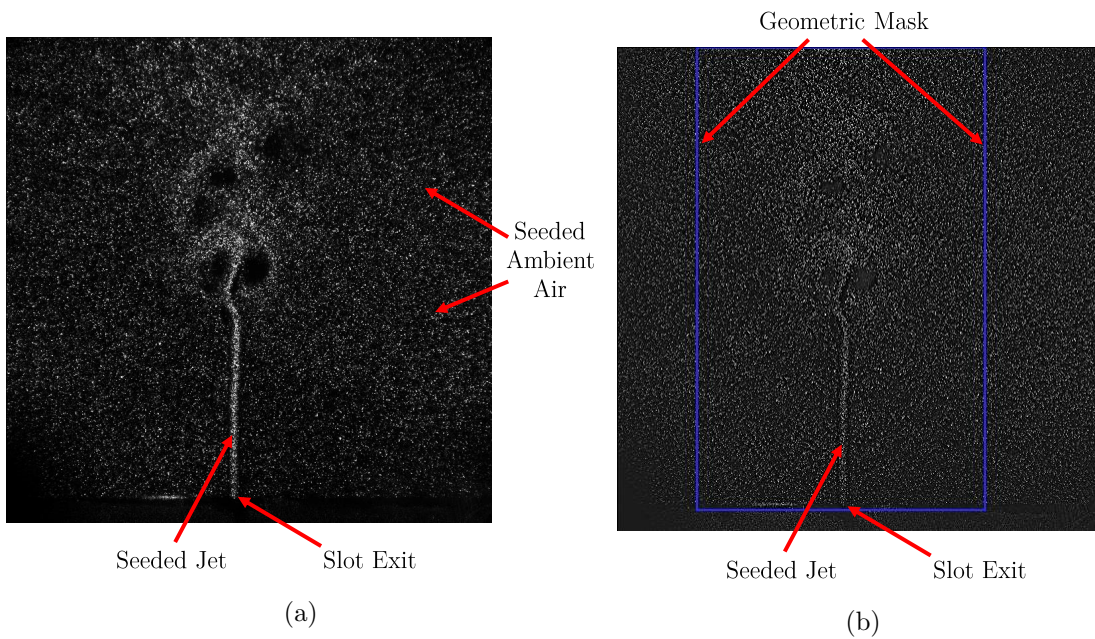


Figure 4.1: Sample images of the near-exit jet flow showing examples of (a) the raw image, and (b) the pre-processed image with the geometric mask shown

The overall scheme for processing was determined after various tests were completed using different settings that changed the size of the interrogation windows, the amount of interrogation window overlap, the number of passes, the shape of the interrogation windows, and the direction of the interrogation windows, if relevant. Ultimately, a multi-pass cross-correlation scheme with decreasing interrogation window sizes was utilized for the processing of the near-exit region data. First, a 64×64 px interrogation window with 75% overlap was prescribed twice to the images so that a coarse reference vector field could be obtained. Following this, a 32×32 px interrogation window with 87% overlap was prescribed twice to the reference vector field. For all of the passes, a square interrogation window was

prescribed, as such a shape was found to resolve the data best and also allow for the fastest correlation of data. Additionally, a high accuracy option was enabled for the final passes, which utilized a B-spline-6 reconstruction methodology was used.

Finally, post-processing steps included the use of a median filter, removal of vector groups, data interpolation, and data smoothing. For the median filter, universal outlier detection was used, where a vector candidate was compared to its 5×5 neighbouring data, removed if its residual was above a 1.3 threshold, and either inserted or re-inserted if its residual was below a 2.5 threshold. Following the use of the median filter, vector groups were removed if they contained less than 5 vectors. Next, empty spaces were filled with interpolated vectors if at least two neighbouring vectors existed. Prior to vector interpolation, the vector field was checked to ensure sure that enough proper vectors for interpolation had been calculated and left over by the preceding processing steps. After applying the pre-processing, processing, and post-processing steps to the individual images within the image set, the corresponding average mean and RMS vector fields for the image set were then obtained.

4.2.2 Image Processing Scheme for Far-Exit Region

The pre-processing steps for the far-exit region utilized a sliding background subtraction and particle intensity normalization. For the sliding minimum subtraction, a scale length of 10 was used. Afterwards, a scale length of 4 px was applied for particle intensity normalization. Unlike the near-exit data, no masks were applied to the far-exit region; the spread of the jet resulted in a majority of the field of view in the far-exit region being filled with the jet itself.

When processing the far-exit image sets, a multi-pass correlation scheme with a decreasing interrogation window size was used. To start, a 64×64 px interrogation window with 50% overlap was prescribed twice to the images so that an initial, coarse vector field could be obtained. Next, a 24×24 px interrogation window with 50% overlap was prescribed twice. Square interrogation windows were chosen for all of the passes to allow for the stereo-PIV data to be properly resolved within the lowest computational time. In addition to the outlined steps, a high accuracy option was enabled for the final passes; for this part, a B-spline-6 reconstruction methodology was used.

Finally, the obtained vector fields were prescribed with a median filter during the post-processing step. The median filter was smoothed with a $1 \times 3 \times 3$ linear filter, which was used to remove any false vectors found within the velocity field. After, the average mean

and RMS vector fields for the image set were then obtained.

4.2.3 Stitching of the Two Vector Fields

With the average vector fields obtained for the near-exit and far-exit regions, further post-processing steps were necessary to apply in order to obtain the results that are presented later in this study. Particularly, the near-exit and far-exit velocity fields needed to be stitched together to conduct a proper analysis on the jet. Thus, the following discussion gives an overview of the steps taken to stitch the two vector fields together. The average vector fields were exported and stored as structure arrays, which contained data fields (X , Y , U , and V arrays) and information fields about the vector field file itself (name, computer location, file history, etc.). The data fields corresponding to spatial locations in X and Y , were $1 \times m$ and $1 \times n$ row vectors, respectively. The data fields corresponding to velocity values U and V , were each matrices of size $(n \times m)$.

Determination of the Maximum Streamwise Velocity

As a first step, U_{\max} was found statistically, as shown in Figure 4.2. The first quarter of the data corresponded to the jet data furthest upstream, and was chosen as this is where the maximum velocity value should be physically located. The ten maximum velocity values were found in each j -th column within the first quarter of the data. Next, the five maximum velocity values were found in each i -th row (Figure 4.2a). All of the maximum velocity values obtained were stored in one array, and the outliers were then removed. Any value above three standard deviations of the mean value was deemed to be an outlier. With the outliers removed, the maximum velocity value of the remaining values was taken to be the maximum velocity value of the entire field (Figure 4.2b). With the maximum velocity value determined, the next step of the stitching process involved determining the location of the centerline.

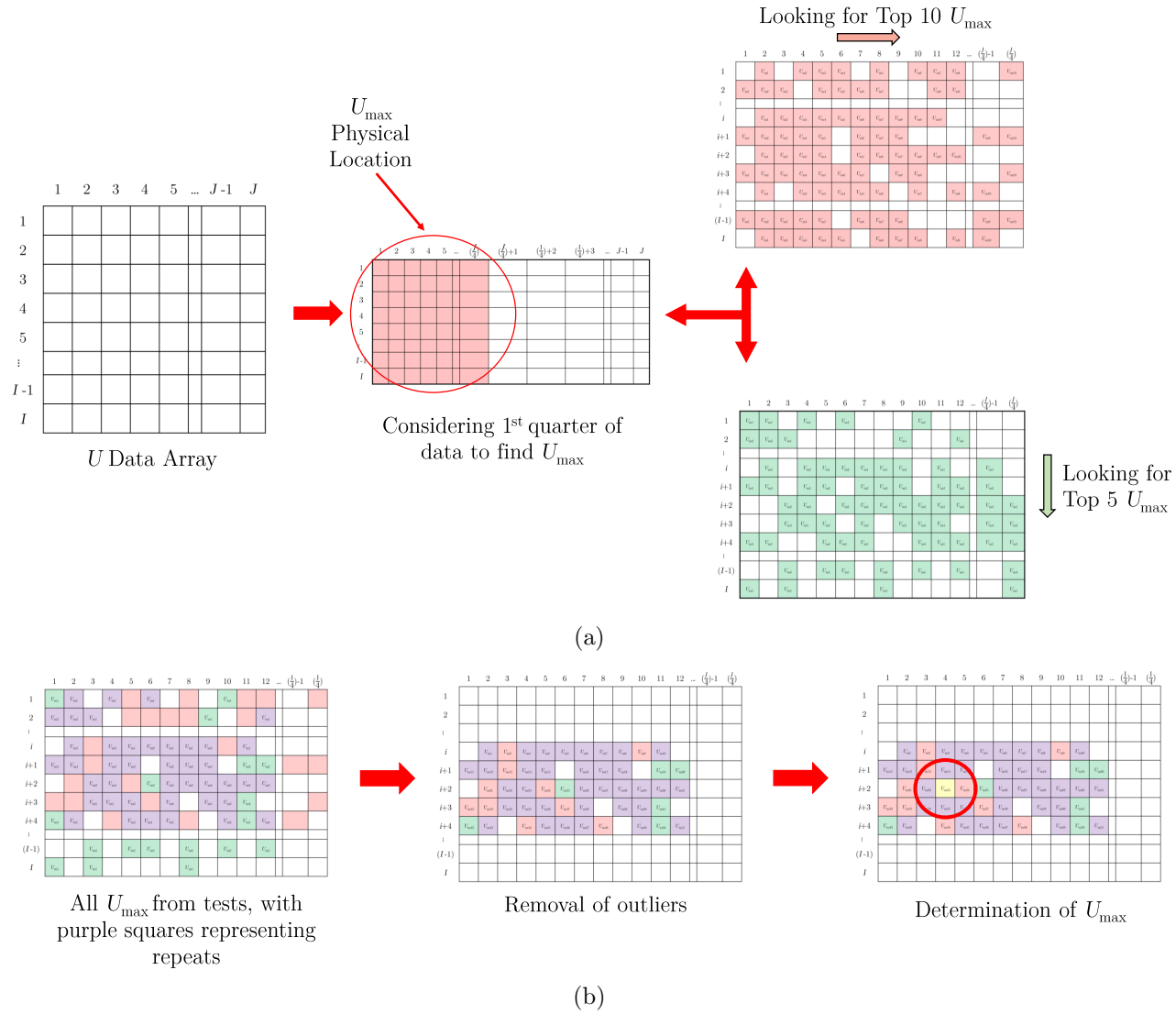


Figure 4.2: Simplified schematic showing (a) determination of U_{\max} values in the first quarter of U data, and (b) the determination of the flow field's overall U_{\max}

Determination and Alignment of the Centerline

Before stitching the two data sets together, it was important to locate each data set's centerline and evaluate the straightness of each centerline. To locate the centerline, the maximum velocity value's i -th location at each j -column was determined, as shown in a simplified schematic in Figure 4.3. Once every maximum velocity location value was found, outliers above three standard deviations were removed from the results. Next, a line of best fit based on the centerline locations was made. The last X -location within the FOV was used as input in the equation of the line of best fit. The resulting Y -value was compared to the Y -location of the previously located maximum velocity in order to evaluate the centerline's straightness throughout the FOV. If no discrepancy was found between the two Y -values, then the centerline location was determined by taking the mode of the centerline locations; the Y data was then centered about the centerline location. However, if a discrepancy was found between the two Y -locations, then the angle formed between the two points was calculated. The vector field was then rotated in the opposite direction about the location of the maximum velocity to straighten the centerline. The maximum velocity value's i -location at each j -column was determined again, and the mode of the centerline locations was taken as the centerline location. After, the Y data was then centered accordingly.

Determination of the Overlap Region

With both velocity fields centered and straight with respect to their centerlines, the overlap region between the two velocity fields needed to be found. Prior to doing so, the downstream velocity field data was mirrored as the far-exit camera was located on the opposite side of the laser sheet as the near-exit camera. It was also necessary that the far-exit data be interpolated so that its spatial resolution matched that of the near-exit data. Once these two steps were accomplished, the start of the overlap region was found by considering streamwise velocity data. The overlap region start was located where the maximum velocity in the far-exit field appeared in the near-exit field's data. The end of the overlap region was determined by finding where the centerline velocities in the near-exit field stopped appearing in the far-exit field. With the overlap region between the two data sets found, the next task to accomplish involved the determination of the optimal location within the overlap region to crop and stitch the velocity fields.

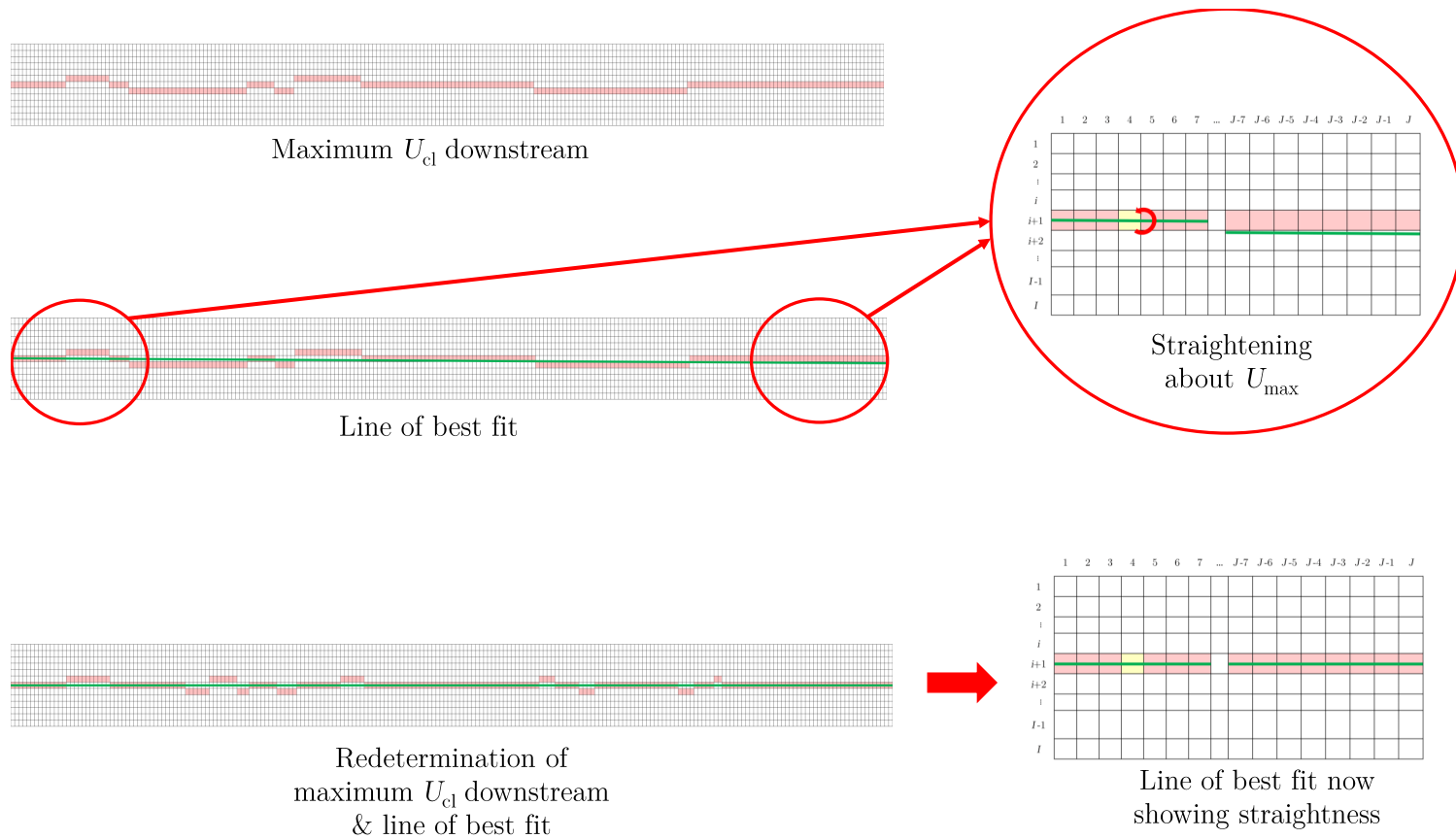


Figure 4.3: A simplified schematic showing the determination of the downstream U_{cl} , and the straightening of the centerline's line of best fit about U_{max}

Considerations to Determining the Optimal Cut-off Location

The optimal cut-off point was found by consider U_{cl} , $Y_{\frac{1}{2}}$, and the centerline u' . First, each data set's U_{cl} in the overlap region was considered. Each U_{cl} from the near-exit region was compared to each U_{cl} in the far-exit region. At each j -th position within the near-exit data set, the absolute differences between the velocity at that point and all the points within the far-exit data set were calculated. Both data sets were row vectors of size $(1 \times p)$, thus the resulting matrix was of size $(p \times p)$. Any values in the resulting array equal to or near zero corresponded to the near-perfect or perfect overlap of the centerline velocities from each data set, respectively. The same process was then repeated for the $Y_{\frac{1}{2}}$ and u' values within the overlap region, with results stored in other $p \times p$ sized matrices. Finally, an overall average matrix of size $(p \times p)$ was obtained by taking the average of all three matrices. The overall minimum value within this matrix was searched for. The column location corresponded to the cut-off point within the near-exit region, and the row corresponded to the cut-off point within the far-exit region.

Final Steps of the Stitching Methodology

The final steps of the stitching process involved cropping the vector fields and putting them together in a new structure array. Both fields were cropped at their respective cut-off location, and the sides of each were cropped so that the jet was centered. In the near-exit field, the X -value closest to zero was located, and any values before this point were cropped. For the far-exit field, the last 10 data columns were cropped out downstream to remove spurious vectors. With both data sets cropped, a new structure array for the stitched velocity field was made. The Y data from the far-exit field was copied to the stitched velocity field data as it would remain the same. A new X data column vector was made using the the near-exit data with additional values to accommodate the far-exit's data. The number of points added corresponded to the number of columns in the far-exit region, and they were given the spatial resolution of the upstream data. Next, the near-exit region was padded with rows of zeros so that its total number of rows matched the total number of rows of the far-exit region. Once this was completed, the U and V data sets were put together in the new structure array. With this, the process of stitching the two vector fields was complete. Figure 4.4 shows the average streamwise velocity field of a stitched jet. Each experimental configuration's stitched colormaps and vector fields are provided in Appendix D.

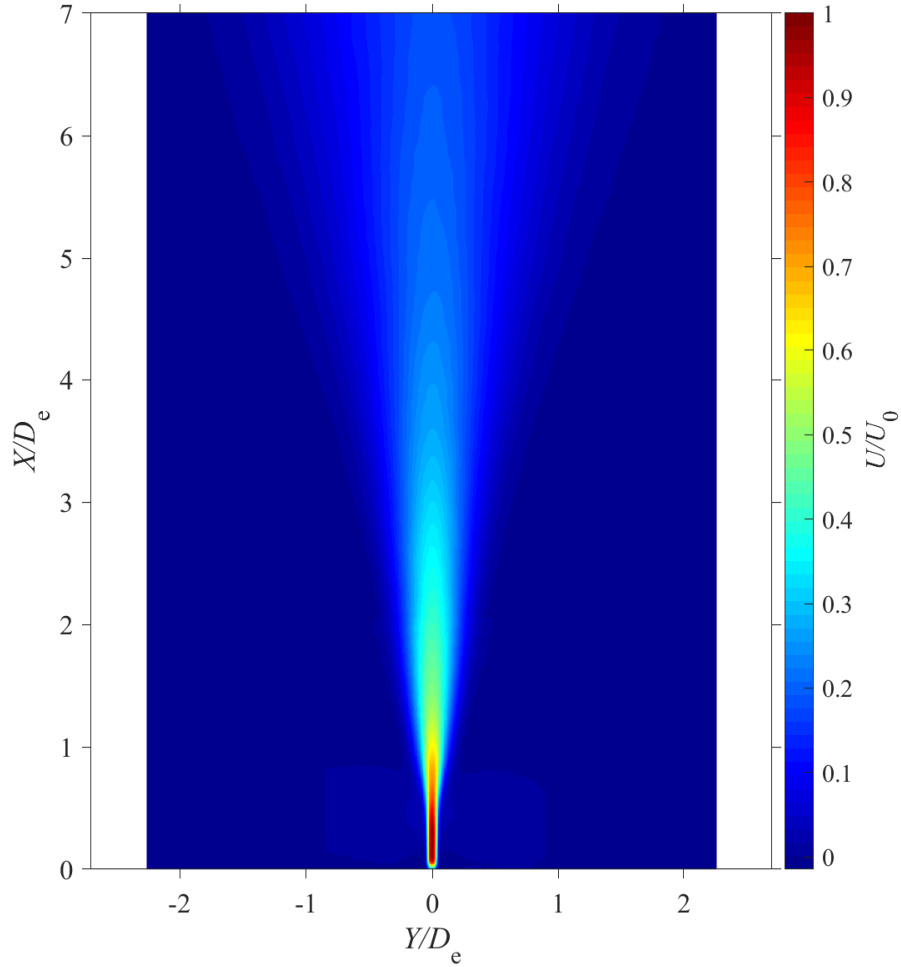


Figure 4.4: A streamwise velocity whole-field map of a jet after the stitching process

4.3 Processing of Stereo-PIV Data

4.3.1 Image Processing Scheme

The stereo-PIV image sets obtained from each of the three downstream planes were processed using the same scheme. To pre-process the images, a sliding background subtraction and particle intensity normalization were applied; scale lengths of 10 and a 5 were used for the background count subtraction and particle intensity normalization, respectively. A geometric mask was prescribed to the images to reduce the processing time, and was placed around the jet and some of the ambient air.

To process the stereo-PIV image sets, a multi-pass correlation scheme with a decreasing interrogation window size was used. Initially, a 96×96 px interrogation window with 75%

overlap was prescribed twice; following this, a 32×32 px interrogation window with 87% overlap was prescribed twice. Square interrogation windows were chosen for all of the passes to allow for the stereo-PIV data to be properly resolved within the lowest computational time. In addition to the outlined steps, a high accuracy option was enabled for the final passes; for this part, a B-spline-6 reconstruction methodology was used.

Finally, the obtained vector fields were prescribed with a median filter during the post-processing step. The median filter was smoothed with a $1 \times 3 \times 3$ linear filter, which was used to remove any false vectors found within the velocity field. When closer to the slot exit, reflections from the experimental setup became a notable issue, and as a result, many data sets taken within this region exhibited spurious trends in the ambient fluid. As a result, an additional 15×15 Gaussian filter was also applied to the stereo-PIV data. After applying the pre-processing, processing, and post-processing steps to the individual images within the image set, the corresponding mean and RMS vector fields for the image set were then determined.

4.3.2 Definition and Determination of the Jet Boundary

With the average vector fields obtained for the stereo-PIV measurements, further post-processing steps were necessary to apply in order to obtain the results that are presented later in this study. A key post-processing step was the determination of the jet boundary. In order to determine the boundary of the jet, it was necessary first form a definition of the jet boundary, and then develop a methodology to apply the definition to the stereo-PIV data.

Jet Boundary Definition Considerations

When developing the definition of the jet boundary, two types of definitions were considered; first, a set value was considered, followed then by a ratio cut-off value. The use of a set velocity value for the boundary cut-off was inspired by the work completed by Meslem, Nastase, and Allard (2010). In their investigation, a set value of 0.2 m/s was used to define the boundary cut-off point. The value was deemed acceptable by the authors as it corresponded to the minimum velocity change detectable by humans - a condition that was pertinent to the application of their results to the HVAC field. Due to the lack of existence of an value intrinsic to the SAGD process, a fixed-value definition could not be made for the jet flow. As such, the use of a set value was quickly dismissed, and the definition of the jet's boundary was instead developed by considering the ratio of the local velocity to that

of the in-plane centerline velocity.

Cutoff ratio values inspired by the work of others were initially considered before a value suitable for this study was determined. The use of a ratio cut-off value was influenced in part by the boundary-layer theory described by Schlichting and Gersten (2017). A cut-off ratio of $0.01 U_{cl}$ was at first considered, as this was analogous to the use of $0.99 U_0$ for laminar boundary layers. Using $0.01 U_{cl}$ proved to be feasible for the first two stereo-PIV planes (i.e., $X = 3D_e$ and $X = 5D_e$). However, the value could not be used for the last stereo-PIV plane (i.e., $X = 7D_e$), as the boundary extended beyond the field of view in many cases. A cut-off ratio of $0.5 U_{cl}$ was then considered as Tsuchiya and Horikoshi (1986) used such a criteria in their analysis of rectangular jets. When this definition was implemented to the stereo-PIV data, it was observed that large velocity gradients existed, which caused the boundary to have a highly irregular shape in comparison to the shape formed by the $0.01 U_{cl}$ cut-off ratio. Thus, this definition was rejected for use as it was observed that its adoption would result in a gross misrepresentation of the true jet boundary. From here, the cut-off velocity ratio was reduced from a value of $0.5 U_{cl}$ to a value of $0.01 U_{cl}$ until a ratio value proved to be acceptable for utilization. Eventually a cut-off ratio of $0.05 U_{cl}$ proved to be feasible for all three planes, and was adopted for use in this study.

Determination of the Jet Centerline

With the a definition of the jet boundary formed, attention was then placed on developing an algorithm to correctly implement the definition to the processed stereo-PIV data. First, it was necessary to locate the centerline of the jet within the observation plane, determine U_{cl} , and center the data about the location of U_{cl} . The entire streamwise velocity data was divided by the maximum in-plane streamwise velocity. Using the maximum streamwise velocity was deemed acceptable as it was observed that this velocity would be higher than the actual U_{cl} value. Due to this, the boundary formed by $0.05 U_{max}$ would be smaller than the boundary formed by $0.05 U_{cl}$, and lead to a smaller data set being used to find the centerline. At each i -position, the velocity values were searched, and it was determined whether at least one value was equal to or greater than $0.05 U_{max}$; if no value was found, then the velocity values at the i -position were removed. With the remaining data, the j -location of the maximum velocity was determined at each i -position. Next, a line of best fit was made using the j -positions, and its straightness was evaluated. If the line was not straight, the angle of rotation necessary to straighten the line was calculated. the stereo-PIV data was then rotated accordingly, and the new j -positions of the maximum velocities were then

determined. The location of the centerline was then determined by taking the mode of the j -positions. The Y -axis ($Z = 0$ mm) was then defined as the midpoint of the centerline. With the centerline and Y -axis locations determined, the Y and Z data were then shifted to center the data accordingly.

Determination of the Jet Boundary Points

With U_{cl} determined and the stereo-PIV data centered, an algorithm was used to move away from the centerline until the Y and Z locations of boundary points was found. In order to accomplish this, two tests were utilized: a horizontal test and vertical test. As shown in Figure 4.5, the horizontal test would first move away from the centerline in the positive Y -direction to determine the two Z -locations where the criteria was fulfilled at a specific Y -location. The same methodology was used afterwards while moving away from the centerline in the negative Y -direction. The vertical tests employed in the same manner, but along the centerline instead. In order to completed these tests, four criteria points were designated: at the minimum Y -axis, maximum Y -axis, minimum Z -axis, and maximum Z -axis points where the jet boundary condition was fulfilled. Although these four points were deemed to likely not be the minimum nor maximum points of the boundary itself, they were taken to be so that the algorithm would have set boundaries for the horizontal test and vertical tests. For each test, the (Y,Z) coordinates of the boundary points were stored in an array.

Once the two test were completed, the (Y,Z) coordinate points in both arrays were analyzed so that erroneous coordinates could be deleted. Two types of coordinate points were looked for by the algorithm: those deemed to be *repeating points* and those deemed to be *false points*. The repeating points corresponded to coordinate points found multiple times by the algorithm, either within one test or between the two tests. False points corresponded to coordinates that were located outside of the boundary. These points were artifacts of the tests themselves and only could appear near the boundary where the jet might have high curvature. Once these two types of coordinate points were deleted, the remaining coordinate points were ordered into a single array. The coordinate points were ordered so that they prescribed a clockwise path that started and ended on the positive Y -axis. With the coordinate points in order, a polygonal shape corresponding to the jet boundary was obtained, as shown in Figure 4.6.

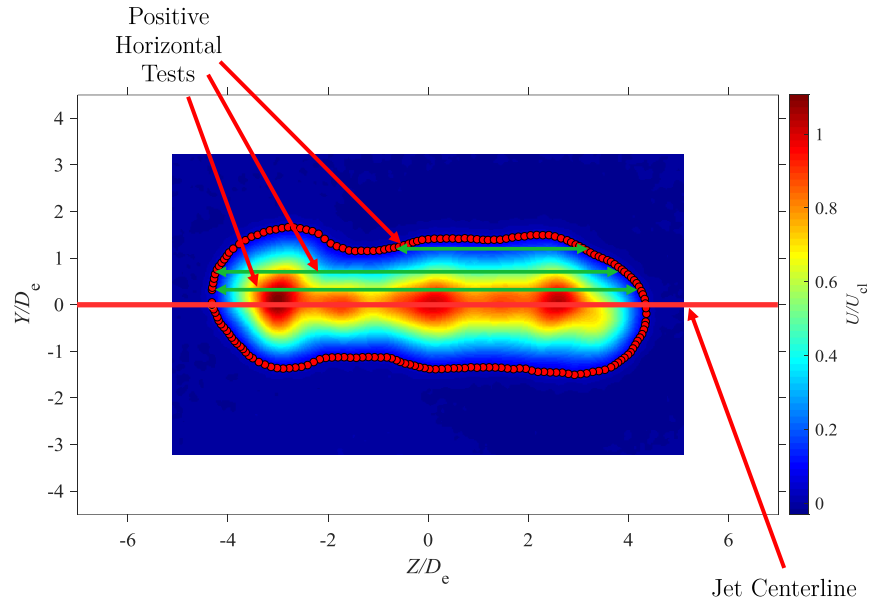


Figure 4.5: Annotated screen capture to exemplify the horizontal test employed to determine the jet boundary

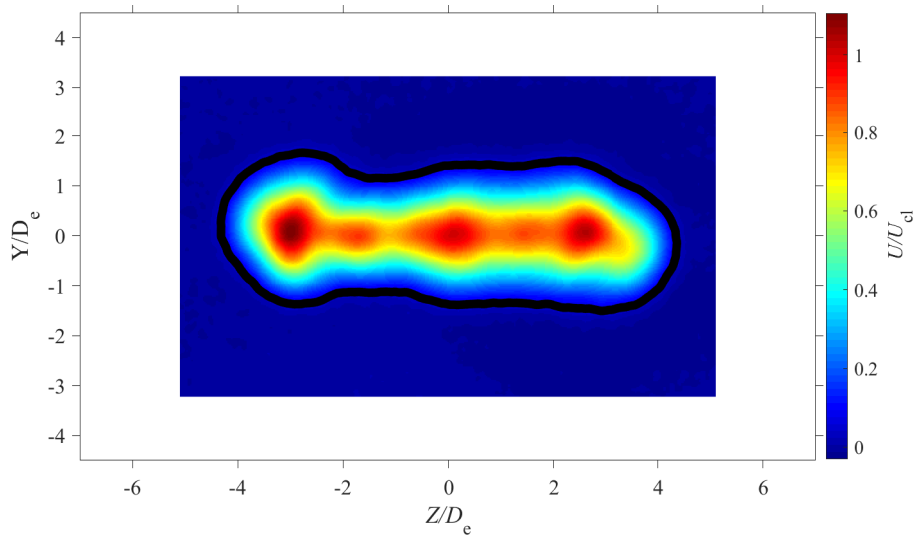


Figure 4.6: A streamwise velocity whole-field map of a jet plotted with its calculated jet boundary

4.4 Uncertainty of PIV Data

With PIV experimentation, many factors lead to uncertainties in the measurements taken. Raffel et al. (2018) discusses the root causes of such uncertainties and how they form. The authors discuss how such uncertainties are inherent throughout the entirety of an experiment, with uncertainties being introduced in the setting-up of the experiment, the recording

process, and the evaluation of data. When considering the experimental set-up, some uncertainties that arise include those due to camera misalignment, incorrect calibration, incorrect laser sheet thickness, machine desynchronization, tracer particle lag, peak-locking, and laser misalignment (Raffel et al., 2018; Sciacchitano, 2019). During the recording process, variations in the fluid, such as density, seeding, and velocity gradients, can cause uncertainties in the recordings themselves (Raffel et al., 2018; Sciacchitano, 2019). With regards to the data process, Sciacchitano (2019) states that the chosen type of correlation analysis, interrogation window size and shape used, and cross-correlation peak fit algorithms that are chosen are all factors that can introduce further uncertainty into the system. In order to help minimize these sources of uncertainty, Raffel et al. (2018) recommends that extra time be spent in optimizing experiments and measurement set-ups.

With respect to the preceding discussion, attempts were made to minimize the sources of uncertainty in the PIV setups, and in the data analysis. In order to estimate the uncertainty of the PIV data gathered for this study, commercial software (DaVis Imaging Software, 8.4.0, LaVision GmbH) was used. Using the software, a maximum relative uncertainty of approximately 1.1% for U was found for the 2D-PIV data. The calculation was performed by the software in accordance to the equation described by Sciacchitano and Wieneke (2016). When considering the stereo-PIV data, maximum relative uncertainties of approximately 0.7%, 0.8%, and 0.5%, were found for U , V , and W , respectively. Each u' , v' , and w' had relative uncertainties of approximately 2.2%. The PIV uncertainty propagation techniques discussed by Sciacchitano and Wieneke (2016) were considered for the calculating the uncertainties in other variables. However, only the relative uncertainty for turbulent kinetic energy could be calculated with the collected data, which was found to be approximately 2.9%.

Chapter 5

Influence of Aspect Ratio

5.1 Introduction

Though the influence of AR on rectangular configurations has been previously investigated, no investigation has been performed within a range that can be truly considered as high in value. As such, this chapter stands to present and discuss the changes in the downstream flow characteristics of a free, rectangular jet for high AR s. The results herein are for aspect ratios of $AR = 100, 150, 200,$ and 210 , and correspond to a Reynolds number of $Re = 18000$ ($Q \approx 50$ LPM). This AR range extends past the range of $AR = 1$ to $AR = 50$ that most experimentalists have worked within. The results are presented in two main sections: streamwise flow field observations, and cross-sectional flow field observations. The streamwise and cross-sectional flow field observations correspond to the data gathered from the 2D- and stereo-PIV experiments, respectively. The 2D-PIV measurements were taken in the streamwise direction, within the X - Y plane, at $Z = 0$. These measurements were done in order to observe the evolution of the jet in the near-exit region of the slot, from $X = 0$ to $X = 7D_e$. From this data, the trends in the jets' velocity decays, turbulence intensities, and velocity half-widths are presented. In turn, The stereo-PIV measurements were taken downstream from the slot exit, within the Y - Z plane at $X = 3D_e, 5D_e,$ and $7D_e$. This data was used to gain insight on the jet's whole-field velocity profiles, Reynolds shear stresses, turbulent kinetic energy, streamwise vorticity, and boundary shape. Following this, a summary of the conclusions based upon the discussions within the two sections are then given to end the chapter.

5.2 Streamwise Flow Field Observations

The follow subsections present and discuss the results of the 2D-PIV measurements. The results from these measurements are presented first as they are inline with the way in which many previous experimentalists, as listed in Table 2.1, have presented their findings on free, rectangular jets. The results are simpler to observe, and as a result, any conflicting observations between this study and previous studies are more easily determined. With a simple understanding of any conflicting observations obtained, the more complex cross-sectional data can than be properly analyzed to give further insights on why such conflicting observations arise.

5.2.1 Centerline Decay

Observations on Equivalent Diameter versus Slot Width

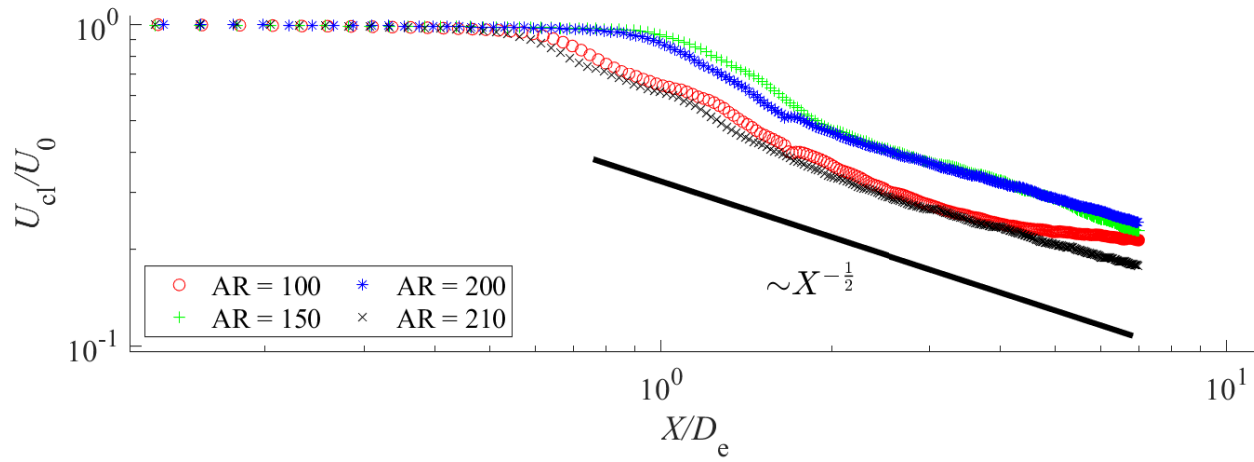
As discussed in Chapter 2, Hussain and Husain (1989) argued that for higher AR s, an elliptical jet is expected to behave more like a planar jet, and in such a scenario, h proves to be more important as a length scale than D_e . Previous studies show rectangular jets also tend to act more like a planar jet as AR increases (Mi et al., 2005; Sforza & Trentacoste, 1967). Figures 5.1a and 5.1b show the mean centerline decay against X , which has been nondimensionalized by D_e and h , respectively. For both plots, the axes have been set to log-log form, as is typically done for such plots (Mi et al., 2005; Sfeir, 1979; Sforza et al., 1966). From Figure 5.1a, it is observed that there is a near-collapsing of $AR = 100$ with $AR = 210$, and $AR = 150$ with $AR = 200$, and the data sets remain collapsed for nearly the entirety of the FOV. In comparison, in Figure 5.1b, it is seen that there is partial collapsing of $AR = 150$ with $AR = 200$, but to a lesser degree. It is also observed that the data for $AR = 200$ is instead located between $AR = 100$ and $AR = 150$.

Based on the trends observed in Figures 5.1a and 5.1b, the suitability of D_e or h as a lengthscale is unclear. The partial collapsing indicates that D_e may have a greater relevance as a length scale in comparison to h , which displayed less collapsing of data. However, Hussain and Husain's work shows a collapsing of data in the far-field at approximately $X = 50D_e$, which constitutes a measurement length that is $7.14\times$ larger than the measurement length used in this study. As such, it can not be fully concluded whether or not the full collapse of data does in fact occur with either variable. In spite of being unable to conclusively determine which lengthscale is best, D_e is used for the remainder of this study, as it allows for all data sets to end near each other at $X = 7D_e$.

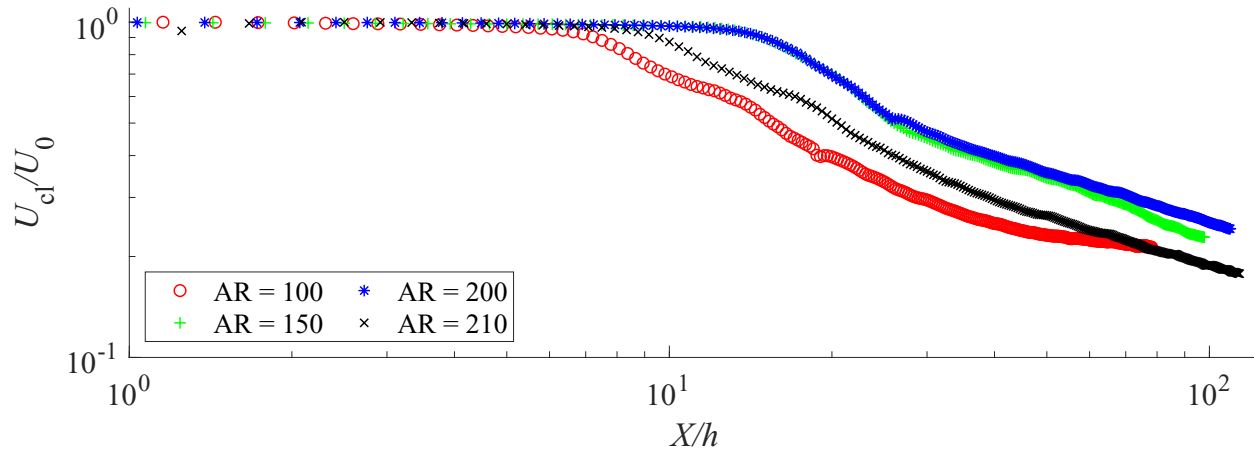
Observations on the Velocity Decay

By focusing on Figure 5.1a, some observations can be made on the decay profiles of the four jets. First, with reference to the annotation line underneath the data sets, it appears that slightly past $X = D_e$, all four configurations display a decay rate similar to that of a planar jet. For $AR = 100$, it appears that near $X = 5D_e$ the decay rate begins to slow down. With respect to the observations of others (Mi et al., 2005; Sfeir, 1979; Sforza & Trentacoste, 1967), a slowing down of the decay rate is typically indicative of a transition from the CDR to the ATDR. However, the veracity in a transition actually occurring cannot be fully substantiated, as this is beyond the FOV of the data. The other three AR s do not appear to exhibit a slowing down in decay rate, which follows the observation by others that an increase in AR results in a lengthening of the CDR and shortening of the ATDR (Krothapalli et al., 1981; Mi et al., 2005; Sfeir, 1979; Sforza & Trentacoste, 1967). It is notable that $AR = 150$ appears to increase its decay rate near the end of the FOV. This may be due to two factors: small measurement inconsistencies, or the planar jet limit has not been achieved. When considering $AR = 200$ and $AR = 210$, this appears to be the case, as both run nearly parallel to each other. However, to fully verify if the planar jet limit has indeed been reached, further investigation is warranted.

With regards to an increase in AR , some deviations from the trends seen for low AR configurations are observed. First, it appears that the PCR region increases (in general) as AR increases. This is in contrast with the observations made by others (Krothapalli et al., 1981; Mi et al., 2005), who established that a shortening of the PCR occurs as AR increases. The works of Sfeir (1979) and Tsuchiya and Horikoshi (1986) established, that differences in the cross-sectional exit profile can account for changes in the downstream flow. Thus, one explanation for the observed discrepancy between this study and the works of Krothapalli et al. (1981) and Mi et al. (2005) is that the cross-sectional exit profiles differ enough that there is a change in how the velocity decays. A secondary explanation is that the use of higher AR s results in significant changes in the flows in the near-exit region, which cannot be explained by considering the velocity decay solely. Second, it appears that some change in the flow occurs between $AR = 200$ and $AR = 210$, as the latter appears to transition differently to the CDR in comparison to $AR = 150$ and $AR = 200$. It is observed that the velocity decay for high AR configurations follows similar decay trends as those seen for lower AR , with some notable deviations, which may be a consequence of a different cross-sectional exit profile, or of using high AR s.



(a)



(b)

Figure 5.1: Velocity decay at $Re = 18000$ with the X-axis nondimensionalized by (a) equivalent diameter and (b) slot width. Every 5th data point has been plotted for clarity

5.2.2 Streamwise Velocity Turbulence Intensity

Figure 5.2 shows the downstream turbulence intensity along the centerline, with every fifth data point displayed for clarity. The sudden decrease in turbulence intensity near the slot exit is a reflection of the difficulty in taking velocity measurements near this location, which results in an underestimation of the centerline velocity. All four data sets exhibit three similar downstream trends: an increase in the turbulence intensity towards a peak value, a decrease in value after the peak is reached, and an apparent tendency towards an asymptotic value afterwards. As there is a decrease in value, some data sets exhibit a disparity in data, which is a result of the stitching methodology described in Chapter 4. Further downstream, most AR s (with the exception of $AR = 100$) slowly increase in magnitude. These overall trends are observed to be inline with the observations originally described by Sfeir (1979), which was also observed by others (Krothapalli et al., 1981; Quinn, 1992b; Tsuchiya & Horikoshi, 1986).

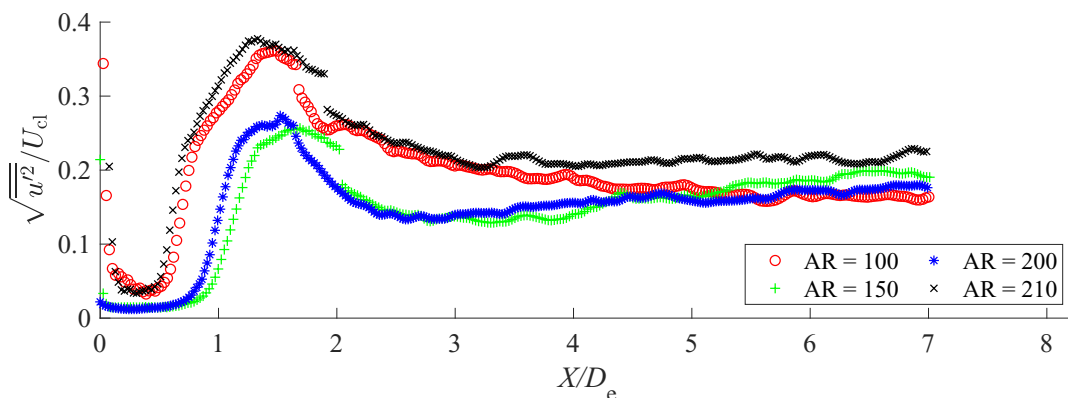


Figure 5.2: Normalized turbulence intensity against X/D_e at $Re = 18000$. Every 5th data point has been plotted for clarity

When comparing all four AR s' to each other, some observations can be made. First, it is clear that there is a pairing of data sets similar to that observed for the velocity decay, with $AR = 100$ pairing with $AR = 210$, and $AR = 150$ pairing with $AR = 200$. Secondly, the observed trends in the peak turbulence intensity appear to diverge from the observed trends seen for low AR configurations. There is a divergence in that the magnitude of the turbulence intensity peaks do not have an apparent linear trend. As AR increases, the peak magnitude first appears to decrease, from $AR = 100$ to $AR = 150$, and then increase afterwards. Furthermore, the maximum peak location does not appear to have a linear trend, as an increase in AR initially increases the downstream location of the peak, until an

apparent limit is reached between $AR = 150$ and $AR = 200$, after which the location appears to decrease as AR increases. This latter trend is in agreement with other observations (Lozanova & Stankov, 1998), but the former is not. The turbulence intensities' asymptotic values cannot be fully discerned as the downstream measurement length has not allowed for the data to enter a region of axisymmetry. However, it is observed that the values approach a value near 0.2, which is slightly lower than the values observed by others (Krothapalli et al., 1981; Quinn, 1992b; Sfeir, 1979; Tsuchiya & Horikoshi, 1986). Like the trends seen for the velocity decay, it is observed that high AR configurations follows similar turbulence intensity trends as those seen for lower AR , but with some deviations, which appear to indicate a change in near-exit flow with respect to AR .

5.2.3 Velocity Half-Width

As 2D-PIV measurements could only be taken on the X - Y plane, a full analysis of the velocity half-widths could not be made within this study. However, Figure 5.3 presents the plotting of the lateral velocity half-widths against the downstream distance, both of which have been nondimensionalized by D_e . Within Figure 5.3, only the positive $Y_{1/2}$ values have been plotted. However, the velocity half-width plots within Appendix C are plotted with both positive and negative $Y_{1/2}$ values visible.

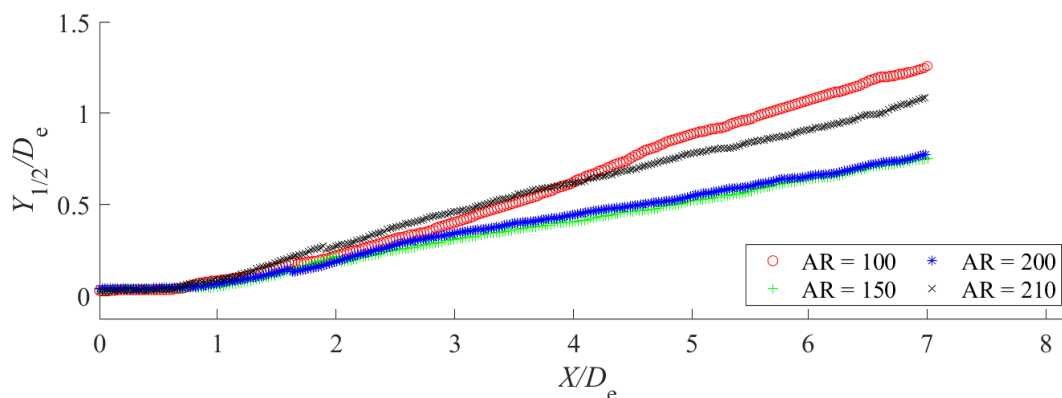


Figure 5.3: Normalized lateral velocity half-width against X/D_e at $Re = 18000$. Every 5th data point has been plotted for clarity

Initially, similar trends are observed near the slot exit in Figure 5.3, but some changes are seen further downstream. All four $Y_{1/2}$ data sets appear to collapse onto each other, with no apparent changes occurring until $X = D_e$ and $X = 2D_e$. Within this range, slight changes between the four data sets become apparent. Part of these deviations are a product of the stitching process. However, between $X = 2D_e$ and $X = 3D_e$ there is an appreciable

change in the $Y_{1/2}$ trends. Unlike the works presented within Table 2.1, it is been observed that as AR increases, the four $Y_{1/2}$ data sets do not exhibit increasing linear variation. Instead, it is observed that as AR increases, there appears to be a decrease in $Y_{1/2}$, until $AR = 150$, which collapses with $AR = 200$. Afterwards, an increase in $Y_{1/2}$ is observed for $AR = 210$, which is between $AR = 100$ and the collapsed pair. Like previous observations, this change in behaviour appears to be indicative of some AR -based limit. As $Y_{1/2}$ relates to the deformation of the jet, this indicates that all four jets may be deforming differently from one another. However, as information on the changes in $Z_{1/2}$ are absent, such conclusions cannot be fully made.

5.3 Cross-sectional Flow Field Observations

This section presents the results of the stereo-PIV experiments, which stand to supplement the 2D-PIV experiments. The following subsections discuss the in-plane nature of the rectangular jet configurations at $X = 3D_e$, $5D_e$, and $7D_e$. All of the whole-field plots presented in this section are formatted in the same way. The whole-field maps are ordered by row with respect to AR , and by column with respect to downstream measurement plane. Every whole-field plot is available in Appendix E. Every plot was made to have the same colormap limits, with the exception of a few, which needed to have different colormap limits. As such, the figures have been given a different colormap to visually highlight its larger colormap limits.

5.3.1 Whole-Field Average Velocity Measurements

Streamwise Velocity Maps

Figure 5.5 shows the streamwise velocity map for every AR configuration at each cross-sectional plane. Each velocity field has been nondimensionalized by U_{cl} . A comparison of all four jet configurations shows that the cross-sectional streamwise velocities have a high AR dependency. At $X = 3D_e$, each of the four jets exhibit profiles that are different from one another. However, when each jet is looked at individually, and their downstream evolution observed, clear trends are observed.

Of all four jets, $AR = 100$ appears to be approaching axisymmetry at $X = 7D_e$, which agrees with the slowing down of its velocity decay plot observed earlier. However, it is clear that both cross-planes still exhibit a notable degree of dissimilarity with each other. The evolution of $AR = 150$ and $AR = 200$ are similar to each other, with $AR = 150$ appearing to take on a greater barbell shape, like that discussed by Vouros et al. (2015), than $AR =$

200. Such similarity between the two, however, explains the collapsing of $AR = 150$ and $AR = 200$ in Figure 5.1a. When observing the profiles for $AR = 210$, it appears that some critical AR has been passed, as its profiles are markedly different than those seen for $AR = 200$. It is observed that much of the velocity has been induced away from the center at $X = 3D_e$. Further outward induction of the velocity appears to be occurring at $X = 5D_e$, where $AR = 210$ appears to almost bifurcate. However, further downstream, it appears that the jet is beginning to tend towards axisymmetry, as the induction of velocity towards the centerline is occurring. The initial induction of the velocity away from centerline helps to explain the observed collapsing of $AR = 100$ and $AR = 210$ in Figure 5.1a, as the decay is based on the centerline velocity values. Overall, the experimental in-plane velocity results give insight to the 2D-PIV results, and the highly complex evolution of each jet.

By considering the centerline velocity profiles in each of the cross-planes, it is clear that high- AR jets behave much differently than their low- AR counterparts. When considering the spanwise direction, it is clear that none of the four configurations exhibit the commonly observed saddle-back velocity profile. At minimum, some of the jets exhibit three peaks in the spanwise direction. It does appear that at $X = 7D_e$, both $AR = 150$ and $AR = 200$ have spanwise velocity profiles that are approaching a saddle-back shape, but this cannot be confirmed. The occurrence of multiple peaks is an indication that the vorticity dynamics are different for high- AR configurations, and suggests the presence of more streamwise vortices than previously observed by other authors. There does not appear to be a linear relation between the number of peaks formed and AR , however. When considering the lateral velocity profiles along the spanwise direction, a considerable amount of variability is observed to exist. This is due to the formation of multiple velocity peaks and troughs in the spanwise direction. This variation in the lateral direction is at odds with the work of Marsters and Fotheringham (1980) who observed little variation in the lateral velocity profile along the spanwise direction. In their study, they noted that the three-dimensional nature of the flow had little influence on the jet, except for near the ends of the jet. $AR = 200$ appears to agree with this statement, but the other configurations do not. Overall, this indicates that the three-dimensional nature of the jet has a higher degree of influence near the origin as high AR s are approached.

Observations on the Jet Asymmetries

By examining the jet configurations presented within Figure 5.5, it is clear that some asymmetries exist in the cross-sectional shape. Some of these asymmetries are observed to

be minor, as is the case in Figure 5.5c. Such minor asymmetries can easily explained by the affects of small room draughts on the system, and are not of specific concern for discussion. Larger asymmetries, however, are of greater concern, as their existence must be thoroughly considered and their consequences understood.

When trying to determine the sources of these large asymmetries, consideration towards this study’s setup must be made. Such disturbances may arise from manufacturing defects, tracer particle agglomeration, room draughts, and small misalignment’s in the setup assembly. Whatever the source of disturbance, it is observed that the overall scale of the slot exit is small enough ($\sim 10^{-1}$ mm) in the lateral direction that the flow field at the exit is substantially affected if such a disturbance occurs.

For this study, large asymmetries are observed for $AR = 100$, as shown in Figures 5.5a, 5.5b, and 5.5c. It is observed that at $X = 3D_e$, the jet is not laterally uniform along the spanwise direction. As a consequence, at $X = 5D_e$, the jet has a large asymmetry, with an apparent bottle-necking of the jet between the leftmost and central peaks. Further downstream, this has resulted in the leftmost peak being larger than its two counterparts, and it also appears to be separating from them. It is clear that if such asymmetries as those seen for $AR = 100$ form, then their influence is felt by the jet far downstream. Overall, this highlights the sensitivity of high- AR jets to small-scale disturbances, and the difficulties in obtaining consistent measurements with high- AR configurations.

Observations on the use of Centerline Velocity for Nondimensionalization

As discussed at the beginning of this section, some of the whole-field figures needed to be given larger colormap limits. This was a direct result of the use of U_{cl} or U_{cl}^2 to nondimensionalize the cross-sectional fields, which is standard practice when dealing with cross-sectional jet data (Marsters and Fotheringham (1980); Quinn (1992b); Tsuchiya and Horikoshi (1986)). The ability to do so, and effectively obtain comparable, dimensionless flow fields has been brought up as a concern in this study, however. This is particularly the case for $AR = 210$ at $X = 5D_e$, as shown in Figure 5.5k. At this location, the jet appears to almost be bifurcated, and a result, U_{cl} is significantly lower in magnitude in comparison to the surround velocities. As a result, the dimensionless flow-field has much larger values in comparison to the other flow-fields presented in Figure 5.5.

The variation of the velocity along the spanwise direction is the cause for such concerns. As previously discussed, the three-dimensional nature of the jet appears to have a higher degree of influence near the origin as high AR s are approached. The effects of this

observation, together with the observation that high- AR jets are sensitive to small-scale disturbances, leads to many of the jets having notable variation in the spanwise direction, as shown for $AR = 210$ at $X = 5D_e$ in Figure 5.4.

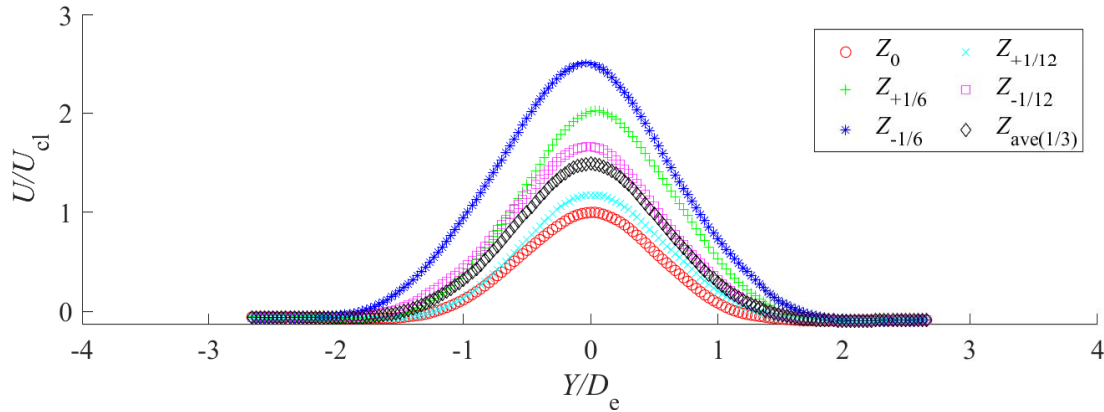


Figure 5.4: A plot showing the variation of the streamwise velocity profile along the spanwise direction for $AR = 210$. All data points plotted

Though Figure 5.4 shows an extreme case, the variation of the streamwise velocity along the spanwise direction is clearly significant for high- AR jets. The streamwise velocities correspond the profiles on the centerline, one sixth the spanwise length away from the centerline in both directions, one twelfth the spanwise length away from the centerline in both directions, and the average of the streamwise velocities within a third spanwise length about the origin. In this case, it is clear that use of U_{cl} does not properly capture the near-centerline variations that are occurring. Though not adopted for this study, it is observed that some averaging along the spanwise direction (e.g., $Z_{ave(1/3)}$ in Figure 5.4) may provide a better value for nondimensionalization purposes, as it captures and balances out such variations.

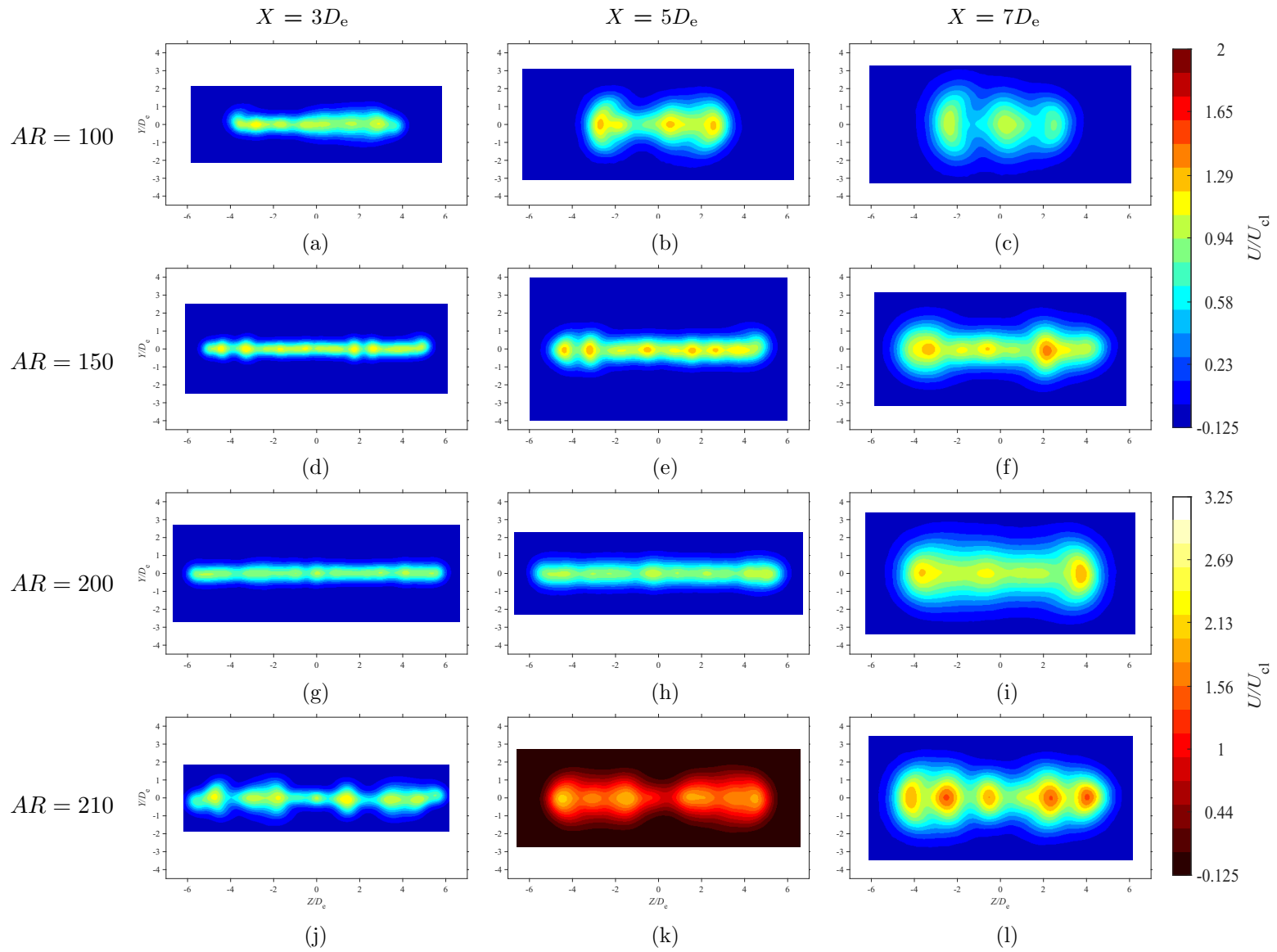


Figure 5.5: Average streamwise velocity maps at each in-plane location for each AR configuration at $Re = 18000$

Lateral Velocity Maps

Figure 5.6 shows the lateral velocity maps for every AR configuration at each cross-sectional plane, and have been nondimensionalized by U_{cl} . When observing V , it is clear that the entrained velocity is not constant in the spanwise direction. With reference to Figure 5.5, it is observed that V is entrained more where U peaks are located. Such an occurrence is in agreement with the ideas presented by Quinn et al. (1983), where velocity entrainment increases in specific areas to form the velocity peaks of the saddle-back profile. With respect to the works of Krothapalli et al. (1981) and Lozanova and Stankov (1998), the amount of entrained velocity at the jet boundary does not appear to decrease downstream. But, it is important to note that their observations were significantly further downstream than in this study. No clear AR dependency is visible for V , but it is observed that the magnitude of U peaks directly affect the magnitude of V peaks. As the U peaks increase in magnitude, so do the V peaks.

Spanwise Velocity Maps

Figure 5.7 shows the spanwise velocity maps for every AR configuration at each cross-sectional plane, and have been nondimensionalized by U_{cl} . In general, W does not appear to have large variation, with slight increases in magnitude as the flow evolves downstream. Along the centerline, many of the W profiles appear symmetrical, which is in agreement with Quinn (1992b). However, there is a distinct change in the flow field of $AR = 210$ at $X = 3D_e$, which corresponds to Figure 5.7j. The reason for multiple negative/positive W pairs appearing may be due to the unique shape of U seen in Figure 5.5j. Four distinct *bulges*, as described by Vouros et al. (2015), form at the ends of the jet. It is observed that the formation of these bulges allows for more velocity to be entrained towards the local lateral edges of the bulges themselves. This local entrainment helps to explain the growth of the four bulges to form two distinct bulges at $X = 5D_e$. No AR dependency is observed for W , nor does there seem to be a U peaks dependency.

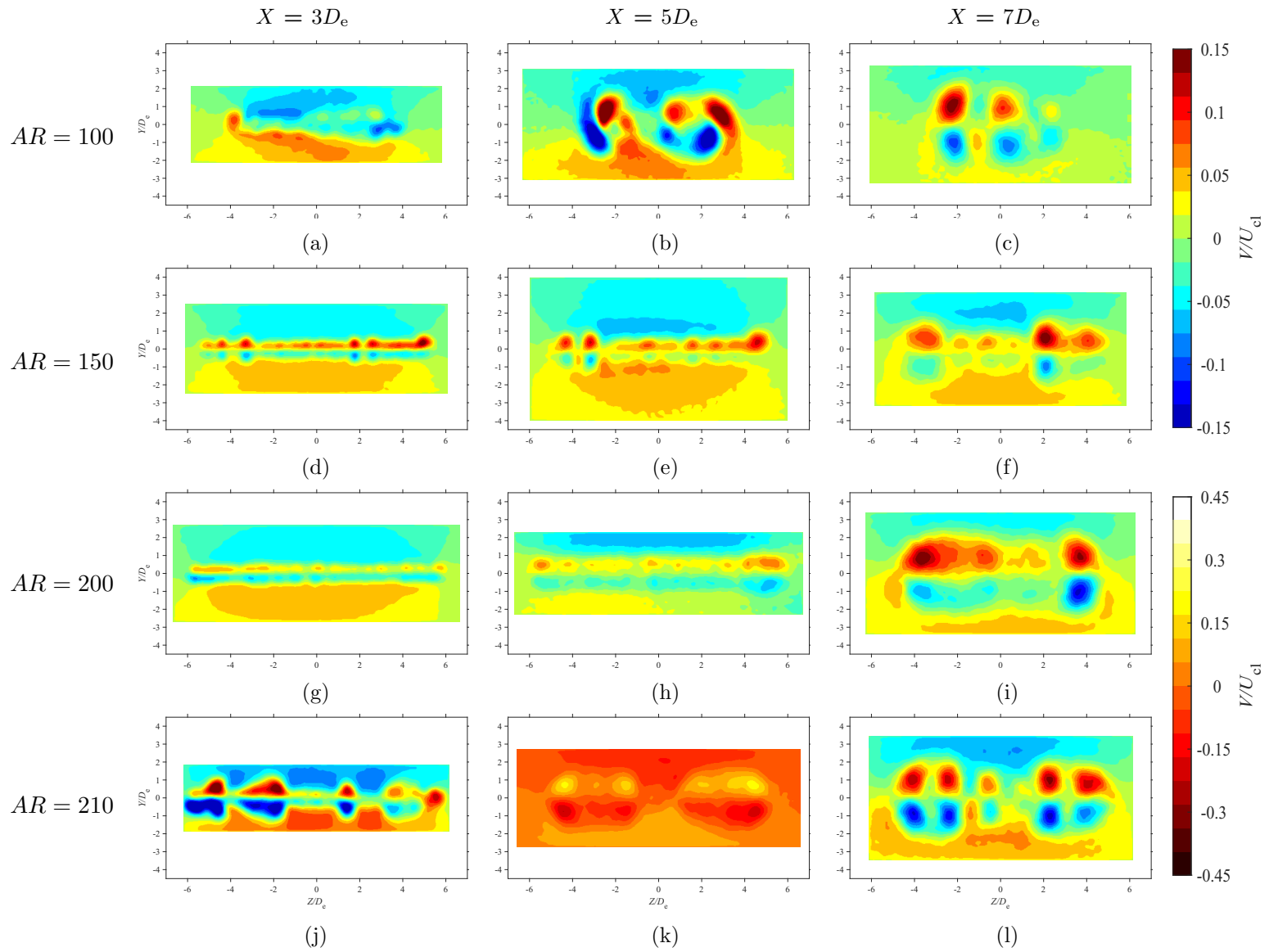


Figure 5.6: Average lateral velocity maps at each in-plane location for each AR configuration at $Re = 18000$

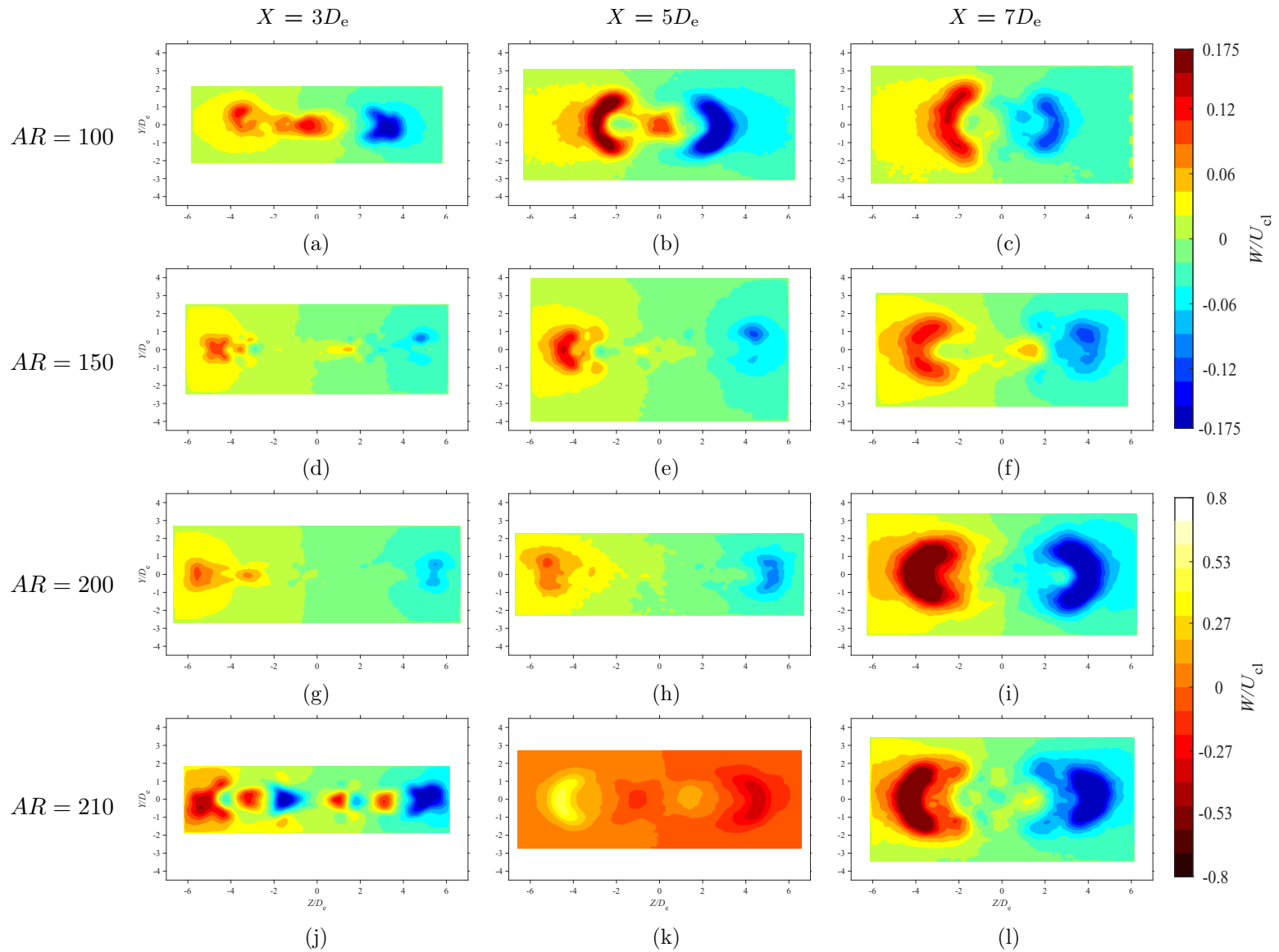


Figure 5.7: Average spanwise velocity maps at each in-plane location for each AR configuration at $Re = 18000$

5.3.2 Whole-Field Average Turbulence Intensity Measurements

Figures 5.8, 5.9, and 5.10 are provided in the following three pages, and they correspond to the whole-field measurements for u' , v' , and w' respectively. Each turbulence intensity field has been nondimensionalized by U_{cl}^2 .

5.3.3 Streamwise Turbulence Intensity Measurements

From the whole-field u' measurements presented in Figure 5.8, some clear observations can be made. First, when compared to the whole-field streamwise plots in Figure 5.5, there is good agreement with u' . That is, minimum and maximum u' values occur in regions where there are high velocity gradients (i.e., regions with high shear stress), which occurs near the U peaks observed in Figure 5.5. When considering the shape of u' in the lateral direction, some AR s, such as $AR = 150$ and $AR = 200$, clearly show a saddle-back shape on the centerline, which is in agreement with the observations made by Sfeir (1979) and Krothapalli et al. (1981). Further downstream, the jets' saddle-back distributions on the centerline begin to subside, as is to be expected as axisymmetry is approached. When considering the spanwise direction, it is observed that the shape of u' in the spanwise direction differs appreciably in comparison to the lateral direction. This is expected with the formation of many U peaks. However, the formation of the saddle-back shape described by Krothapalli et al. (1981) is observed for $AR = 100, 150,$ and 200 , which is indicative of the beginning of the merging of the two shear layers separated in the spanwise direction.

5.3.4 Lateral and Spanwise Turbulence Intensity Measurements

With regards to v' and w' , fewer observations can be made when looking at Figures 5.9, and 5.10. When compared to their V and W counterparts, both v' and w' have small and large values occurring in regions where there are appreciable velocity gradients. As discussed by Krothapalli et al. (1981), the values of both v' and w' are of similar magnitudes, but it is noted that w' appears to be slightly larger in value than v' . Krothapalli et al. also observed that v' and w' , have approximately the same values on the cross-sectional planes. This is not observed to be the case for some of the jet configurations. However, the measurements conducted by Krothapalli et al. were further downstream in the ATDR, whereas the present study's measurements were confined in the CDR. As such, the discrepancies between both studies' observations are expected.

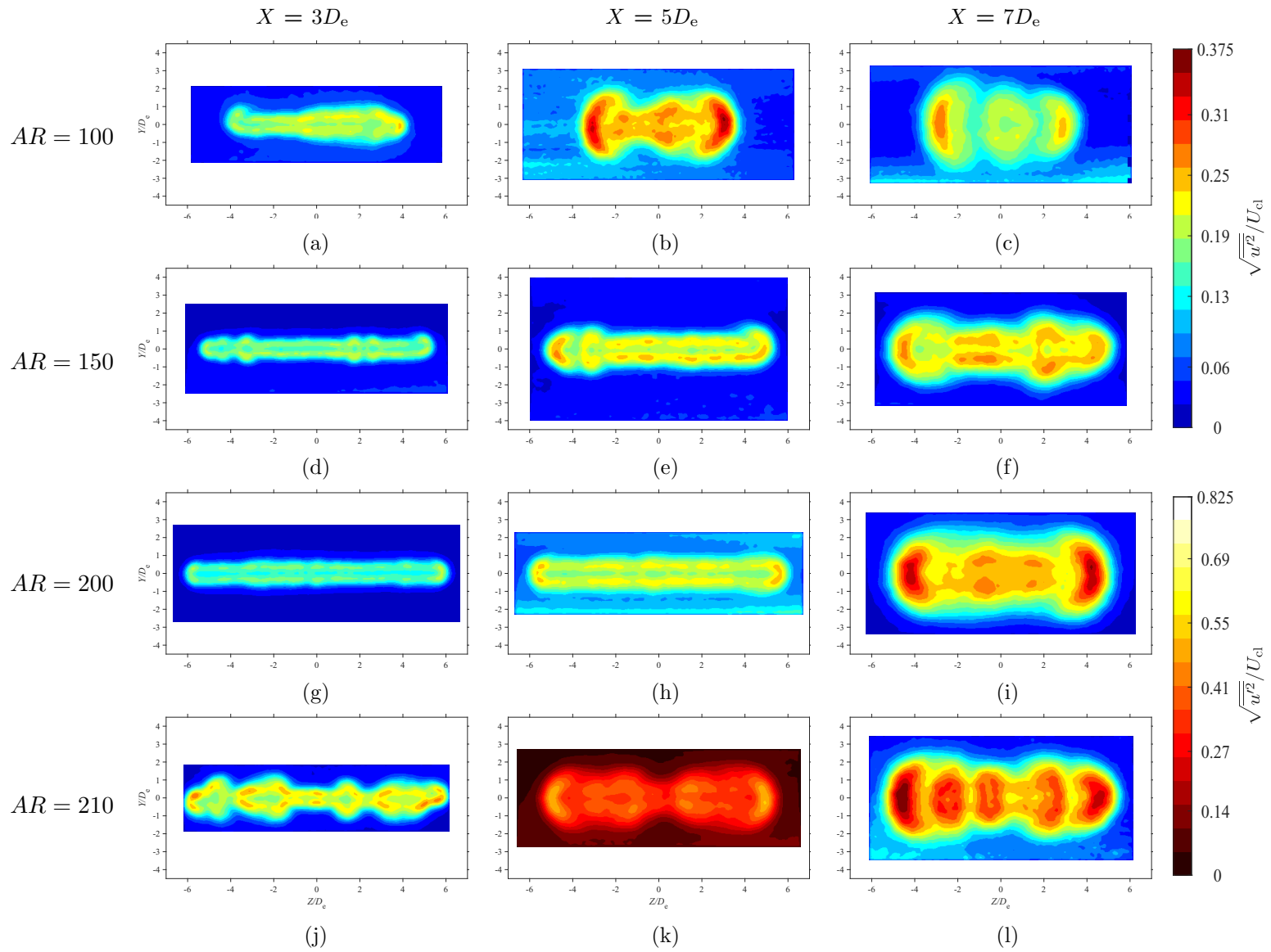


Figure 5.8: Average streamwise turbulence intensity maps at each in-plane location for each AR configuration at $Re = 18000$

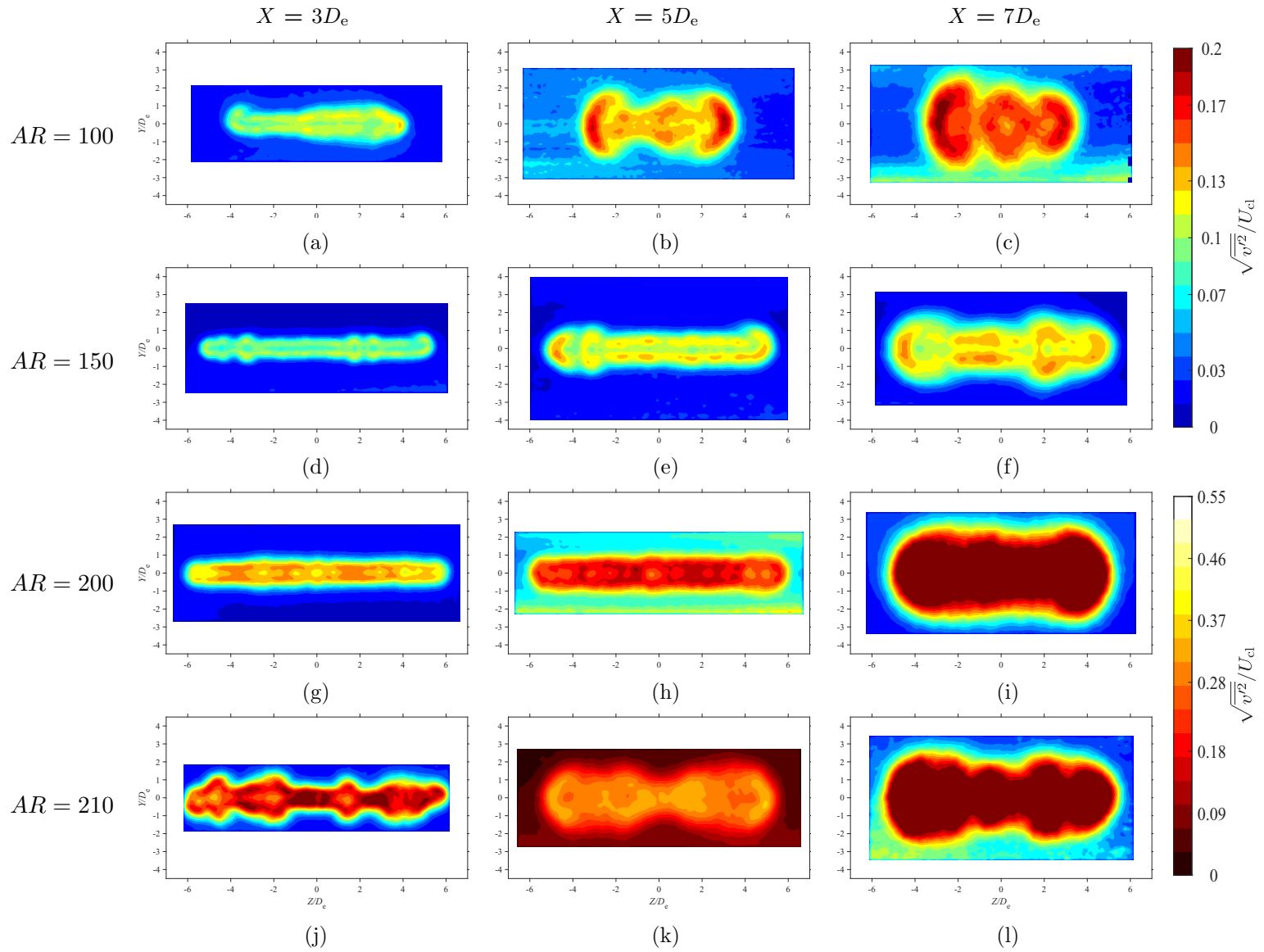


Figure 5.9: Average lateral turbulence intensity maps at each in-plane location for each AR configuration at $Re = 18000$

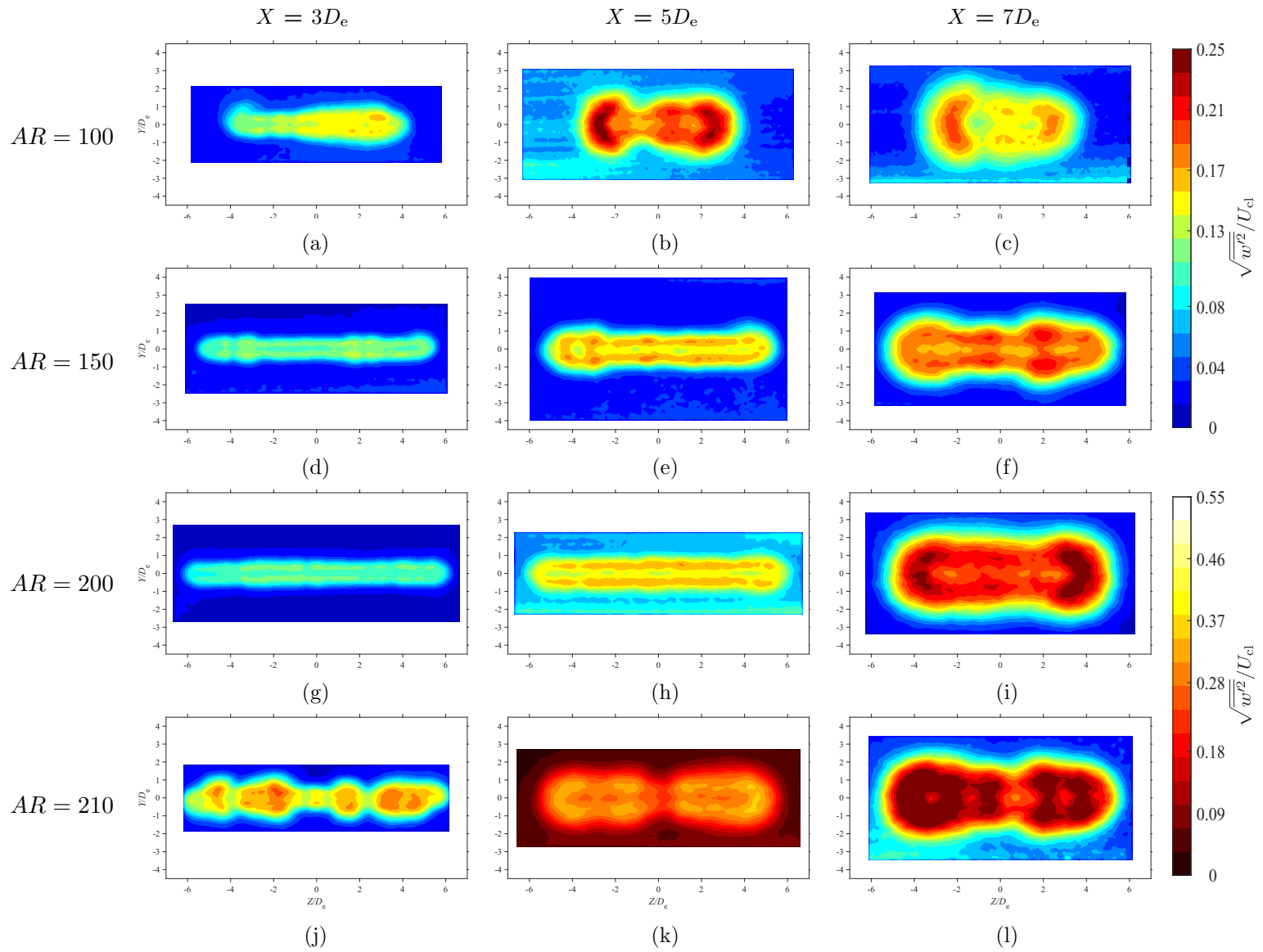


Figure 5.10: Average spanwise turbulence intensity maps at each in-plane location for each AR configuration at $Re = 18000$

5.3.5 Reynolds Shear Stresses

Figures 5.11, and 5.12, are provided in the following two pages, and they correspond to the whole-field measurements for $\overline{u'v'}$, and $\overline{u'w'}$ respectively. Each Reynolds shear stress field has been nondimensionalized by $U_{cl}^2 \times 100$.

5.3.6 Lateral Reynolds Shear Stress

The lateral Reynolds shear stress maps are shown in Figure 5.11. When compared to the whole-field plots in Figures 5.5, and 5.8, there appears to be good agreement with the observations made by others. The lateral Reynolds shear stress, appears to be well correlated with the strain ($\partial U/\partial y$) - that is, where large U gradients in the lateral direction occur, as was also the case for u' . This was noted to occur near the U peaks observed in Figure 5.5. This observation is in line with the works of Quinn (1992b) and Lozanova and Stankov (1998). It is also observed that there is a dependency of $\overline{u'v'}$ with AR . As AR is increased, it appears that $\overline{u'v'}$ increases as well, which is in agreement with the observations of Quinn (1992b).

5.3.7 Spanwise Reynolds Shear Stress

Analogous observations can be made when looking at the spanwise Reynolds shear stress maps in Figure 5.11. When compared to the whole-field plots in Figures 5.5, and 5.9, similar trends are seen, but instead in the spanwise direction. The spanwise Reynolds shear stress appears to be well correlated with the strain ($\partial U/\partial z$) such that large minimum and maximum $\overline{u'w'}$ pairs occur near U peaks. This is strongly observable in Figure 5.5l, where five minimum/maximum pairs form around the five U peaks. Overall, this observation is in line with the works of Quinn (1992b) and Lozanova and Stankov (1998). Again, there is a dependency of $\overline{u'w'}$ with AR , where $\overline{u'w'}$ increases as AR increases, which is again in agreement with the observations of Quinn (1992b).

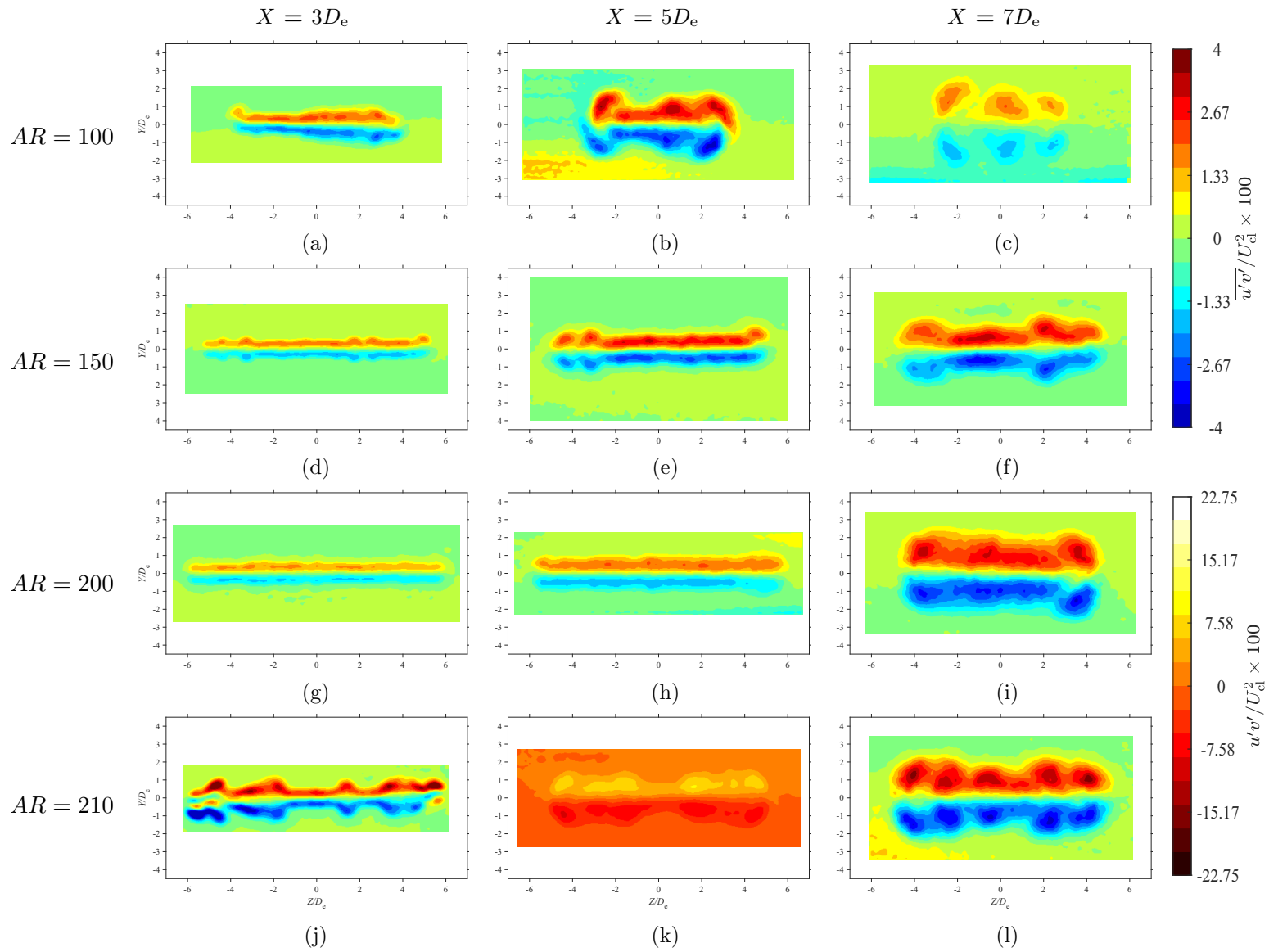


Figure 5.11: Average lateral Reynolds shear stress maps at each in-plane location for each AR configuration at $Re = 18000$

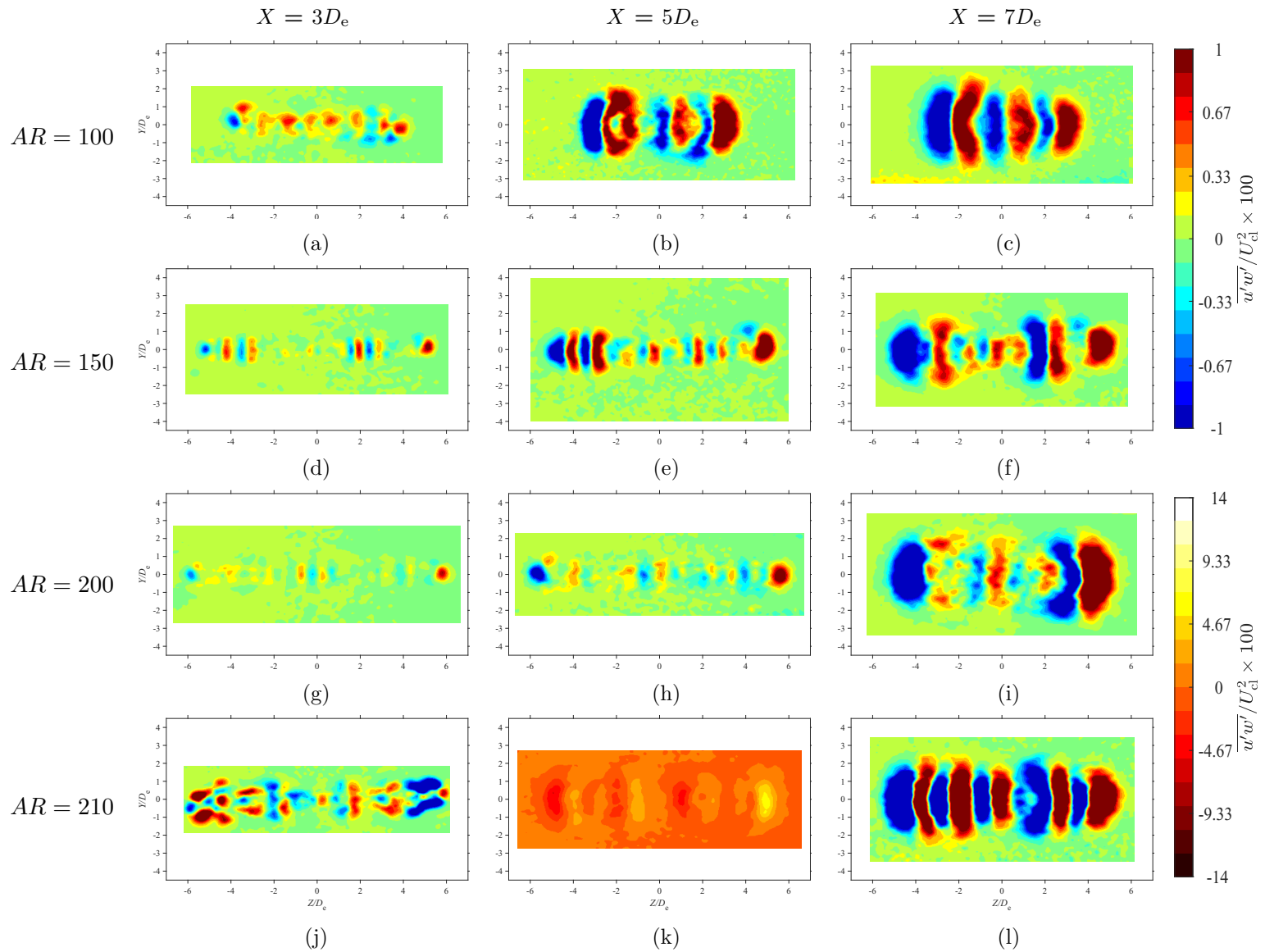


Figure 5.12: Average spanwise Reynolds shear stress maps at each in-plane location for each AR configuration at $Re = 18000$

5.3.8 Turbulent Kinetic Energy

The whole-field measurements for the turbulent kinetic energy are presented within Figure 5.13. Each TKE field has been nondimensionalized by $U_{cl}^2 \times 100$. At $X = 7D_e$, similar observations to those of Quinn (1992b) may be made. First, the TKE is higher within the $X-Z$ plane than the $X-Y$ plane at this location. This is due to the multi-peak U profiles located along the spanwise direction causing high velocity gradients, which in turn increase the production of TKE . Secondly, at this location, as AR increases, TKE increases. At $X = 3D_e$ and $5D_e$, these trends do not occur, however. This is likely a result of the velocity gradients causing the production of TKE along the jet perimeter rather than on the cross-sectional planes. Nonetheless, from this study, the streamwise velocity gradients are found to be important in the production of TKE , which is in agreement with Quinn (1992b).

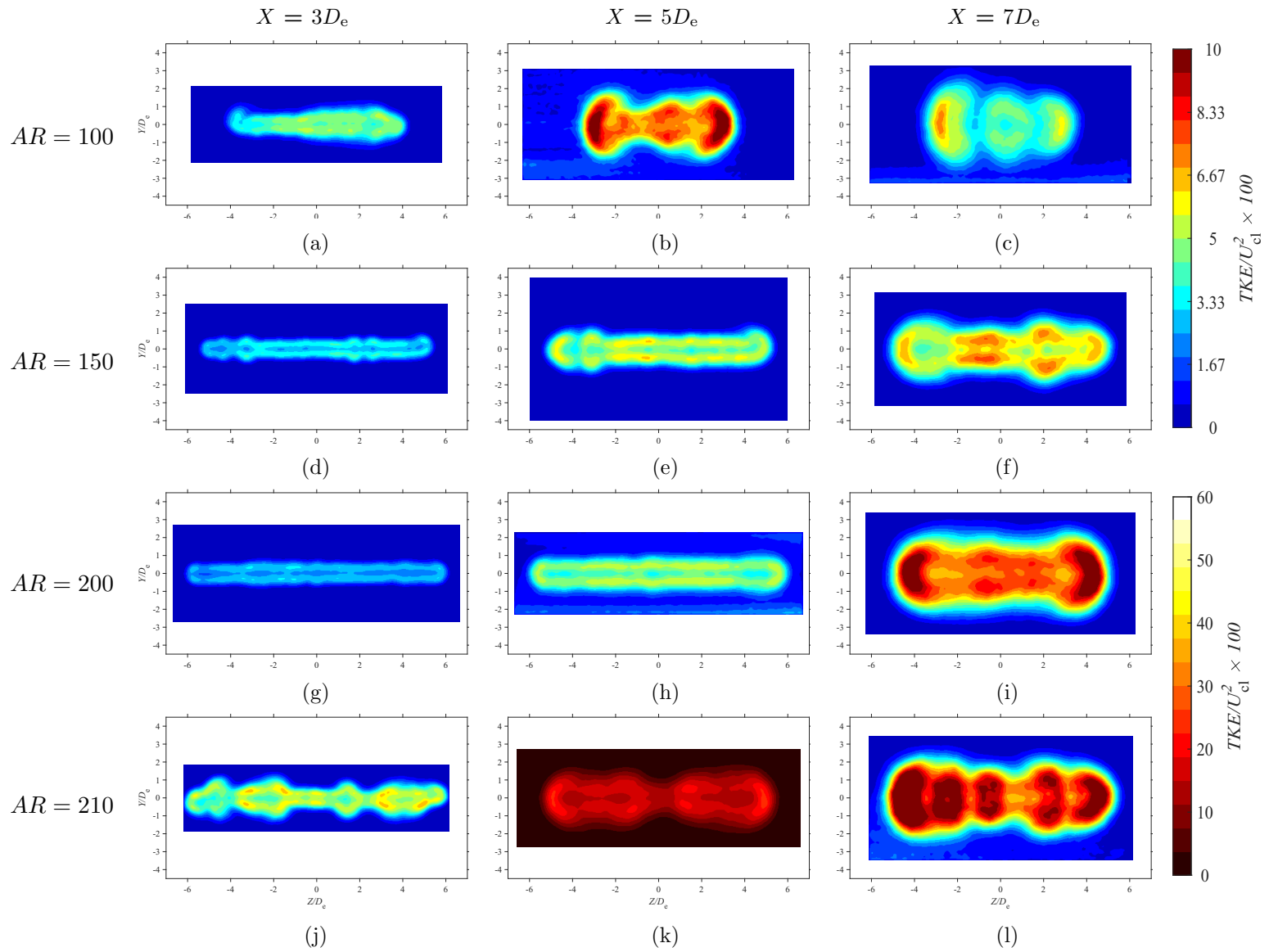


Figure 5.13: Average TKE maps at each in-plane location for each AR configuration at $Re = 18000$

5.3.9 Streamwise Vorticity

The whole-field measurements for the streamwise vorticity, ω_x , are presented with Figure 5.14. Each field has been nondimensionalized by D_e/U_{cl} . Throughout the preceding discussion, many of the observed trends for high- AR configurations have gone against the trends previously observed by others for low- AR s. Based on Figure 5.5, a noticeable deviation was the emergence of multiple U peaks along the spanwise direction. Based upon the works of Quinn (1992a) and Zaman (1996), it was proposed that a change in the ω_x -dynamics had occurred. Clearly, from the results presented within Figure 5.14, this is indeed the case, as more than four counter-rotating vortex pairs are found to exist in each jet configuration at $X = 3D_e$. As each jet flows downstream, the number of streamwise vortices appears to lower, which is likely due to the complex interaction of the ω_θ -dynamics (azimuthal vortex rings) and the ω_x -dynamics, as discussed by Zaman (1996). The reasoning for the formation of a higher than normal amount of counter-rotating vortex pairs is likely a direct byproduct of AR . Though outside the scope of this study, it is likely that there is some $AR_{critical}$ at which more than four counter-rotating vorticity pairs begin to form. As previously discussed when considering jet asymmetries, small-scale disturbances can greatly affect the flow. By considering that the formation of streamwise vortices is due to pressure imbalances that form at the slot exit (Quinn et al., 1983), it is likely that these small-scale disturbances directly affect the near-exit pressure field, which in turn leads to more vortex pairs forming. Nonetheless, it is clear that high- AR jets display higher three-dimensionality, which affects the flow field far more than previously seen for low- AR studies.

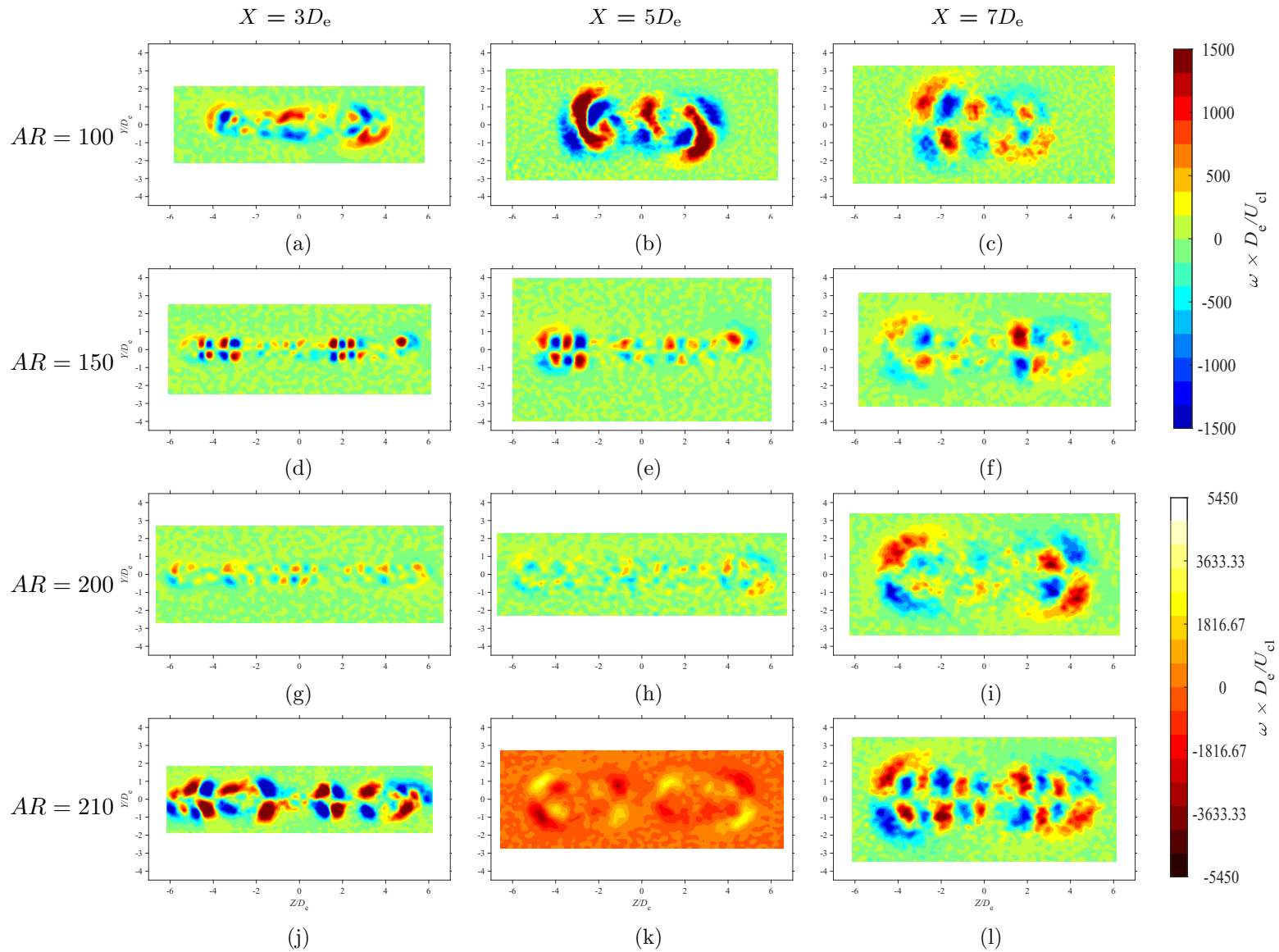


Figure 5.14: Average vorticity maps at each in-plane location for each AR configuration at $Re = 18000$

5.3.10 Cross-Sectional Boundary Deformation

As discussed in Chapter 2, the interaction of the ω_θ -dynamics and the ω_x -dynamics affect the downstream cross-sectional changes in a jet. As the vorticity fields in Figure 5.14 produced more vortex pairs than expected, the jet boundaries are presented with Figure 5.3.10. As mentioned in Chapter 4, the jet boundaries are found by considering where in the whole-field a $0.95U_{cl}$ criteria is fulfilled. No AR -based dependencies are observed for the boundaries. This is due to the sensitivity of the vorticity dynamics to the upstream slot conditions. The vastly different boundaries shapes help to explain the spreading trends observed in Figure 5.3, where it appeared that $AR = 100$ spread much more quickly than the other configurations, and $AR = 150$ and $AR = 200$ appeared to spread at the same rate as each other. From such results, it is observed that cross-sectional measurements are far more necessary for high- AR jets, as their complexity cannot be simply explained through classical characteristics modeling (i.e., velocity decay, and velocity half-width).

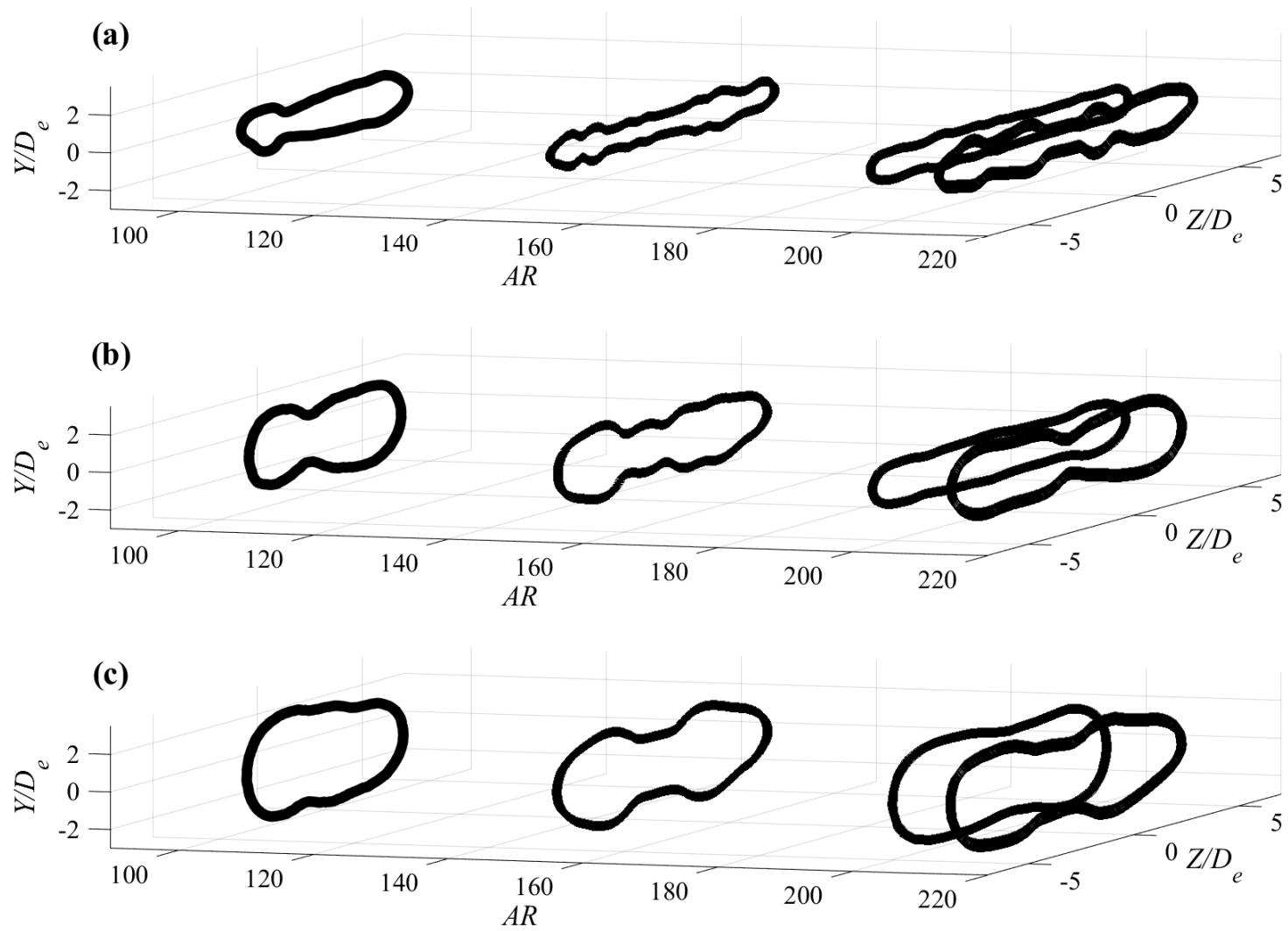


Figure 5.15: Jet boundary evolution at $Re = 18000$ at (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$

5.4 Conclusions

The streamwise and cross-sectional measurements of free, rectangular jets were taken using 2D- and stereo-PIV techniques, respectively. The PIV experiments measured the velocity field within the near-exit region, which consisted of the PCR and CDR. Within the present experimental investigation, the influence of high AR s was of interest, with specific interest on AR configurations of $AR = 100, 150, 200,$ and 210 .

The 2D-PIV measurements allowed for the jets' velocity decays, streamwise turbulence intensities, and lateral velocity half-widths to be examined. By considering each variable, some agreement was found to exist between the observations made by others on low- AR configurations and the configurations considered in this study. However, notable deviations in trends were also found, which indicated an overall change in the flow conditions for the high- AR configurations. Through 2D-PIV alone, these changes could not be accounted for, so in turn, the data from the stereo-PIV experiments were looked towards in order to gain a supplementary understanding of the flows.

The stereo-PIV measurements proved to be far more insightful, in that they revealed much more about the four jet configurations' downstream evolution at various cross-sectional planes. The average streamwise velocity measurements showed that each configuration had markedly different cross-sectional profiles, with differing in-plane velocity distributions. Notably, none of the U -velocity measurement planes exhibited a spanwise saddle-back profile, but instead each had more U -peaks than observed by other researchers. The differing in-plane velocity distributions led to differing observations on the nature of the turbulence intensities, shear stresses, and turbulent kinetic energy.

When the vorticity dynamics of each configuration was considered, it was found that the distribution of the streamwise vortices were significantly different than seen for low- AR configurations. Rather than having four streamwise vortex pairs, each high- AR configuration had a higher number of pairs. It was concluded that the formation, interaction, and evolution of these ω_x -dynamics led to the overall deviation in common trends typically observed for free, rectangular jets. Overall, it was concluded that AR has a significant influence on the formation of the ω_x -dynamics. However, it was also concluded that the sensitivity of the system to small-scale disturbances (e.g., manufacturing defects, and setup misalignments) also strongly influenced the formation of the ω_x -dynamics, which not only lead to large changes in the observed variables' in-plane distributions, but also lead to large asymmetries in the jets themselves.

Chapter 6

Influence of Reynolds Number

6.1 Introduction

The studies that have been conducted on free, rectangular jets have inclusively focused on AR and its influence towards the downstream flow field. Limited focus has been put towards the affects of Re , however. The work of Dimotakis (2000) has discussed the concept of a critical Re at which flows are fully turbulent. Beyond such a limit, and before the point at which compressibility effects take place, it is expected that no substantial changes to the flow field occur. It is observed that Dimotakis's ideas have not been thoroughly investigated. That is, the occurrence of the critical Reynolds number for rectangular jets has not been looked for by experimentalists. Thus, this study presents an analysis of jets at $Re = 10800, 14400, 18000, 21600, 25200,$ and 28800 , at an aspect ratio of $AR = 150$. The overall layout of this chapter follows a similar layout as the previous chapter, where the two main sections discuss data trends pertaining to streamwise and cross-sectional observations, respectively. Again, the streamwise results are extracted from the 2D-PIV measurements that were taken within the X - Y plane, and the cross-sectional observations are from the stereo-PIV measurements, which were taken downstream of the X - Z plane. Following the two main sections, the chapter closes with a section detailing the conclusions made from the discussions within the two aforementioned sections.

6.2 Streamwise Flow Field Observations

The results of the 2D-PIV measurements are presented herein. As before, these results are presented first as they are simpler to observe, and their differences between other studies are easier to discern. As demonstrated in the previous chapter, any deviations in trends can then be further investigated by looking towards the results from the stereo-PIV measurements.

6.2.1 Centerline Decay

The nondimensionalized velocity decay profiles for all six Re configurations is presented in log-log form in Figure 6.1. With reference to the annotation line underneath the data sets, it appears that between $X = D_e$ and $X = 2D_e$, all six configurations begin to display a planar jet-like decay rate. The decay rates appear to vary slightly among all six configuration such that an apparent crossing over of data sets near $X = 2D_e$. Beyond this location, the data sets do not appear to fully collapse onto each other. This could be partially explained as being caused by uncertainties in the data, which were introduced by setup misalignments or the data processing scheme. The two lowest Reynolds numbers ($Re = 10800$ and $Re = 14400$) appear to deviate significantly from the other data sets, which indicates that the two flows are not fully turbulent flow. It is also observed that as Re increases, the PCR shortens. This observation appears to be in contradiction with the works of Barber et al. (2017) and Krothapalli et al. (1981). It is likely the added influence of a high AR configuration is likely the reason for this discrepancy, as the trends for high- AR s were observed to not conform to the established trends of low- AR configurations in Chapter 5 (Figure 5.1a).

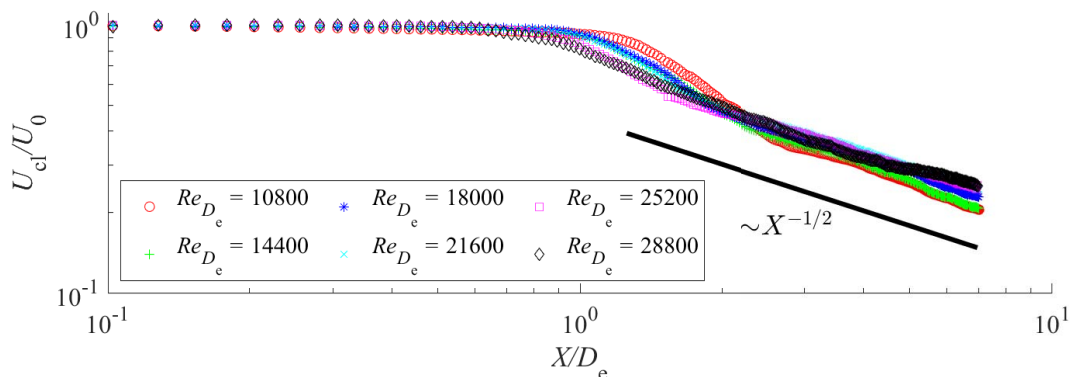


Figure 6.1: Velocity decay at $AR = 150$ with the X -axis nondimensionalized by equivalent diameter. Every 5th data point has been plotted for clarity

6.2.2 Streamwise Velocity Turbulence Intensity

The nondimensional, downstream turbulence intensities along the centerline are provided in Figure 6.2. The overall trends of all six data sets are observed to be inline with the observations originally described by Sfeir (1979), with an initial increase in u' towards a peak value, followed by a decrease in value, and then an increase in value towards an asymptotic value. Again near the peak value, some data sets exhibit a disparity in data, which is a result of the stitching methodology described in Chapter 4.

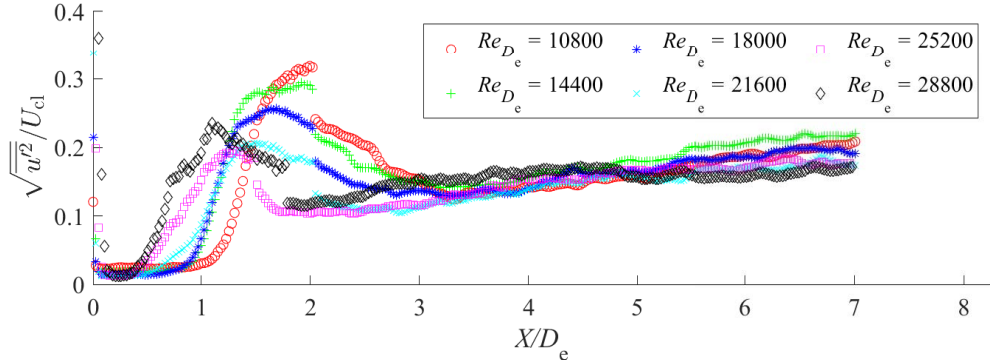


Figure 6.2: Normalized turbulence intensity against X/D_e at $AR = 150$. Every 5th data point has been plotted for clarity

In Figure 6.2, some trends appear to occur as Reynolds number increases. First, the magnitude of the turbulence intensity peaks appear to have a linear trend, with the exception of $Re = 28800$. As Re increases, the peak magnitude decreases, but increases once $Re = 28800$ is reached. This may be due to slight compressibility effects taking place in the region for $Re = 28800$. The maximum peak location also appears to have a linear trend. An increase in Re results in a shortening of the maximum peak location for all Reynolds numbers. This observation contradicts the observations in the works of Sfeir (1979) and Tsuchiya and Horikoshi (1986). As previously noted, this is likely due to the added influence of a differing cross-sectional exit profile for this study. It is observed that an asymptotic value near 0.2 is being approached by all data sets downstream, but the true asymptotic value can not be discerned.

6.2.3 Velocity Half-Width

Figure 6.3 presents the lateral velocity half-widths against the downstream distance, which have both been nondimensionalized by D_e . Only the positive $Y_{1/2}$ values have been plotted within Figure 6.3, but the plots provided within Appendix D provide both positive and negative $Y_{1/2}$ values. Of the six $Y_{1/2}$ data sets, $Re = 10800$ and $Re = 14400$ appear to deviate significantly from the other data sets. The two data sets do not collapse with the other four, which remain nearly collapsed for the entirety of the FOV. This agrees with the previous observation that the two flows are not fully turbulent flows. For higher Reynolds numbers, the variation in $Y_{1/2}$ is minimal, which indicates the development of fully turbulent flow, and a weak influence of Re . Both observations are in agreement with the findings of

Barber et al. (2017).

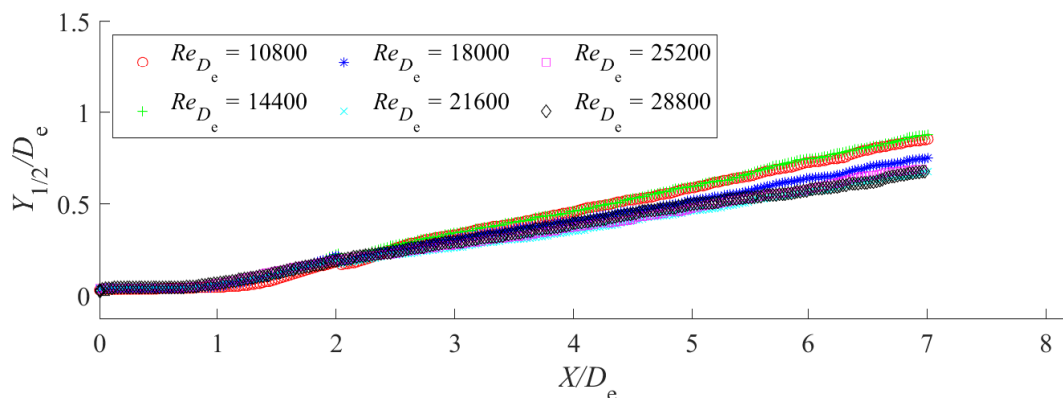


Figure 6.3: Normalized lateral velocity half-width against X/D_e at $AR = 150$. Every 5th data point has been plotted for clarity

6.3 Cross-sectional Flow Field Observations

Though streamwise measurements have been useful for downstream flow analyses for low- AR configurations, the preceding chapter has proven that much more information is gained when considering cross-sectional measurements. Thus, the following subsections discuss the in-plane nature of rectangular jet configurations at $X = 3D_e$, $5D_e$, and $7D_e$. For each flow variable considered, the results are presented in a similar way. The whole field maps have been ordered by row with respect to Re , and by column with respect to downstream measurement plane. Not all fields are able to be observed together, so the whole-field plots have been arranged within two pages, which both contain 9×9 figures. Every whole-field plot is available in Appendix E. The plots have been normalized by using U_{cl} or U_{cl}^2 , as is common for cross-sectional measurements. The centerline velocity is the local cross-sectional velocity located at $Y = 0$ and $Z = 0$. Every plot was made to have the same colormap limits, with the exception of a few, which needed to have different colormap limits. The need for different colormaps is a result of the difficulties in using U_{cl} as a nondimensional parameter, as discussed in the previous chapter. As such, the figures have been given a different colormap to visually highlight their larger colormap limits.

6.3.1 Whole-Field Average Velocity Measurements

Streamwise Velocity Maps

Figures 6.4 and 6.5 show the streamwise velocity map for every Re configuration at each cross-sectional plane. Each velocity field has been nondimensionalized by U_{cl} . When con-

sidering all six jet configurations, there appears to be an Re dependency. When looking at $X = 3D_e$, some Reynolds numbers appear to look very similar to each other, and also appear to evolve in a similar way downstream.

Initially, $Re = 10800$ and 14400 have a similar shape to each other at $X = 3D_e$. The same is observed for the other configurations, with $Re = 18000$ and 21600 , and $Re = 25200$ and 28800 having similar shapes as each other. With regards to an Re -based trend, it appears that the bulges, as described in the preceding chapter, dampen as Re increases. That is, the outwardly protrusions of the bulges become less pronounced as Re increases. When moving towards $X = 5D_e$, the jets become even more similar to each other in shape, with $Re = 10800$, 14400 , and 18000 , and $Re = 21600$, 25200 , and 28800 appearing to be similar to each other. Again, as Re increases, the bulges that exist appear to dissipate, with a distinct change appearing between $Re = 18000$ and 21600 . At $X = 7D_e$, the apparent dampening of the bulges continues to occur, with each Re showing a dampening of bulges when compared to the previous Re . Clearly, there is an Re -dependency displayed among the results, and based upon the results at $X = 3D_e$, it appears that there may be some critical Reynolds number that has been passed between $Re = 21600$, and 25200 . This is based upon the observation that $Re = 25200$ and 28800 appear to be nearly identical to each other. Some differences between the evolution of the two downstream are observed however, but may be due to slight measurement errors, or slight upstream disturbances.

It is observed that some clear changes are present when the centerline velocity profiles in the cross-planes are considered. Specifically, when the spanwise direction is considered, it is clear that none of the configurations exhibit a saddle-back velocity profile, but the amount of peaks along the spanwise-axis appear to dissipate with Re . That is, the velocity profile in the spanwise direction becomes more constant. The dissipation of the peaks is indicative that the vorticity dynamics are changing as Re increases, and suggests that the mechanisms behind the formation of the streamwise vortices are becoming dampened as the inertial forces of the flow begin to dominate. When considering the lateral velocity profiles along the spanwise direction, some variations are observed to exist, but reside as Re increases. This is clearly due to the formation of multiple velocity peaks and troughs in the spanwise direction. However, as Re increases, the amount of peaks and troughs dissipate, and as a result, the lateral velocity on the centerline become more similar.

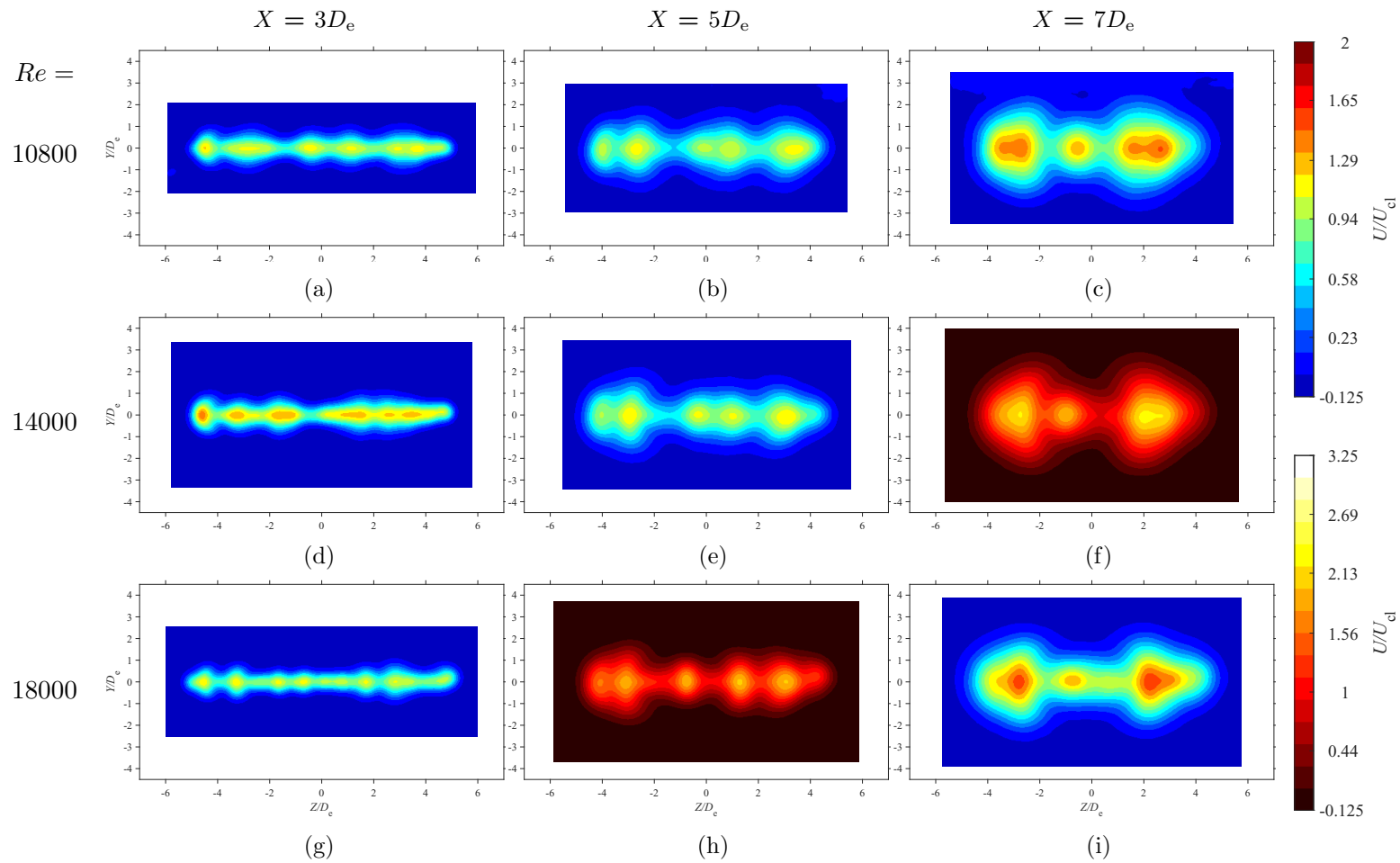


Figure 6.4: Average streamwise velocity maps at each in-plane location for $Re = 18000$, 14400 , and 18000 at $AR = 150$

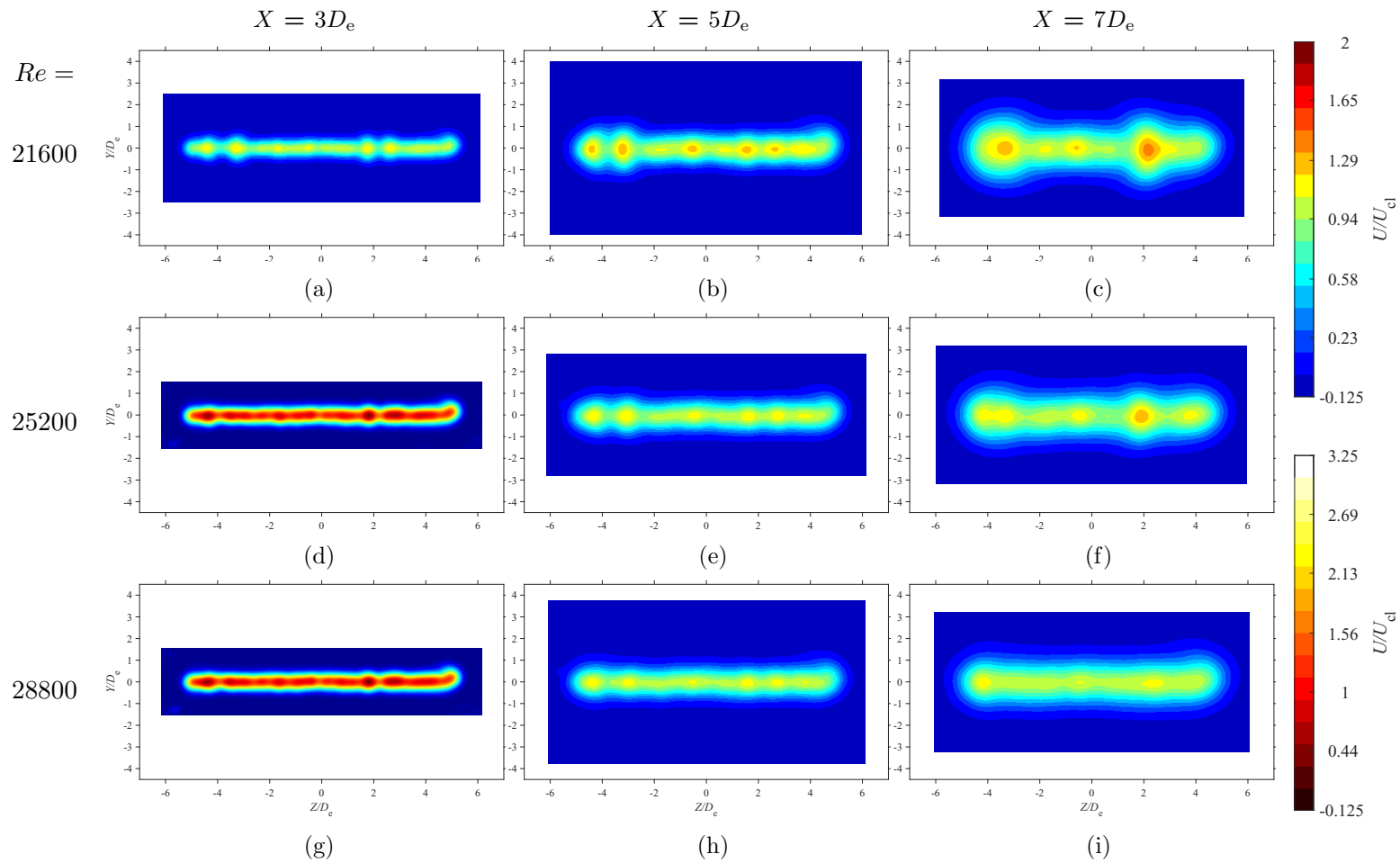


Figure 6.5: Average streamwise velocity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

Lateral Velocity Maps

Figures 6.6 and 6.7 shows the lateral velocity maps for every Re at each cross-sectional plane, which have been nondimensionalized by U_{cl} . With reference to the U fields, and by observing the V fields, it is clear that the entrained velocity is not constant in the spanwise direction, which is a result of the U peaks that form along the spanwise direction. As previously observed, V is entrained more where U peaks are located, which is in agreement with Quinn et al. (1983). Notably, as the bulges of the jet dissipate with an increase in Re , the entrainment of V becomes more constant along the spanwise direction, which is expected with a spanwise-consistent flow. Also, it is observed that after $Re = 18000$, the velocity entrainment at the jet boundary decreases downstream, which agrees with the works of Krothapalli et al. (1981) and Lozanova and Stankov (1998). However, no apparent Re -based trend exists with V , as the magnitudes of V do not linearly change as Re increases.

Spanwise Velocity Maps

Figures 6.8 and 6.9 show the spanwise velocity maps for every Re configuration, and have been nondimensionalized by U_{cl} . It is observed that as Re increases, W has less negative/positive pairs that exist within the flow. This is a consequence of the U bulges dissipating as Re increases. As they disappear, the local entrainment around them (i.e., the negative/positive V pairs) also disappears. As a result, $Re = 25200$ and 28800 exhibit no variation of W along the centerline. The reason for multiple negative/positive W pairs appearing may be due to the unique shape of U seen in Figure 5.5j. As previously noted, this local entrainment helps to explain the formation of the dumbbell shape described by Vouros et al. (2015), which is clearly observable for $Re = 10800$, 14400 , and 18000 at $X = 7D_e$. For $Re = 21600$, 25200 and 28800 , there appears to be an Re -based phenomenon, as the magnitude of W decreases for each jet as they move downstream. This gives further credence to the idea that the $Re_{critical}$ for fully developed turbulent flow occurs somewhere after $Re = 21600$, as it is expected that flows behave similarly once this transition value is reached.

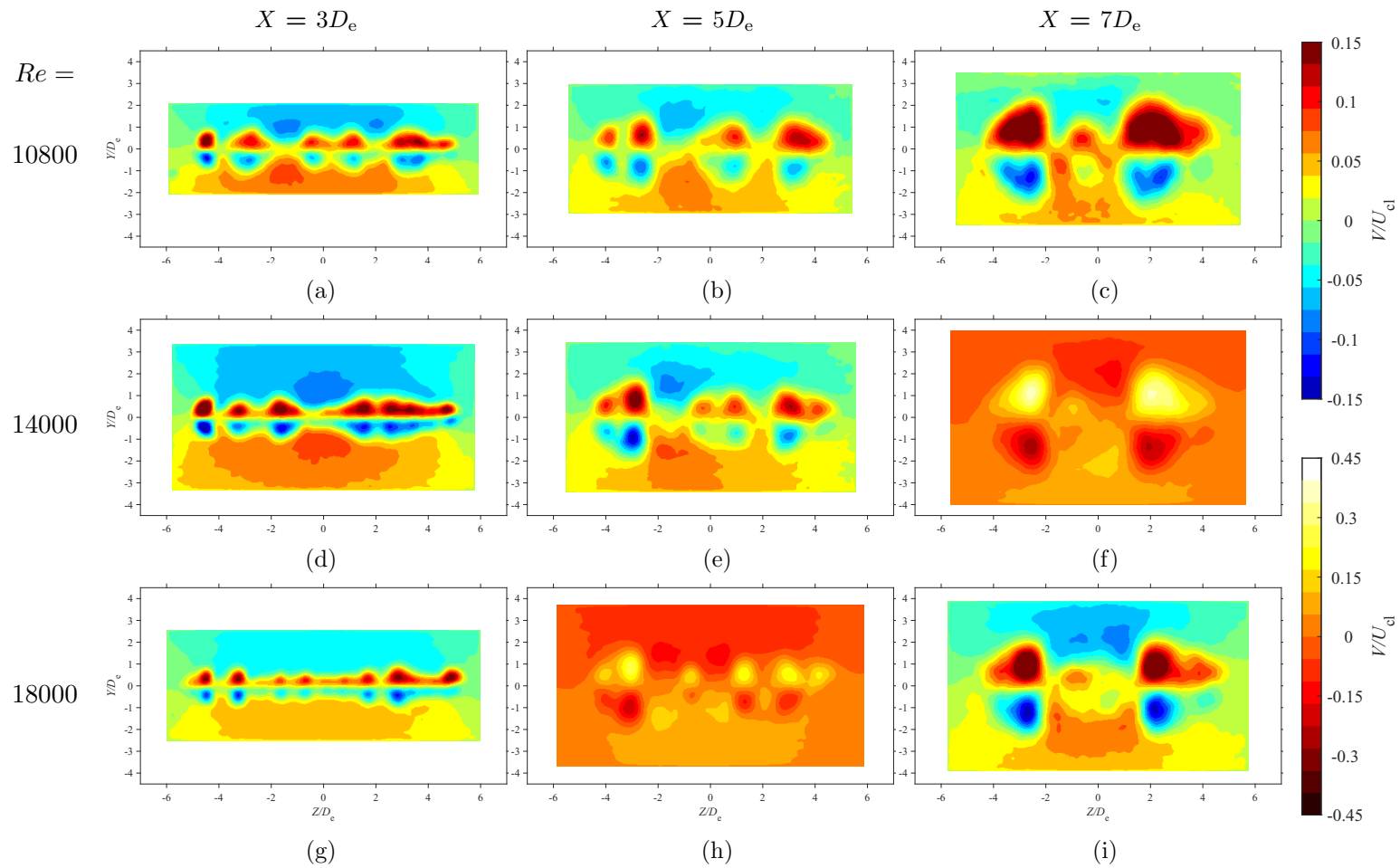


Figure 6.6: Average lateral velocity maps at each in-plane location for $Re = 18000$, 14400 , and 18000 at $AR = 150$

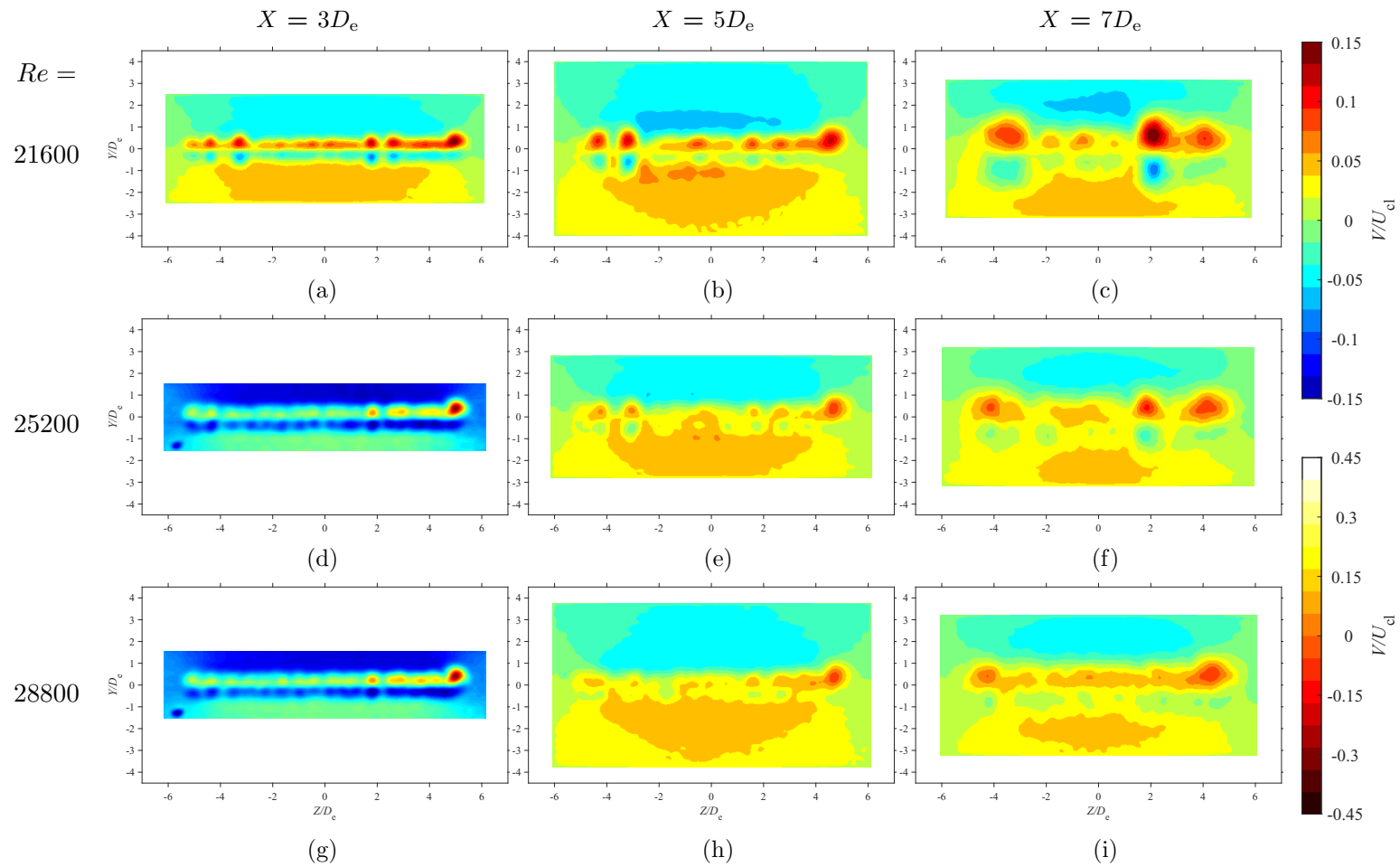


Figure 6.7: Average lateral velocity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

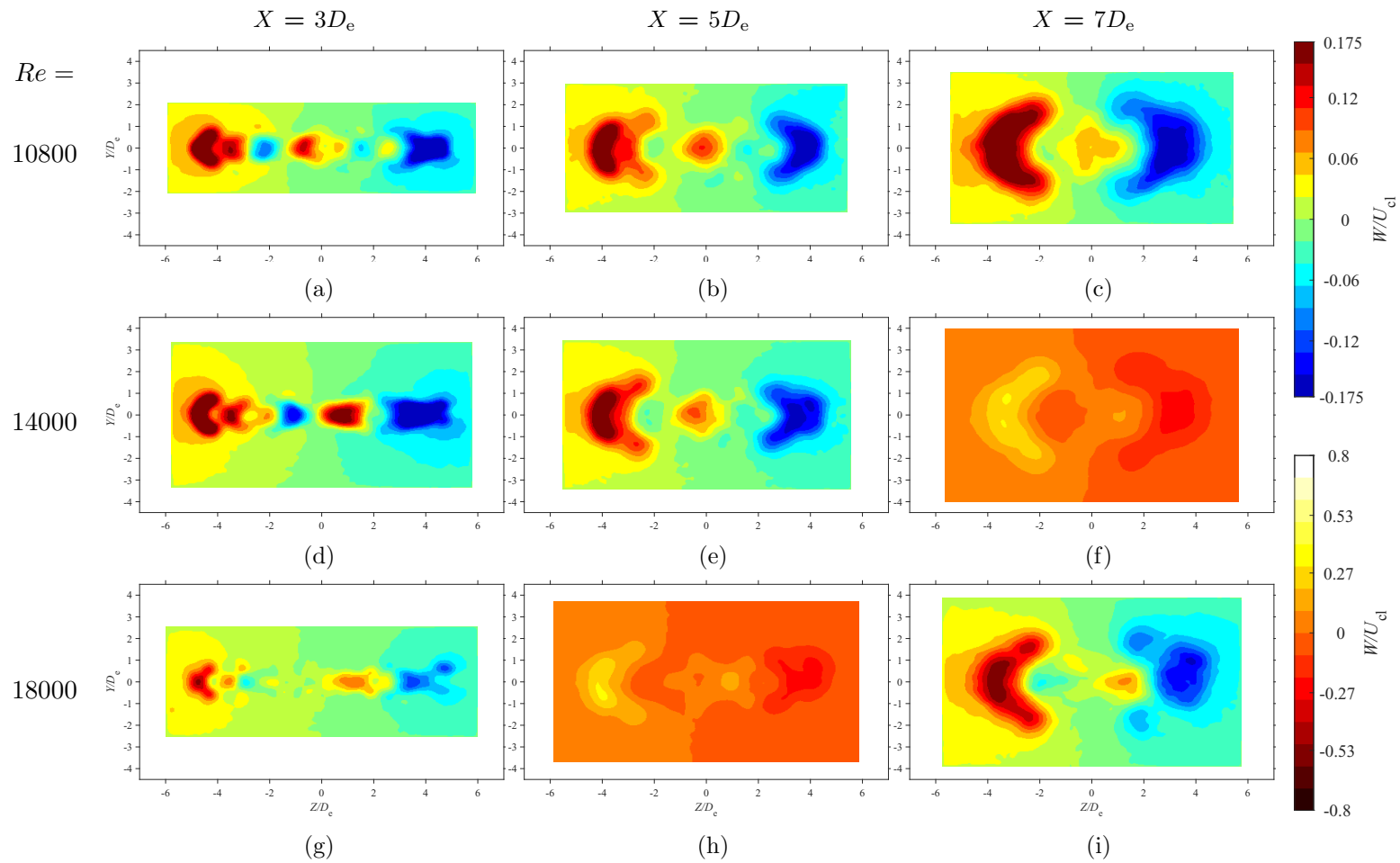


Figure 6.8: Average spanwise velocity maps at each in-plane location for $Re = 18000$, 14400 , and 18000 at $AR = 150$

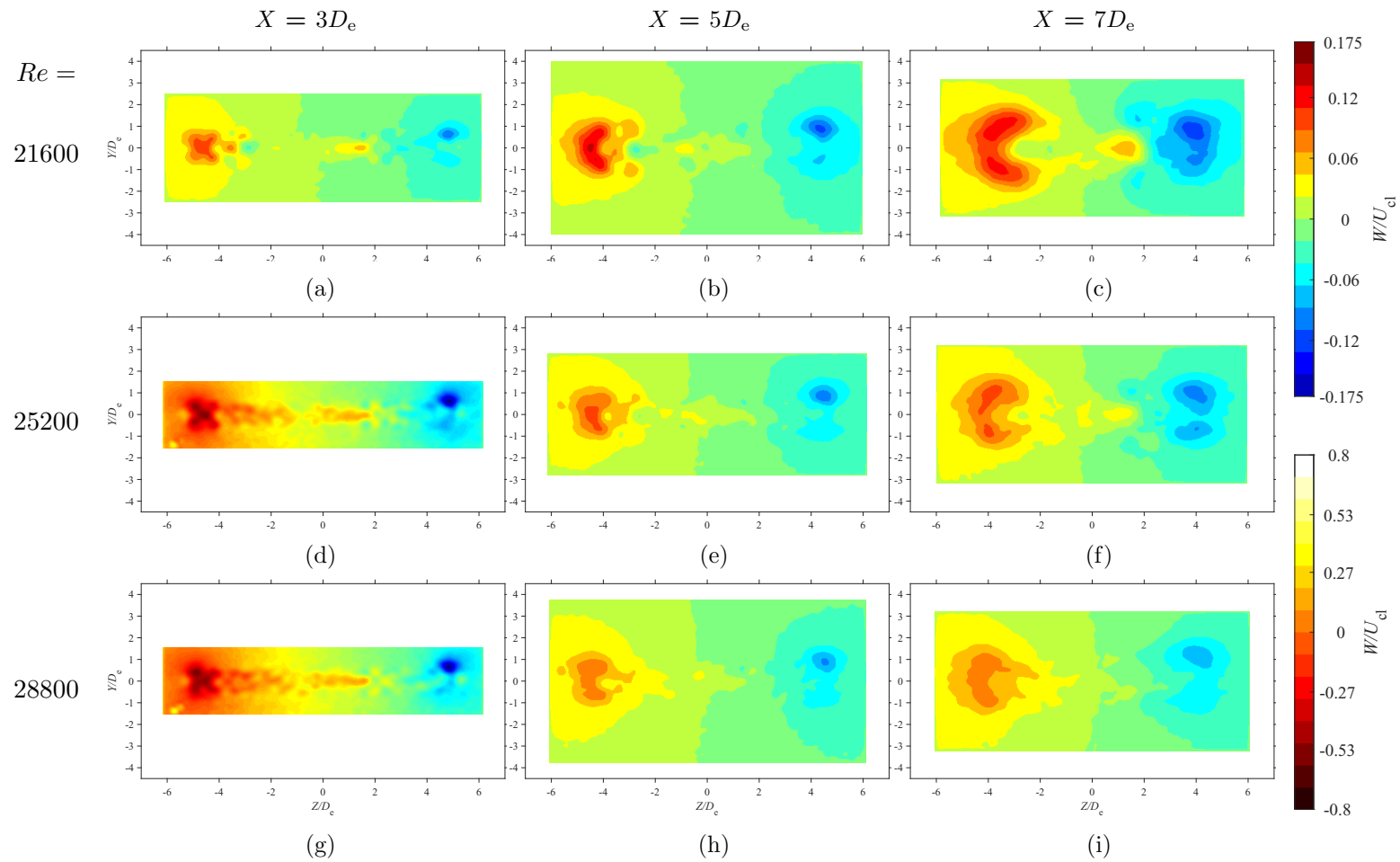


Figure 6.9: Average spanwise velocity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

6.3.2 Whole-Field Average Turbulence Intensity Measurements

The whole-field measurements that correspond to u' , v' , and w' are provided in Figures 6.10 and 6.11, 6.12 and 6.13, and 6.14 and 6.15, respectively. These figures are provided in the following six pages. Each turbulence intensity field has been nondimensionalized by U_{cl}^2 .

6.3.3 Streamwise Turbulence Intensity Measurements

The whole-field u' measurements are presented in Figures 6.10 and 6.10. By comparing the u' and U measurements, it is observed that there is good agreement between the two and their features. As expected u' displays maximum values where high velocity gradients occur. Thus, this happens around the U peaks that are observed in Figures 6.4 and 6.5. In the lateral direction, the saddle-back u' shape described by others (Krothapalli et al., 1981; Sfeir, 1979) is observed on the centerline for all configurations at $X = 3D_e$, and becomes more prominent at Re increases. The jets' saddle-back distributions are observed to subside downstream. As such a trend is expected to occur as a jet moves towards axisymmetry, it is clear that as Re increases, the length of the CDR increases. This observation is based on the fact that the saddle-back is still distinguishable at $X = 7D_e$ for $Re = 21600$, 25200, and 28800. In the spanwise direction, it is observed that the shape of u' varies for $Re = 10800$, 14000, and 18000 as a result of the U peaks that have formed. However, the shapes of u' for $Re = 21600$, 25200, and 28800 are in good agreement with those described by Sfeir (1979).

6.3.4 Lateral and Spanwise Turbulence Intensity Measurements

As before, interpreting the measurements of v' and w' proves to be difficult when observing Figures 6.12, 6.13, 6.14 and 6.15. By comparing v' and w' to their V and W counterparts, it is clear that the values of v' and w' are highest in the jet in regions where velocity gradients occur. Again, the values of both v' and w' are of similar magnitudes, as discussed by Krothapalli et al. (1981). Unlike the preceding chapter, it is noted that v' appears to instead be slightly larger in value than w' . The observation made by Krothapalli et al. (1981) that v' and w' have approximately the same values on the cross-sectional planes is not found to be true. This is not to be expected, however, as the authors' measurements were conducted in the ATDR, and this study's measurements instead took place within the CDR.

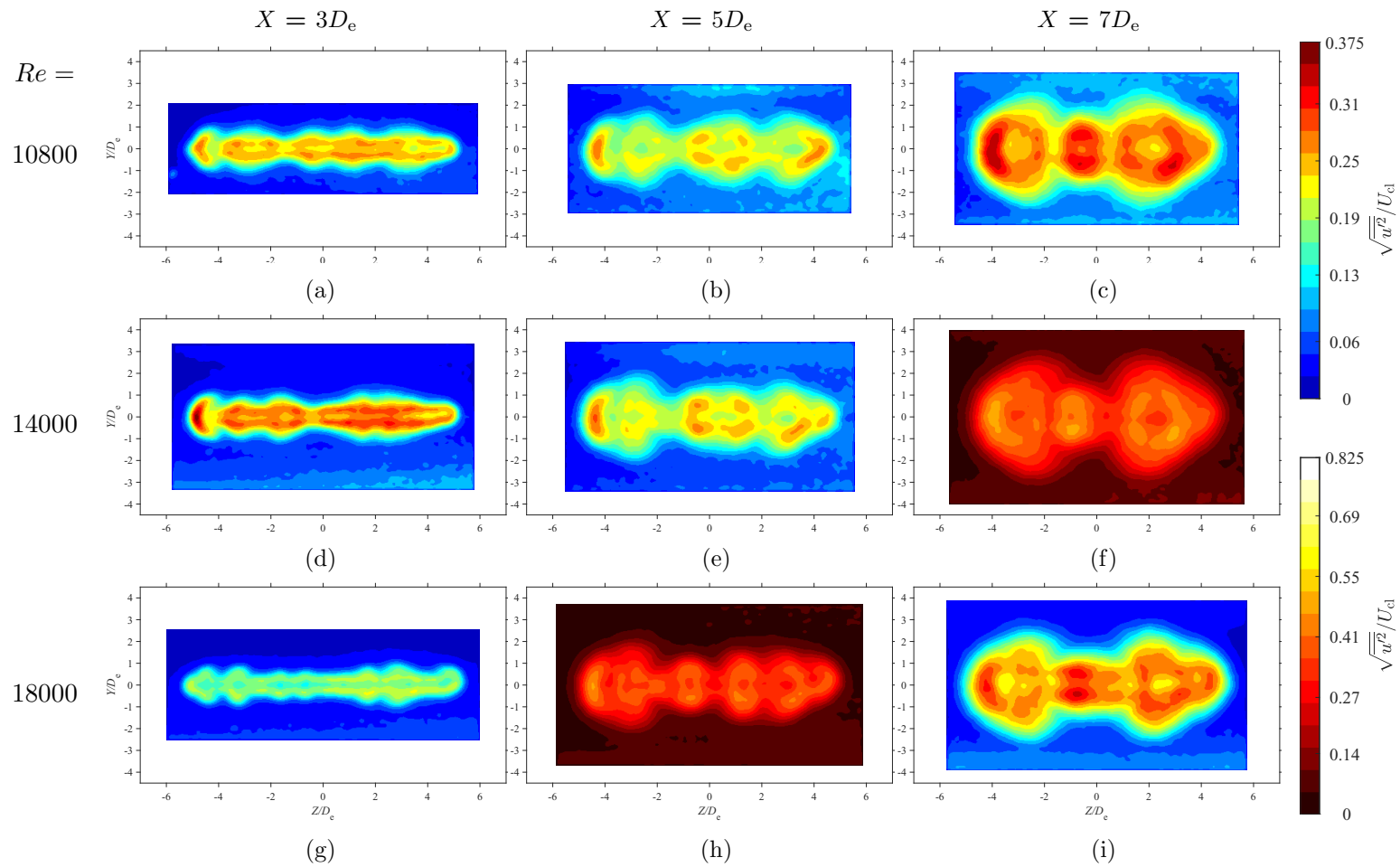


Figure 6.10: Average streamwise turbulence intensity maps at each in-plane location for $Re = 18000$, 14400 , and 10800 at $AR = 150$

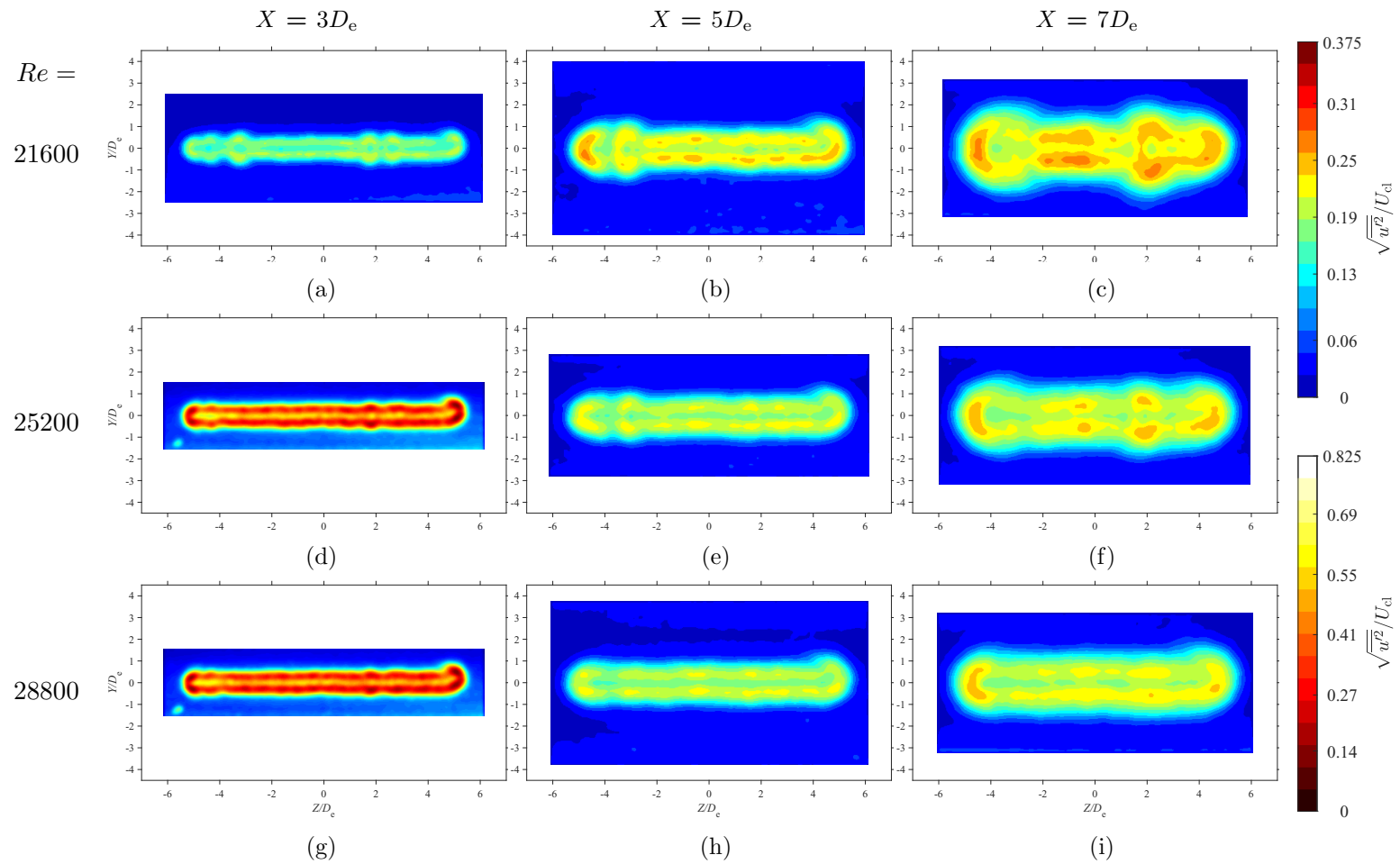


Figure 6.11: Average streamwise turbulence intensity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

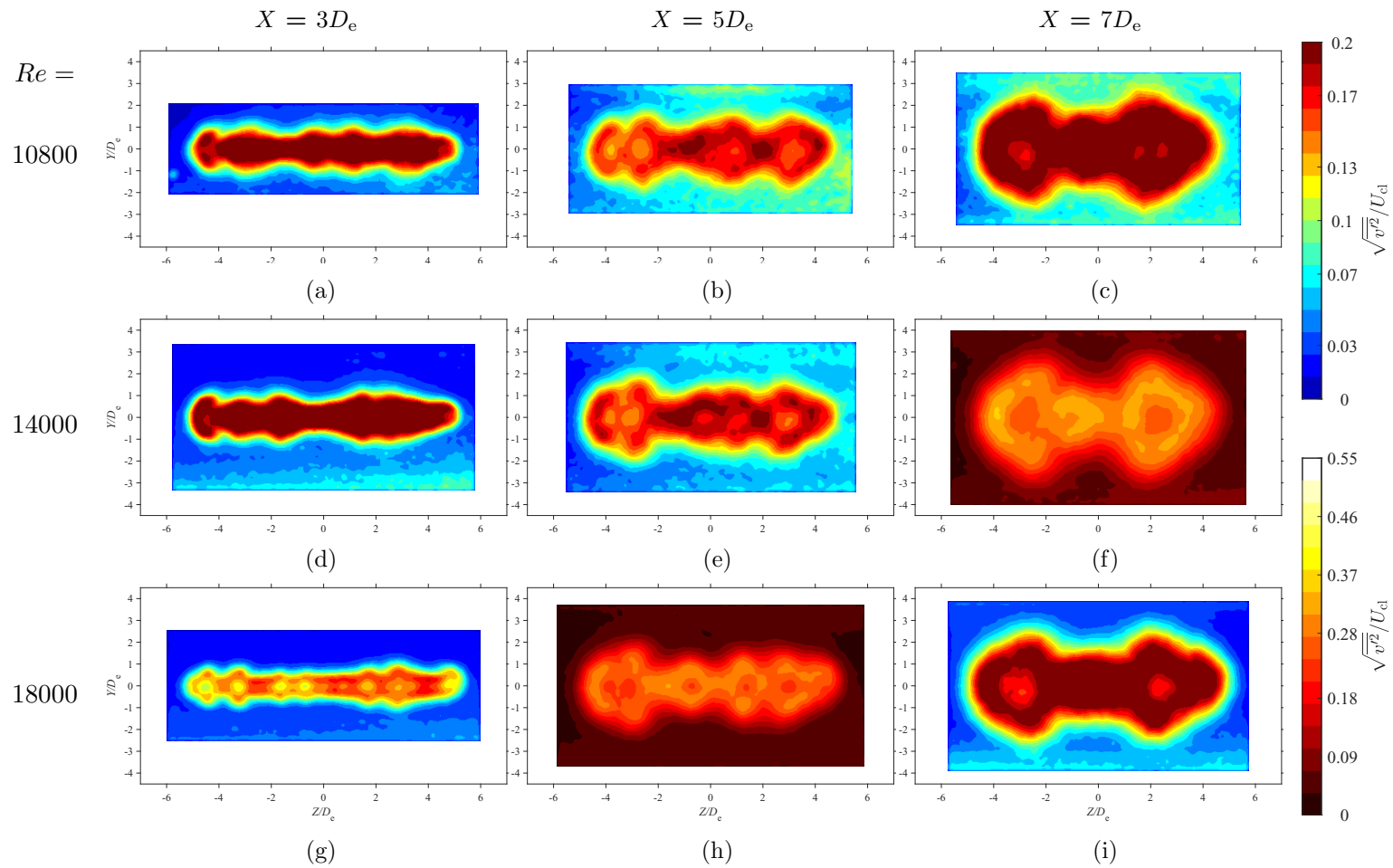


Figure 6.12: Average lateral turbulence intensity maps at each in-plane location for $Re = 18000$, 14400 , and 10800 at $AR = 150$

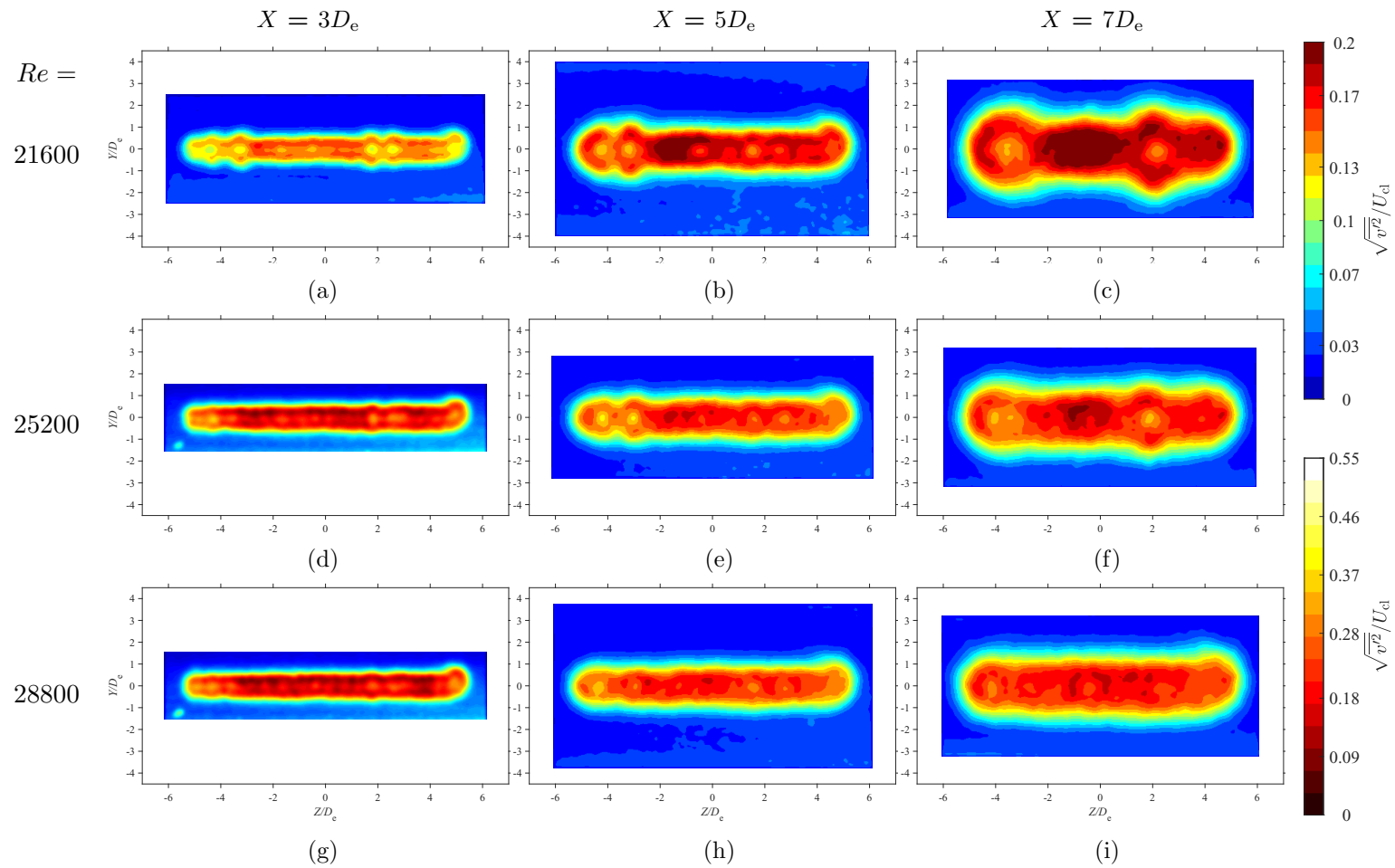


Figure 6.13: Average lateral turbulence intensity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

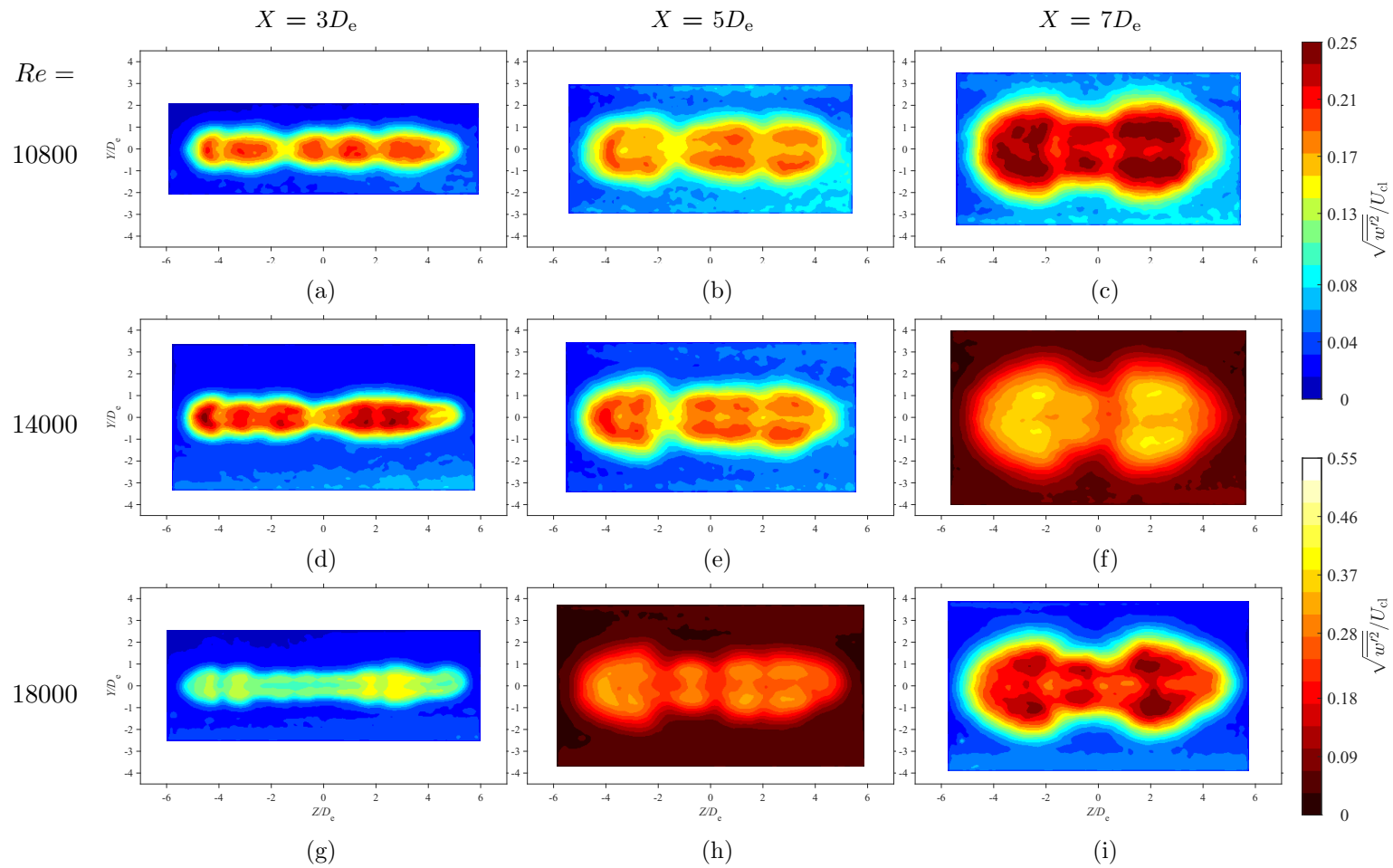


Figure 6.14: Average spanwise turbulence intensity maps at each in-plane location for $Re = 18000$, 14400 , and 10800 at $AR = 150$

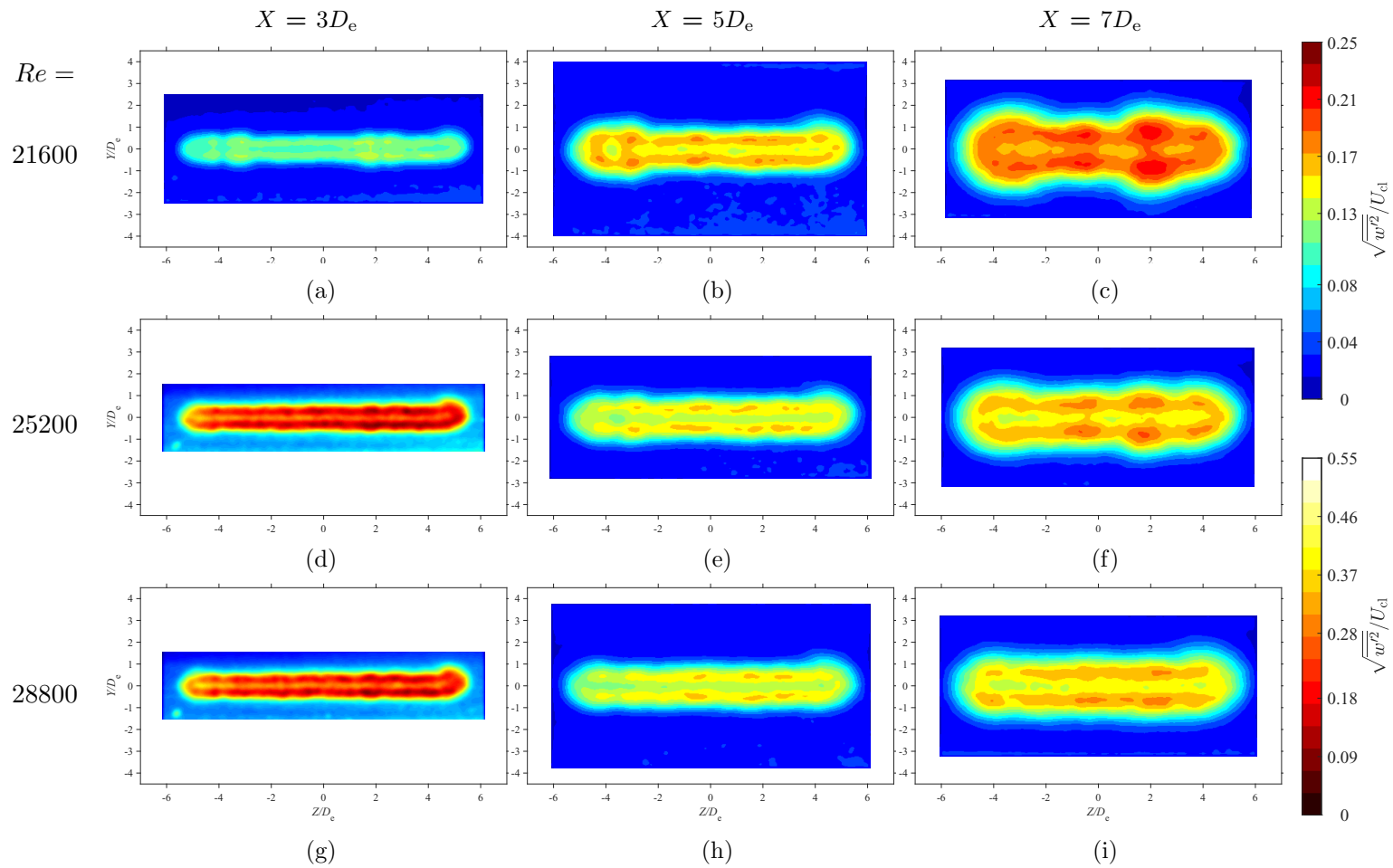


Figure 6.15: Average spanwise turbulence intensity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

6.3.5 Reynolds Shear Stresses

The whole-field measurements for $\overline{u'v'}$ are provided in Figures 6.16, and 6.17, The whole-field measurements for $\overline{u'w'}$ are provided in Figures 6.18, and 6.19. Each Reynolds shear stress field has been nondimensionalized by $U_{cl}^2 \times 100$.

6.3.6 Lateral Reynolds Shear Stress

Figures 6.16, and 6.17 provide the lateral Reynolds shear stress maps. With respect to the whole-fields for U (Figures 6.4 and 6.5) and u' (Figures 6.10 and 6.11)), the magnitude of $\overline{u'v'}$ appears to relate to where large velocity gradients occur in the lateral direction, as expected. These large gradients are most prominent near the location of U peaks, as noted by others (Lozanova & Stankov, 1998; Quinn, 1992b). No clear Re -based effects are observed for $\overline{u'v'}$.

6.3.7 Spanwise Reynolds Shear Stress

The spanwise Reynolds shear stress measurements are provided within Figures 6.18, and 6.19. Like $\overline{u'v'}$, the magnitude of $\overline{u'w'}$ appears to correlate well with where large velocity gradients occur. However, the correlation is with respect to the spanwise direction rather than the lateral direction. By comparing $\overline{u'w'}$ with U (Figures 6.4 and 6.5), it is clear that negative/positive pairs occur where U peaks are located. This observation is also in line with the works of Quinn (1992b) and Lozanova and Stankov (1998). Overall, there are no Re effects that are observed for $\overline{u'w'}$.

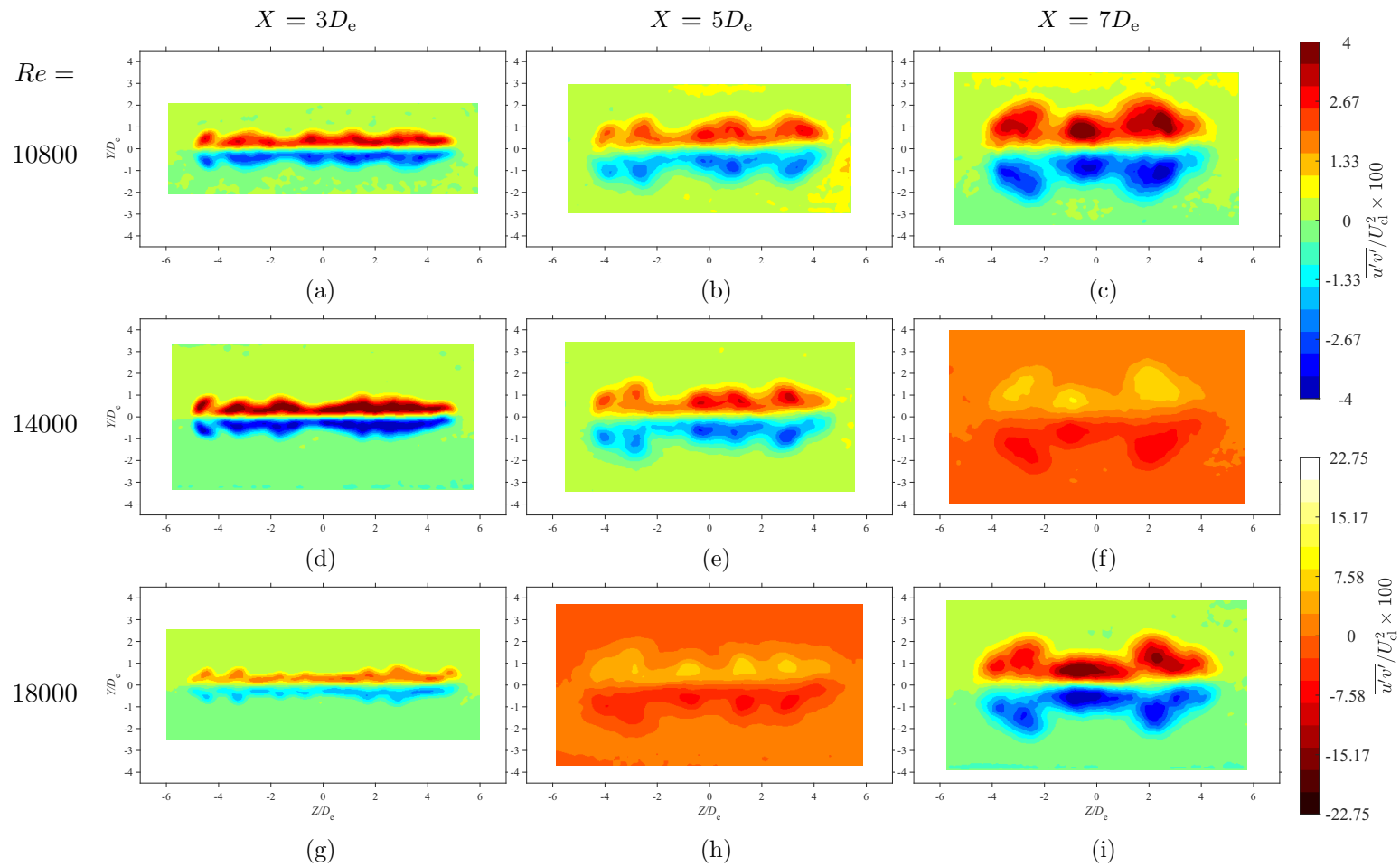


Figure 6.16: Average lateral Reynolds shear stress maps at each in-plane location for $Re = 18000$, 14400 , and 18000 at $AR = 150$

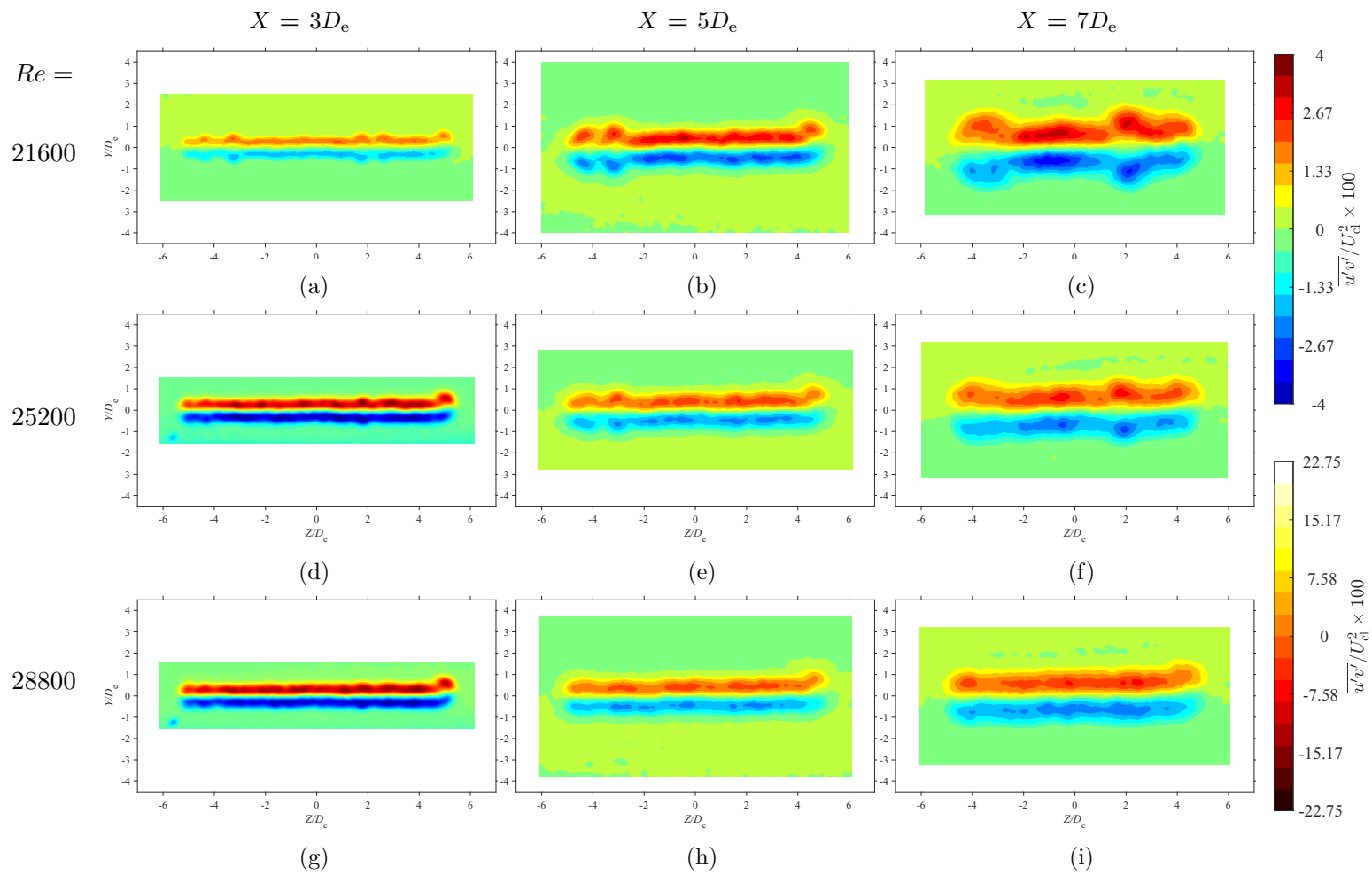


Figure 6.17: Average lateral Reynolds shear stress maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

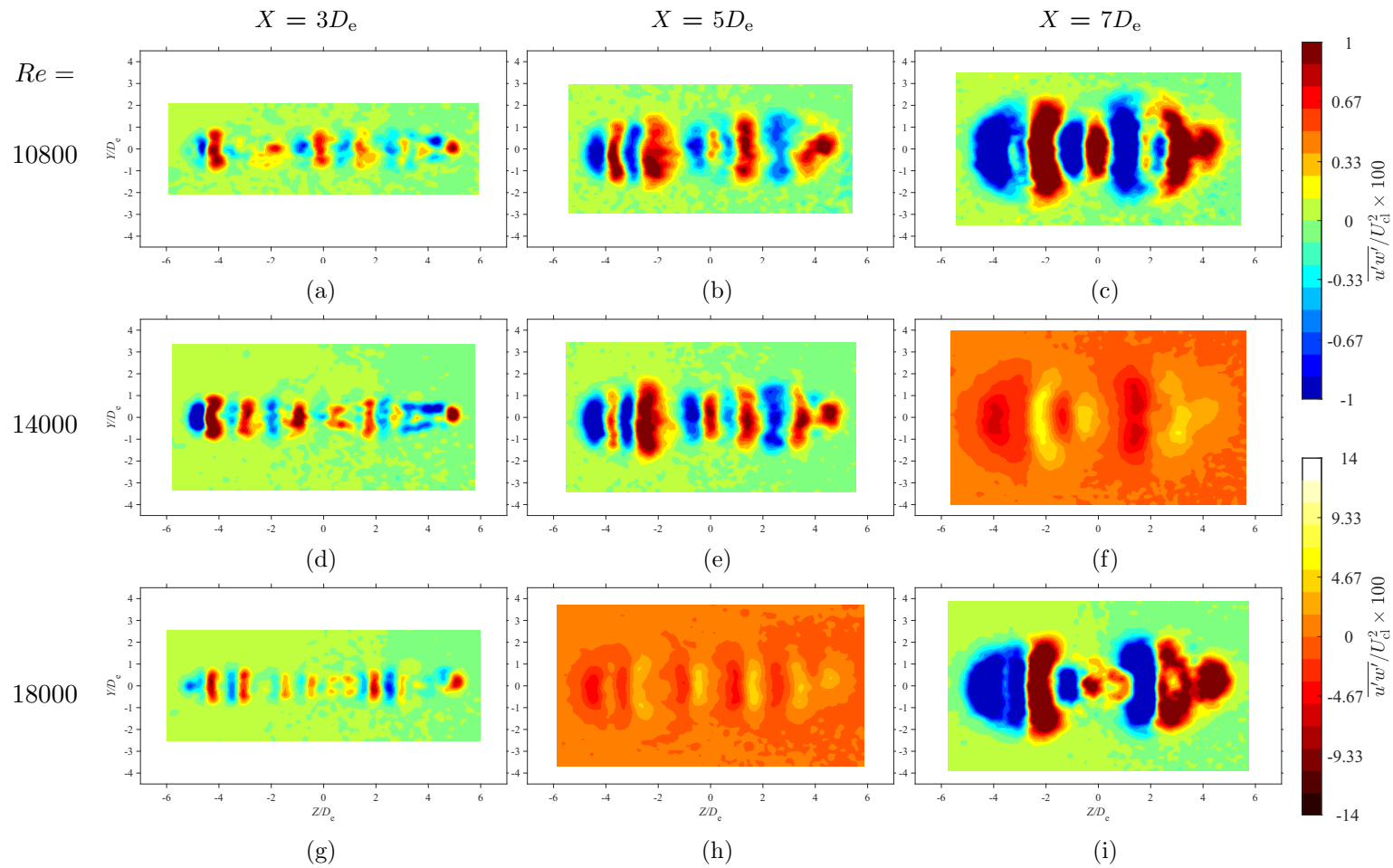


Figure 6.18: Average spanwise Reynolds shear stress maps at each in-plane location for $Re = 18000$, 14000 , and 18000 at $AR = 150$

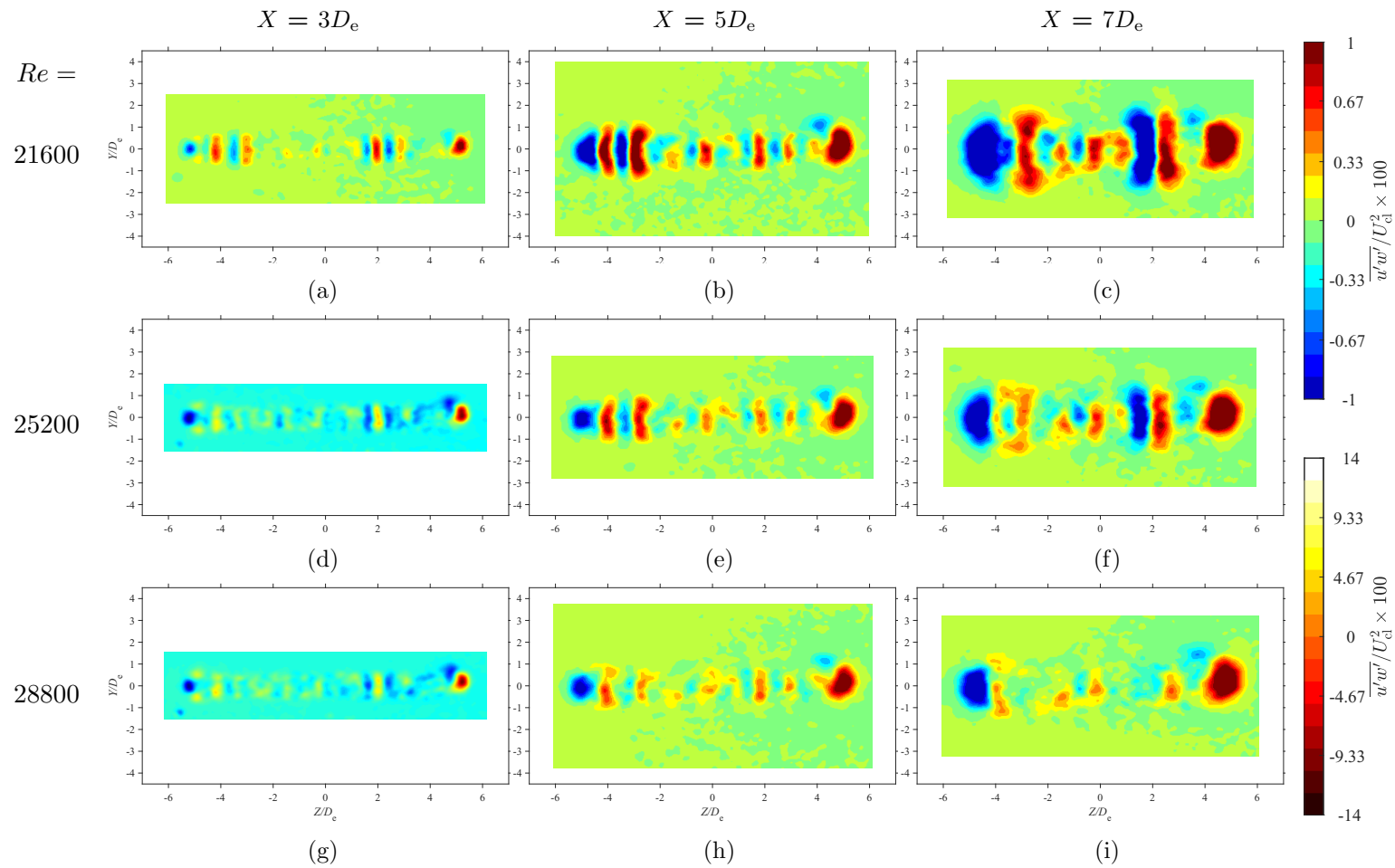


Figure 6.19: Average spanwise Reynolds shear stress maps at each in-plane location for $Re = 21600, 25200,$ and 28800 at $AR = 150$

6.3.8 Turbulent Kinetic Energy

The whole-field measurements for the turbulent kinetic energy are presented within Figures 6.20 and 6.21. Each TKE field has been nondimensionalized by $U_{cl}^2 \times 100$. With regards to $Re = 10800, 14000, \text{ and } 18000$, it is clear that there are similarities between this study's TKE fields and those of Quinn (1992b). In general, it is observed that for these Reynolds numbers, the TKE is higher within the $X-Z$ plane than the $X-Y$ in every plane. This is due to the multi-peak U profiles forming in the spanwise direction, which causes high velocity gradients that increase the production of TKE . Conversely, when $Re = 21600, 25200, \text{ and } 28800$, this is not observed to be the case. In general, the U flow fields for these Reynolds numbers are more consistent along the spanwise direction. As a result, the velocity gradients cause the production of TKE to form almost uniformly along the jet perimeter. As a result, both cross-sectional planes exhibit similar TKE values. Overall, no Re -based effects are observed for all of the configurations.

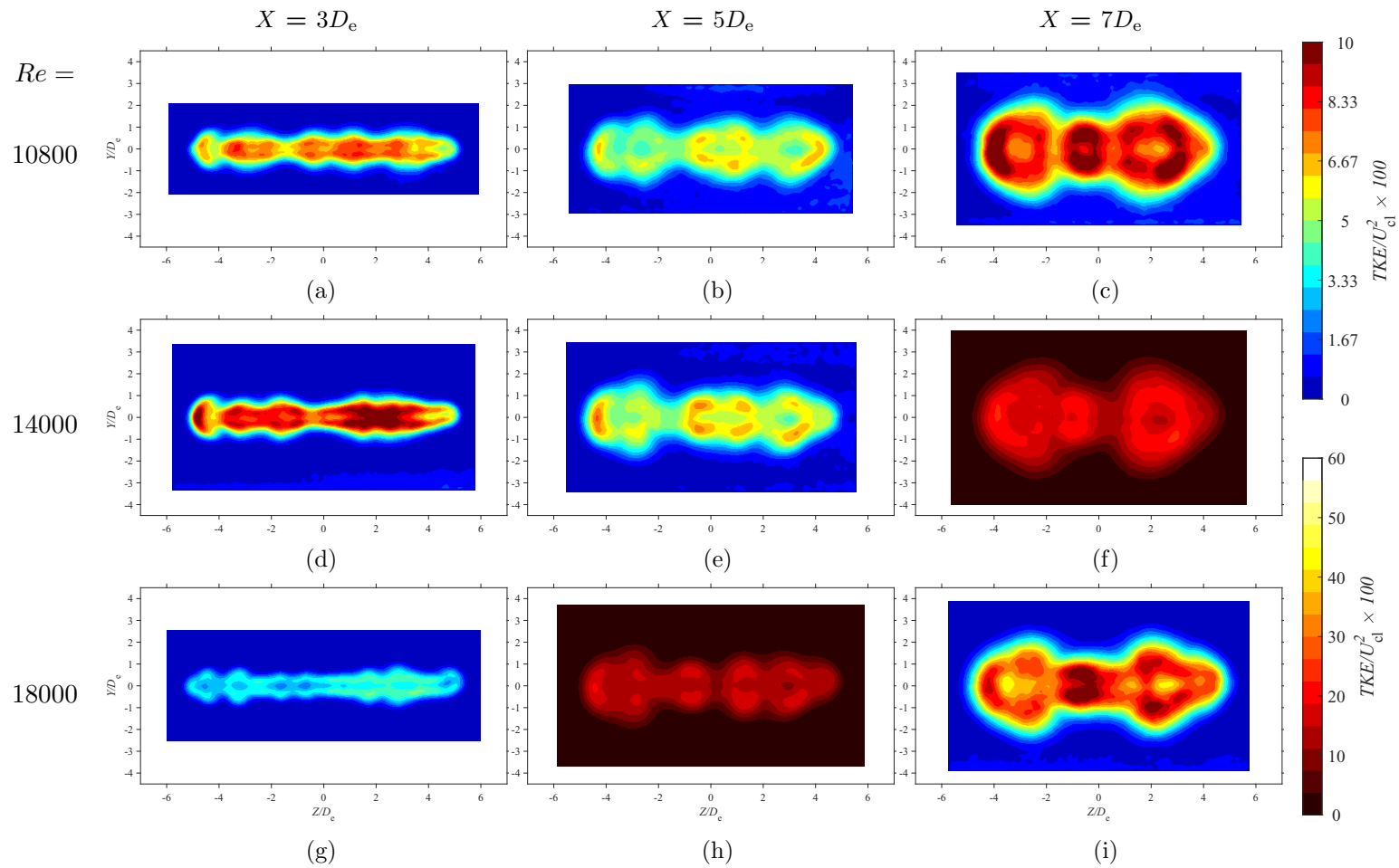


Figure 6.20: Average TKE maps at each in-plane location for $Re = 18000$, 14400 , and 18000 at $AR = 150$

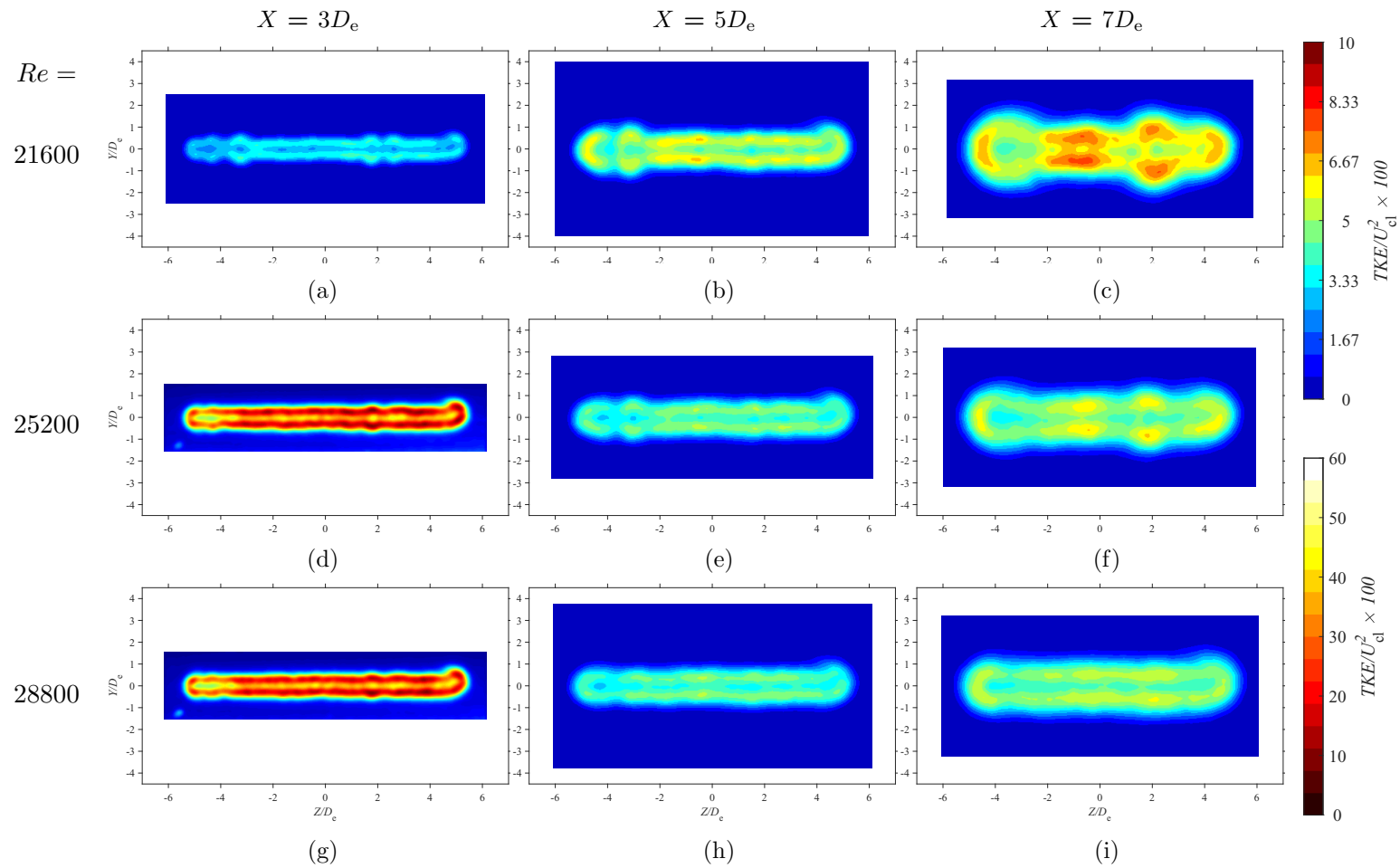


Figure 6.21: Average TKE maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

6.3.9 Streamwise Vorticity

The whole-field measurements for the streamwise vorticity, ω_x , are presented within Figures 6.22 and 6.23. Each field has been nondimensionalized by D_e/U_{cl} . Unlike AR , Re appears to affect ω_x in a linear way as it is increased. As to be expected, all Re configurations present a higher than normal amount of counter-rotating vortex pairs when compared to low- AR configurations. By looking at $X = 3D_e$ in Figures 6.22 and 6.23, it is observed that as Re is increased, the amount of vortex pairs increases, and it appears that they also become more evenly distributed as well. This trend towards an even distribution explains why the jets begin having lateral velocities profiles, with respect to U , that are consistently similar in the spanwise direction. It is also observed that as Reynolds number increases, the vortex pairs dissipate at a faster rate downstream. This is best exemplified when considering Figures 6.22c and 6.23f, which account for $Re = 10800$ and 25200 at $X = 7D_e$, respectively. When comparing the two, it is seen that $Re = 10800$ has many high-magnitude vortex pairs, whereas $Re = 25200$ has few low-magnitude vortex pairs. To explain this, it is proposed that as the evenly spaced array of vortex pairs forms, the counter-rotating vortex pairs work to cancel each other out. This cancelling out effect, along with the fully turbulent nature of the jet, results in a faster dissipation rate of the vortices. Overall, it is observed that ω_x -dynamics carry a strong Re -dependence.

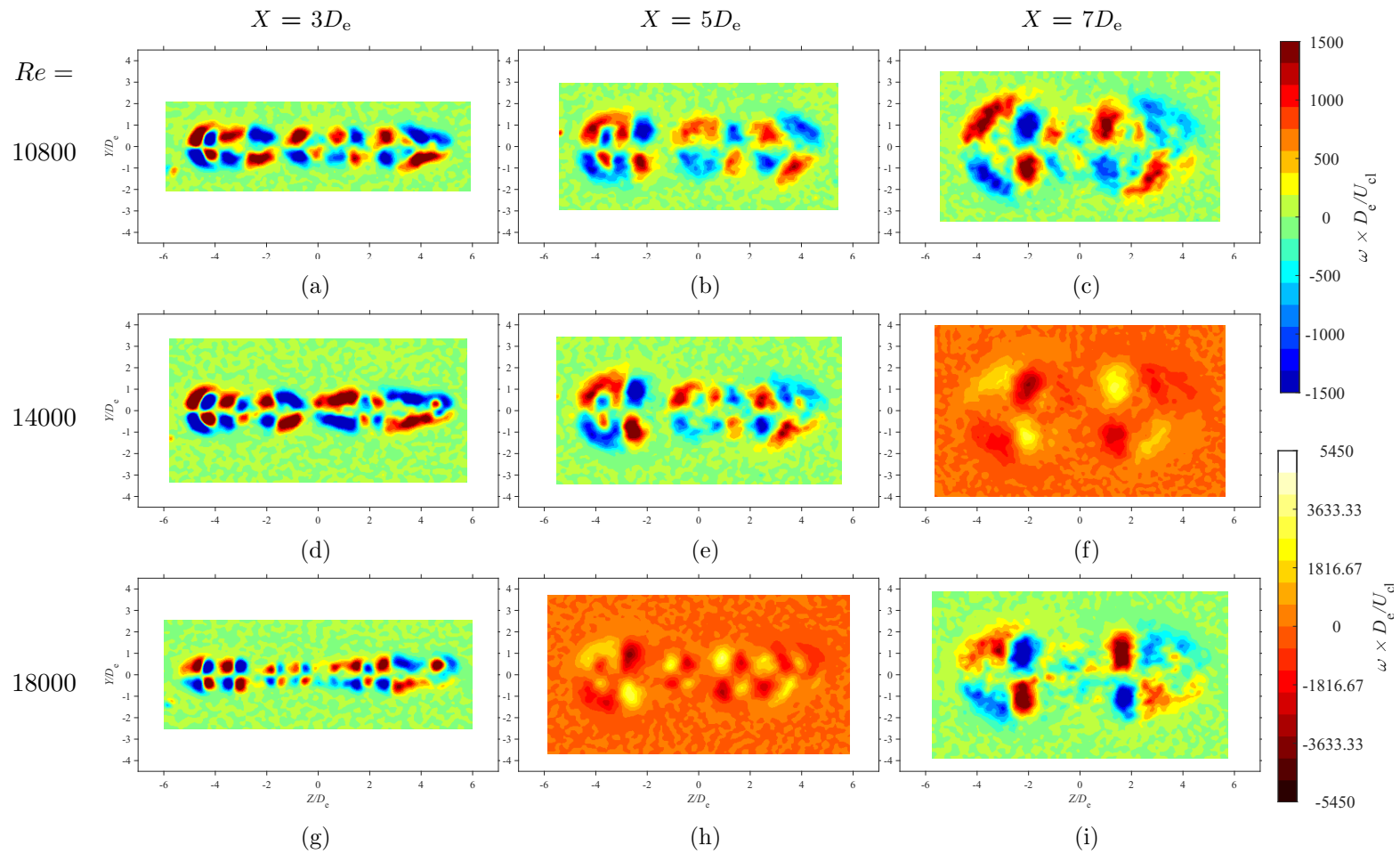


Figure 6.22: Average vorticity maps at each in-plane location for $Re = 18000$, 14400 , and 18000 at $AR = 150$

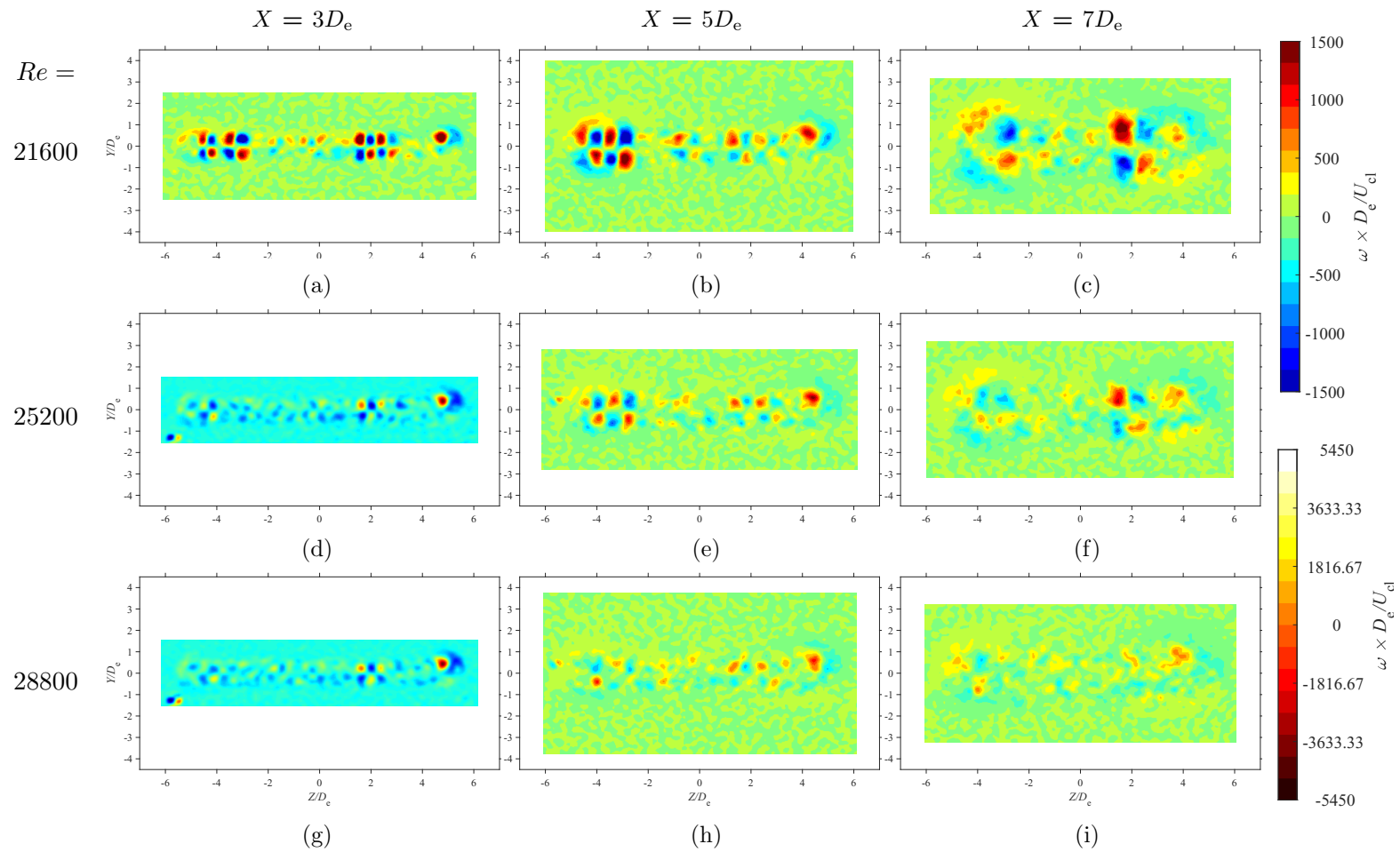


Figure 6.23: Average vorticity maps at each in-plane location for $Re = 21600$, 25200 , and 28800 at $AR = 150$

6.3.10 Cross-Sectional Boundary Deformation

The jet boundaries for all six configuration are presented in Figure 5.3.10. A Reynolds number based influence is clearly seen when observing the boundaries. This is due to the influence of Re on the ω_x -dynamics, which directly affects the jet boundary. Clearly, as Re is increased, the jet begins to take a smoother, elliptic shape. That is, the bulges as described by Vouros et al. (2015) begin to dissipate. Notably, at $X = 3D_e$, $Re = 10800$, 14000 , and 18000 , the jets exhibit the dumbbell shape described by Vouros et al. (2015). This study shows the shape to be a low- Re phenomenon, but the work of Vouros et al. was conducted at $Re_{D_e} = 85000$, which is much higher in value than the jets considered for this study. As such, these observations appear to be at odds with one another. However, this can be explained to be a product of different ω_x -dynamics for low- AR configurations.

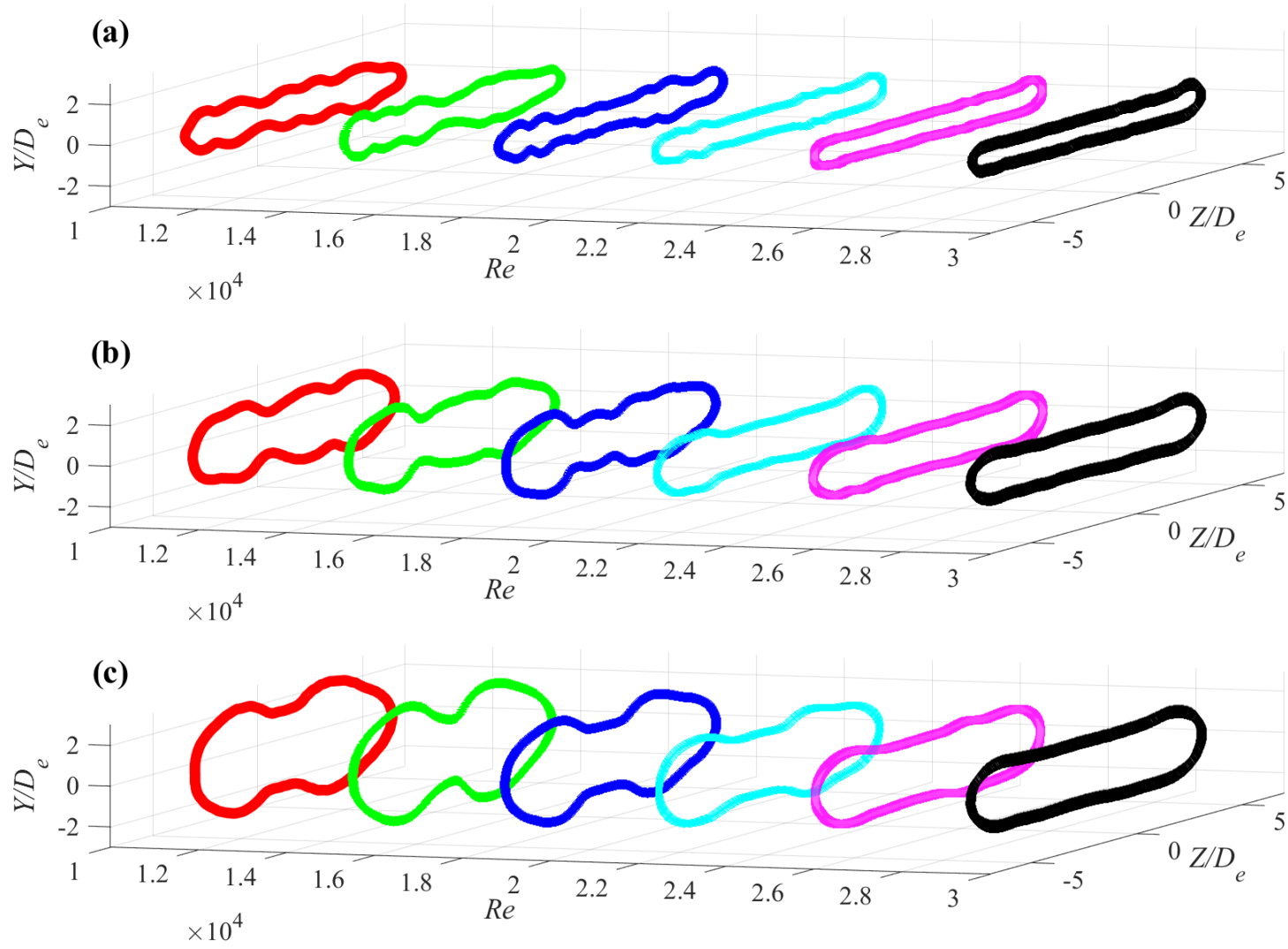


Figure 6.24: Jet boundary evolution at $AR = 150$ (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$

6.4 Conclusions

Like the previous chapter's study, streamwise and cross-sectional measurements of free, rectangular jets were taken using 2D- and stereo-PIV techniques, respectively. The PIV experiments measured the velocity field within the near-exit region, which consisted of the PCR and CDR. Within the present experimental investigation, however, the influence of Re was of interest, with specific focus placed upon values of $Re = 10800, 14000, 18000, 21600, 25200,$ and 28800 . It was of interest to apply the ideas of Dimotakis (2000) to see if a critical Re for fully turbulent flow could be observed in high- AR jets.

By considering the data from 2D-PIV measurements, some clear changes were observed with respect to Re . When velocity decay, turbulence intensity, and velocity half-width were considered, each plot showed a change in trends as Re increased. This was interpreted as the jets still transitioning towards being fully turbulent. $Re = 10800$ and 14000 were observed as being the least developed, with both displaying notable deviations in trends in comparison to the other four jets. It could not be fully determined where the critical Re was, however.

In order to help discern where exactly a fully turbulent flow was occurring, the stereo-PIV measurements were then looked upon. The average streamwise velocity measurements showed a clear trend as Re increased. The jets became more elliptic in shape, with an evenly distributed velocity as Reynolds number increased. It was found that $Re = 25200$ and 28800 presented nearly identical features, which was interpreted to mean that Re_{critical} had been passed somewhere between $Re = 21600$ and 25200 . When considering other velocity profiles, the turbulence intensities, shear stresses, and turbulent kinetic energy, few Re -dependencies were observed, and only existed when $Re = 21600, 25200,$ and 28800 were considered.

By examining the vorticity dynamics of each configuration, it was found that the distribution of the streamwise vortices had a clear Re dependency. The number of streamwise vortex pairs was found to increase with increasing Re . It was also observed that as Re increased, the vortex pairs dissipate much more quickly downstream. Notably, $Re = 21600$ and 25200 both displayed an array of evenly distributed vortex pairs. It was proposed that the evenly spaced nature of the vortex pair arrays led to the neighbouring vortices canceling each other out. The canceling out mechanism, is proposed to be the reason for why the vortex pairs dissipate more quickly downstream as Re increases. The lack of influence from the ω_x -dynamics can also explain why fully turbulent flow behave similarly to each

other. However, such a claim cannot be fully concluded. Finally, as the jet boundary is directly influenced by the vorticity dynamics, it was found that the jet boundary also had a strong Re dependence. The jet was observed to tend towards a more elliptic shape as Re increases.

Chapter 7

Conclusions and Future Work

The results of the two preceding chapters provides insight on the effects of AR and Re on free, rectangular jet flows, respectively. Both 2D- and stereo- PIV techniques were used in air to collect average flow field data. The data was processed and analyzed, and consideration was place upon the significance of the results after analysis. The conclusions drawn are fundamental towards the understanding of both SAGD systems and the work performed by other experimentalists. Further work could be conducted to expand upon the work conducted within this thesis. Thus, herein lies a summary of the conclusions made during this study, and also recommendations on further work that may be conducted.

7.1 Conclusions

7.1.1 The Effects of Aspect Ratio on a Free, Rectangular Jet

Experiments were conducted to observe the flow exiting a slot with high- AR configurations of $AR = 100, 150, 200,$ and 210 . When considering the jets' velocity decay, streamwise turbulence intensity, and lateral velocity half-width, some agreement in the behaviour of high- AR and low- AR configurations was found to exist. However, many conflicting behaviours were also found. Thus, it was concluded that an overall change in the flow conditions was occurring for the high- AR configurations. The jets' cross-sectional streamwise velocity profiles were found to vary from one another, with differing in-plane velocity distributions being observed. The saddle-back streamwise velocity profile was found to not exist along the spanwise direction. Instead, each jet displayed more U -peaks than typically found in the saddle-back profile. Further observations were made with regards to the turbulence intensities, shear stresses, and turbulent kinetic energy. From such observations, many more trends were found to be in conflict with the previous observations on low- AR configurations. When the streamwise vorticity of the jets was considered, it was found that high AR jets

do not display four streamwise vortex pairs like low- AR jets. Instead, each jet displayed a higher number of vortex pairs.

Overall, it was concluded that AR has a significant influence on the formation of the ω_x -dynamics. It was observed that the experimental setup was susceptible to the affects of small-scale disturbances, and that such disturbances also strongly influenced the formation of the ω_x -dynamics at the slot exit. It was concluded that the disturbances at the slot exit led to large asymmetries in the distribution of the ω_x -dynamics, which led to the jets displaying notable asymmetry. With these large asymmetries, it was observed that the streamwise velocity profiles along the spanwise direction vary much more than low- AR jets. From such an observation, it was concluded that the use of U_{cl} and U_{cl}^2 as a normalization parameter is improper for high- AR jets. It was concluded that the ω_x -dynamics led to the overall deviation in common trends typically observed for low- AR rectangular jets.

7.1.2 The Effects of Reynolds Number on a Free, Rectangular Jet

Experiments on the influence of Re on high- AR jets were conducted at Reynolds numbers of $Re = 10800, 14000, 18000, 21600, 25200,$ and 28800 . The velocity decay, turbulence intensity, and velocity half-width plots each showed a change in trends as Re increased. The average streamwise velocity measurements also showed a clear trend as Re increased. The jets' cross-sectional U profiles became more elliptic in shape and more evenly distributed as Re increased. When considering the velocity profiles, the turbulence intensities, shear stresses, and turbulent kinetic energy, and vorticity, it was found that $Re = 25200$ and 28800 presented nearly identical features. From this, it was concluded that $Re_{critical}$ had been passed somewhere between $Re = 21600$ and 25200 . Such a value for $Re_{critical}$ is observed to be higher than the $Re = 1 \times 10^4$ to $Re = 2 \times 10^4$ range stated to be necessary by Dimotakis (2000) to achieve fully turbulent flow. Few Re -dependencies were discernible for the other velocity profiles, the turbulence intensities, shear stresses, and turbulent kinetic energy, however. It was found that Re has an influence on the distribution of the streamwise vortices that form in the flow. The number of streamwise vortex pairs increases as Re increases. Additionally, it was observed that as the vortex pairs dissipate at a higher rate as Re increases. Notably, $Re = 21600$ and 25200 both displayed an array of evenly distributed vortex pairs. It was proposed that the uniform arrangement of the vortex pair arrays led to the neighbouring vortices canceling each other out. Though not concluded, it was proposed that the canceling mechanism of the uniform arrangement is what allows jets to behave similarly to each other at high Reynolds numbers.

7.2 Recommendations for Future Work

The data collected from the experiments, and the conclusions reached from the data have expanded upon the research previously completed by other experimentalists. The work completed has helped provide a fundamental understanding of the fluid flows of interest, and allows for both continued research in the work performed and new research into more complex problems. In terms of complex flow scenarios, the following list provides recommendations of experiments that may be conducted:

- The use of slotted liners entails the use of multiple slots for sand control. Analyses of the effects of multiple slots, and the influence of center-to-center distance between slots would prove to advantageous to research. With regards to center-to-center distance between slots, it would be of interest to observe both the effects of distance between the spanwise edges, and also the effects of distance between the lateral edges.
- The existence of oil sands around the injection and production wells leads to flows that are more complex than those examined in the present study. The development of a methodology to create a porous media that simulates the oil sands would be advantageous as it would allow for a deeper understanding of the complex flows. A parametric study would include the effects of porosity, and also the effects of distance from nozzle exit to porous media entrance. Such a study would stand to form a bridge between the studies of free jets and impinging jets, where there is 100% porosity and 0% porosity, respectively.
- Due to the difficulty in creating steam and maintaining both its quality and temperature during experimentation, air was instead used to simulate the flow at the injection well. The ability to use steam would provide results that more closely mimic the fluid mechanics found at the injection well.
- The encompassing work of this thesis was done to simulate the geometry of the straight slot. Moving forward, by developing a methodology to create test pieces that have the keystone and rolled top profiles, further work could be done to understand the effects of such geometry profiles.
- The result provided presented the observations made through 2D- and stereo-PIV techniques. It is quite clear that flows examined were quite three-dimensional in nature. In order to gain a complete understanding of the three-dimensional structures

that develop in the jets, it would be advantageous to utilize tomographic PIV (tomo-PIV) techniques. Furthermore, temporal information could also be collected through the use of high-speed tomo-PIV techniques.

- As demonstrated by the preceding discussions, the work completed on jets has placed emphasis on the study of the exiting flow. Comparatively, the study of the entering flow has not been approached as rigorously by experimentalists. To date, no work observing the flow field of a fluid entering a narrow slot has been conducted. The likely reason for why there is a lack of experimental investigations on such flow conditions is the lack of applications for such a configuration. When considering the flow near the production well of the SAGD process, it is clear that such a study is necessary. Thus, the need for understanding such flows allows for a novel area in jet research to be explored.
- As injected steam penetrates the surrounding oil sands, multi-phase flows comprising of steam, water, bitumen, and sand fines are created during the SAGD process. The multi-phase flow enters the production well. Thus, the ability to make a working fluid that is similar to the multi-phase SAGD fluids, and that can be used for PIV-based experiments would further develop the examination of converging jet flows.

Works Cited

- Adrian, R. J. (2005). Twenty years of particle image velocimetry. *Experiments in fluids*, 39(2), 159-169.
- Ahmed, T., & Meehan, D. N. (2012). Chapter 6 - introduction to enhanced oil recovery. In T. Ahmed & D. N. Meehan (Eds.), *Advanced reservoir management and engineering (second edition)* (Second Edition ed., p. 541 - 585). Boston: Gulf Professional Publishing.
- Alberta Energy. (2017). *Oil sands facts and statistics*. Retrieved February 21, 2018, from <https://www.alberta.ca/oil-sands-facts-and-statistics.aspx>
- Alberta Government. (2017). *Steam assisted gravity drainage - facts and stats*. Retrieved March 27, 2018, from <https://open.alberta.ca/dataset/f7c779ea-9776-4d59-a1b3-178d533f0ebc/resource/aac73d46-3ae0-478e-9b19-e64afa41b1f2/download/fssagd.pdf>
- Barber, J., Saediamiri, M., & Birouk, M. (2017). Influence of nozzle exit conditions on the near-field mean flow characteristics of low-aspect-ratio rectangular jets. *International Journal of Fluid Mechanics Research*, 44(4), 297-319.
- Bell, J. H., & Mehta, R. D. (1989). Design and calibration of the mixing layer and wind tunnel. *NASA-CR-185472*.
- Bennion, D. B., Gupta, S., Gittins, S., & Hollies, D. (2009). Protocols for slotted liner design for optimum SAGD operation. *Journal of Canadian Petroleum Technology*, 48(11), 21-26.
- Blevins, R. D. (1984). *Applied fluid dynamics handbook*. New York, NY: Van Nostrand Reinhold Co.
- Butler, R. M. (1994). Steam-assisted gravity drainage: concept, development, performance and future. *Journal of Canadian Petroleum Technology*, 33, 455-460.
- Butler, R. M., McNab, G. S., & Lo, H. Y. (1981). Theoretical studies on the gravity drainage of heavy oil during in-situ steam heating. *The Canadian Journal of Chemical Engineering*, 59, 455-460.
- Butler, R. M., & Stephens, D. J. (1981). The gravity drainage of steam-heated heavy oil to parallel horizontal wells. *Journal of Canadian Petroleum Technology*, 20, 90-96.
- CAPP. (2018). *Canada's oil sands fact book* (Booklet). Retrieved from <https://www.capp.ca/publications-and-statistics/publications/316441>
- Carlson, J., Gurley, D., King, G., Price-Smith, C., & Waters, F. (1992). Sand control: Why and how? *Oilfield Review*, 4(4), 41-53.
- Chalaturnyk, R. J., Wagg, B. T., & Dusseault, M. B. (1992). The mechanisms of solids production in unconsolidated heavy-oil reservoirs. In *SPE formation damage control symposium*. Lafayette, LA.
- Coles, D., Cantnell, B., & Wadcock, A. (1978). The flying hot wire and related instrumentation. *NASA-CR-3066*.

- Das, S. K., & Butler, R. M. (1998). Mechanism of the vapor extraction process for heavy oil and bitumen. *Journal of Petroleum Science and Engineering*, *21*, 43-59.
- Deo, R. C. (2005). *Experimental investigations of the influence of reynolds number and boundary conditions on a plane air jet* (Unpublished doctoral dissertation). University of Adelaide.
- Deo, R. C., Nathan, G. J., & Mi, J. (2010). On the influence of initial conditions on a turbulent plane jet. In *17th australasian fluid mechanics conference*. Auckland, New Zealand.
- der Hegge Zijnen, B. G. V. (1958). Measurements of the velocity distribution in a plane turbulent jet of air. *Annual Review of Fluid Mechanics*, *7*, 256-257.
- Dimotakis, P. E. (2000). The mixing transition in turbulent flows. *Journal of Fluid Mechanics*, *409*, 69-98.
- Greaves, M., Saghr, A. M., Xia, T., Turtar, A., & Ayasse, C. (2001). THAI – new air injection technology for heavy oil recovery and in situ upgrading. *Journal of Canadian Petroleum Technology*, *40*, 38-47.
- Gutmark, E., & Grinstein, F. (1999). Flow control with noncircular jets. *Annual Review of Fluid Mechanics*, *31*, 239-272.
- Gutmark, E., Schadow, K. C., Parr, T. P., Hanson-Parr, D., & Wilson, K. (1989). Noncircular jets in combustion systems. *Experiments in Fluids*, *7*(4), 248-258.
- Haan, H. J. D., & Lookeren, J. V. (1969). Early results of the first large-scale steam soak project in the Tia Juana field, western Venezuela. *Journal of Petroleum Technology*, *21*, 101-110.
- Hassan, M. E., & Meslem, A. (2010). Time-resolved stereoscopic particle image velocimetry investigation of the entrainment in the near field of circular and daisy-shaped orifice jets. *Physics of Fluids*, *22*(3).
- Hussain, F., & Husain, H. S. (1989). Elliptic jets. Part 1. Characteristics of unexcited and excited jets. *Journal of Fluid Mechanics*, *208*, 257-320.
- Kaiser, T. M. V., Wilson, S., & Venning, L. A. (2002). Inflow analysis and optimization of slotted liners. *SPE Drilling and Completion*, *17*(4), 200-209.
- Krothapalli, A., Baganoff, D., & Karamcheti, K. (1981). On the mixing of a rectangular jet. *Journal of Fluid Mechanics*, *107*, 201-220.
- Li, L., Ma, Y., Mahmoudi, M., Fattahpour, V., & Lange, C. F. (2018). Steps toward designing the optimum outflow control device for SAGD using computational fluid dynamics simulation. In *SPE canada heavy oil technical conference*. Calgary, AB.
- Lozanova, M., & Stankov, P. (1998). Experimental investigation on the similarity of a 3D rectangular turbulent jet. *Experiments in Fluids*, *24*(5), 470-478.
- Mahmud, H. B., Leong, V. H., & Lestariono, Y. (2019). Sand production: A smart control framework for risk mitigation. *Petroleum*.
- Malmström, T. G., Kirkpatrick, A. T., Christensen, B., & Knappmiller, K. D. (1997). Centreline velocity decay measurements in low-velocity axisymmetric jets. *Journal of Fluid Mechanics*, *346*, 363-377.
- Marsters, G. F., & Fotheringham, J. (1980). The influence of aspect ratio on incompressible, turbulent flows from rectangular slots. *Aeronautical Quarterly*, *31*(4), 285-305.
- Martin, J. C. (1967). A theoretical analysis of steam stimulation. *Journal of Petroleum Technology*, *19*, 411-418.
- Matanović, D., Čikeš, M., & Moslavac, B. (2012). Sand control methods. In *Sand control in well construction and operation* (p. 29-83). Berlin, Heidelberg: Springer Berlin Heidelberg.

- Melling, A. (1997). Tracer particles and seeding for particle image velocimetry. *Measurement Science and Technology*, 8(12), 1406-1416.
- Meslem, A., Nastase, I., & Allard, F. (2010). Passive mixing control for innovative air diffusion terminal devices for buildings. *Building and Environment*, 45(12), 2679-2688.
- Mi, J., Deo, R. C., & Nathan, G. J. (2005). Characterization of turbulent jets from high-aspect-ratio rectangular nozzles. *Physics of Fluids*, 17(6).
- Mi, J., Kalt, P., Nathan, G. J., & Wong, C. Y. (2007). PIV measurements of a turbulent jet issuing from round sharp-edged plate. *Experiments in fluids*, 42(4), 625-637.
- Mi, J., Nathan, G. J., & Luxton, R. E. (2000). Centreline mixing characteristics of jets from nine differently shaped nozzles. *Experiments in Fluids*, 28(1), 93-94.
- Nathan, G. J., Mi, J., Alwahabi, Z. T., Newbold, G. J. R., & Nobes, D. S. (2006). Impacts of a jet's exit flow pattern on mixing and combustion performance. *Progress in Energy and Combustion Science*, 32(5), 496-538.
- Natural Resources Canada. (2017). *Oil resources*. Retrieved February 21, 2018, from <https://www.nrcan.gc.ca/energy/energy-sources-distribution/crude-oil/oil-resources/18085>
- Oertel, H. (2010). Dynamics of fluid flow. In H. Oertel (Ed.), *Prandtl - Essentials of fluid mechanics* (p. 59-263). New York, NY: Springer New York.
- Ott, W. K., & Woods, J. D. (2003). *Modern sandface completion practices handbook*. Houston, TX: Gulf Publishing Company.
- Prasad, A. K. (2000). Particle image velocimetry. *Current Science-Bangalore-*, 79(1), 51-60.
- Quinn, W. R. (1992a). Streamwise evolution of a square jet cross section. *AIAA Journal*, 30(12), 2852-2857.
- Quinn, W. R. (1992b). Turbulent free jet flows issuing from sharp-edged rectangular slots: The influence of slot aspect ratio. *Experimental Thermal and Fluid Science*, 5(2), 203-215.
- Quinn, W. R. (1994). Development of a large-aspect-ratio rectangular turbulent free jet. *AIAA Journal*, 32(3), 547-554.
- Quinn, W. R., Pollard, A., & Marsters, G. F. (1983). On 'saddle-backed' velocity distributions in a three-dimensional turbulent free jet. In *AIAA 16th fluid and plasma dynamics conference*. Danvers, MA.
- Raffel, M., Willert, C. E., Scarano, F., Kähler, C., Wereley, S. T., & Kompenhans, J. (2018). *Particle image velocimetry*. Cham, Switzerland: Springer International Publishing.
- Rajaratnam, N. (1976). *Turbulent jets*. New York, NY: Elsevier.
- RAMP. (2008). *In situ methods used in the oil sands*. Retrieved March 27, 2018, from <http://www.ramp-alberta.org/resources/development/history/insitu.aspx>
- Renpu, W. (2011). *Advanced well completion engineering*. Gulf Professional Publishing.
- Schlichting, H., & Gersten, K. (2017). *Boundary-layer theory*. Berlin, Heidelberg: Springer.
- Sciacchitano, A. (2019). Uncertainty quantification in particle image velocimetry. *Measurement Science and Technology*, 30(9).
- Sciacchitano, A., & Wieneke, B. (2016). PIV uncertainty propagation. *Measurement Science and Technology*, 27(8).
- Sfeir, A. A. (1979). Investigation of three-dimensional turbulent rectangular jets. *AIAA Journal*, 17(10), 1055-1060.
- Sforza, P. M., Steiger, M. H., & Trentacoste, N. (1966). Studies on three-dimensional viscous jets. *AIAA Journal*, 4(5), 800-806.

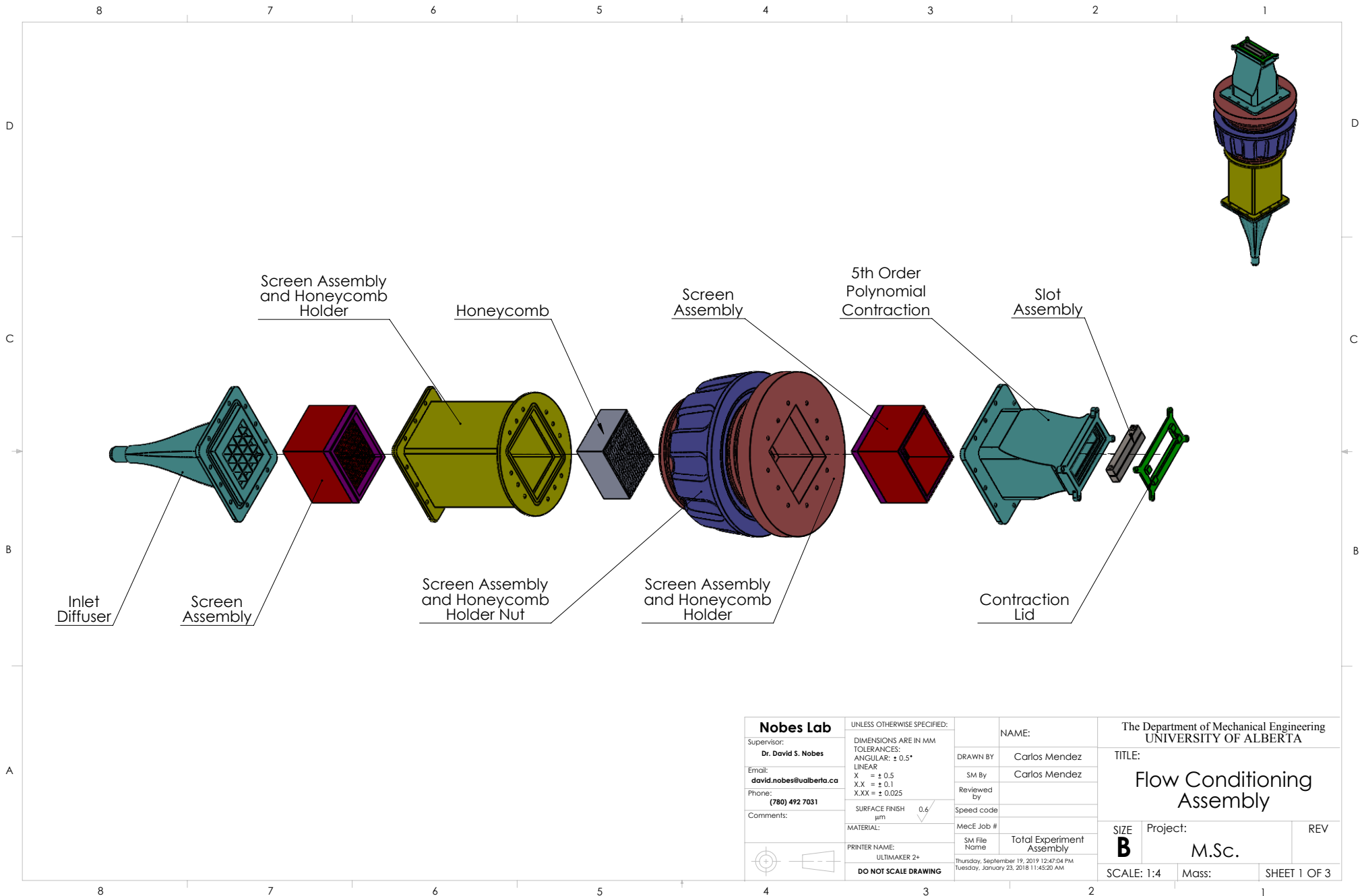
- Sforza, P. M., & Trentacoste, N. (1967). Further experimental results for three-dimensional free jets. *AIAA Journal*, 5(5), 885-891.
- Soloff, S. M., Adrian, R. J., & Liu, Z.-C. (1997). Distortion compensation for generalized stereoscopic particle image velocimetry. *Measurement science and technology*, 8(12), 1441-1454.
- Tesař, V. (2015). Impinging jets. In D. T. H. New & S. C. M. Yu (Eds.), *Vortex rings and jets: Recent developments in near-field dynamics* (p. 191-231). Singapore: Springer Singapore.
- Thompson, B. E., & Whitelaw, J. H. (1984). Flying hot-wire anemometry. *Experiments in Fluids*, 2(1), 47-55.
- Tropea, C., Yarin, A. L., & Foss, J. F. (2007). *Springer handbook of experimental fluid mechanics*. Berlin, Heidelberg: Springer.
- Tsuchiya, Y., & Horikoshi, C. (1986). On the spread of rectangular jets. *Experiments in Fluids*, 4(4), 197-204.
- Viskanta, R. (1993). Heat transfer to impinging isothermal gas and flame jets. *Experimental Thermal and Fluid Science*, 6(2), 111-134.
- Vouros, A. P., Panidis, T., Pollard, A., & Schwab, R. R. (2015). Near field vorticity distributions from a sharp-edged rectangular jet. *International Journal of Heat and Fluid Flow*, 51, 383-394.
- Watmuff, J. H., Perry, A. E., & Chong, M. S. (1983). A flying hot-wire system. *Experiments in Fluids*, 1(2), 63-71.
- Wieneke, B. (2005). Stereo-PIV using self-calibration on particle images. *Experiments in fluids*, 39(2), 267-280.
- Willman, B. T., Valleroy, V. V., Runberg, G. W., Cornelius, A. J., & Powers, L. W. (1961). Laboratory studies of oil recovery by steam injection. *Journal of Petroleum Technology*, 13, 681-690.
- Xie, J., Jones, S. W., Matthews, C. M., Wagg, B. T., Parker, P., & Ducharme, R. (2007). Slotted liner design for SAGD wells. *World Oil*, 228(6), 67-75.
- Xu, G., & Antonia, R. (2002). Effect of different initial conditions on a turbulent round free jet. *Experiments in Fluids*, 33(5), 677-683.
- Yevjevich, V. M. (1965). *Diffusion of slot jets with finite orifice length-width ratios*. Fort Collins, CO: Colorado State University.
- Zaman, K. B. M. Q. (1996). Axis switching and spreading of an asymmetric jet: The role of coherent structure dynamics. *Journal of Fluid Mechanics*, 316, 1-27.

Appendices

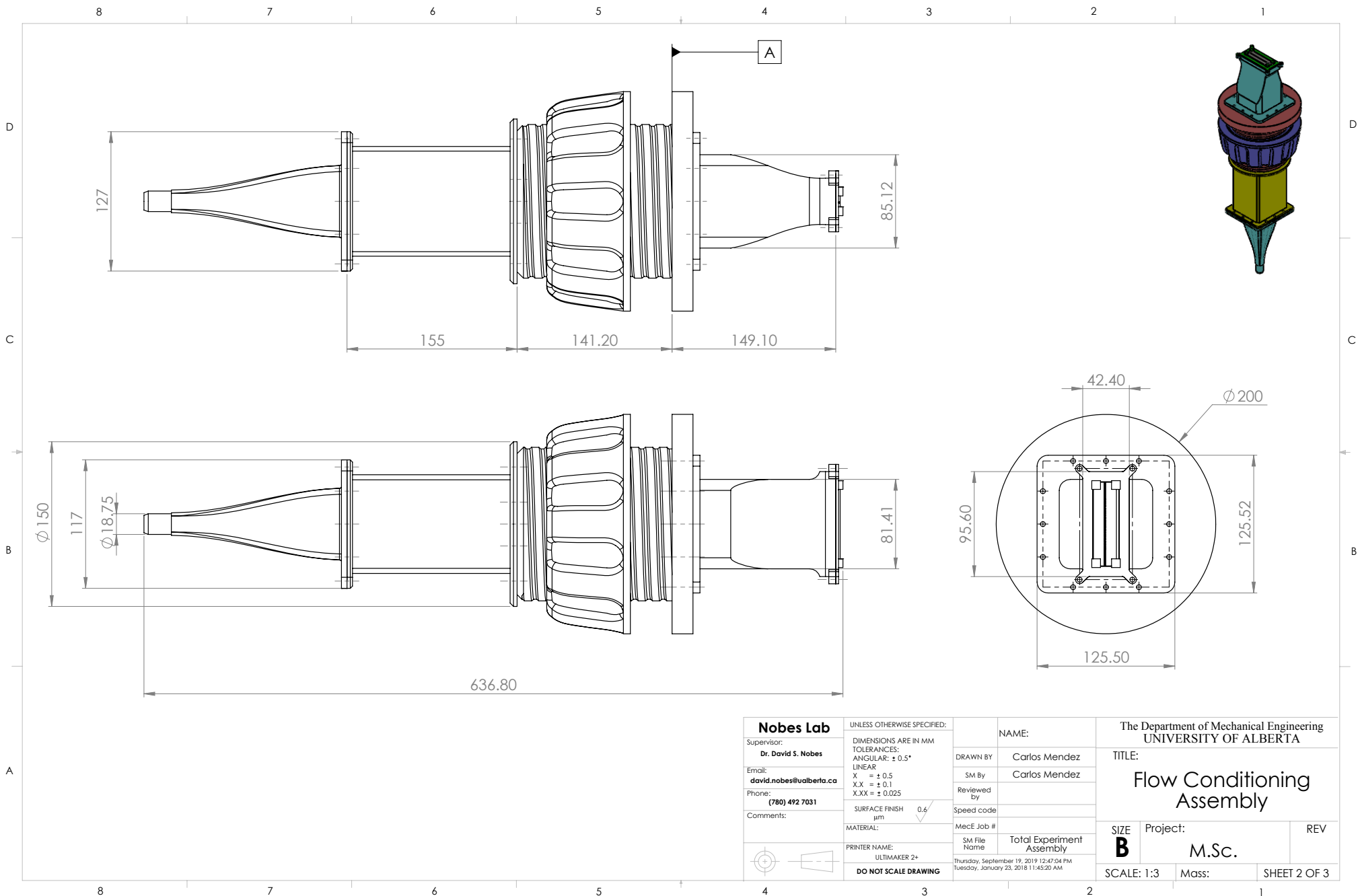
A Flow Conditioning Assembly Drawing Package

This section contains the drawing package for the flow conditioning assembly components in the following order:

- The Flow Conditioning Assembly (pp.136-138)
- The Inlet Diffuser (p.139)
- The Screen Holder Assembly (pp.140-142)
- The Straight Piece (pp.143-145)
- The Honeycomb (p.146)
- The Custom Nut Assembly (pp.147-149)
- The Coupon Contraction (pp.150-152)
- The Slot Assembly (pp.153-154)
- The Coupon Contraction Lid (p.155)

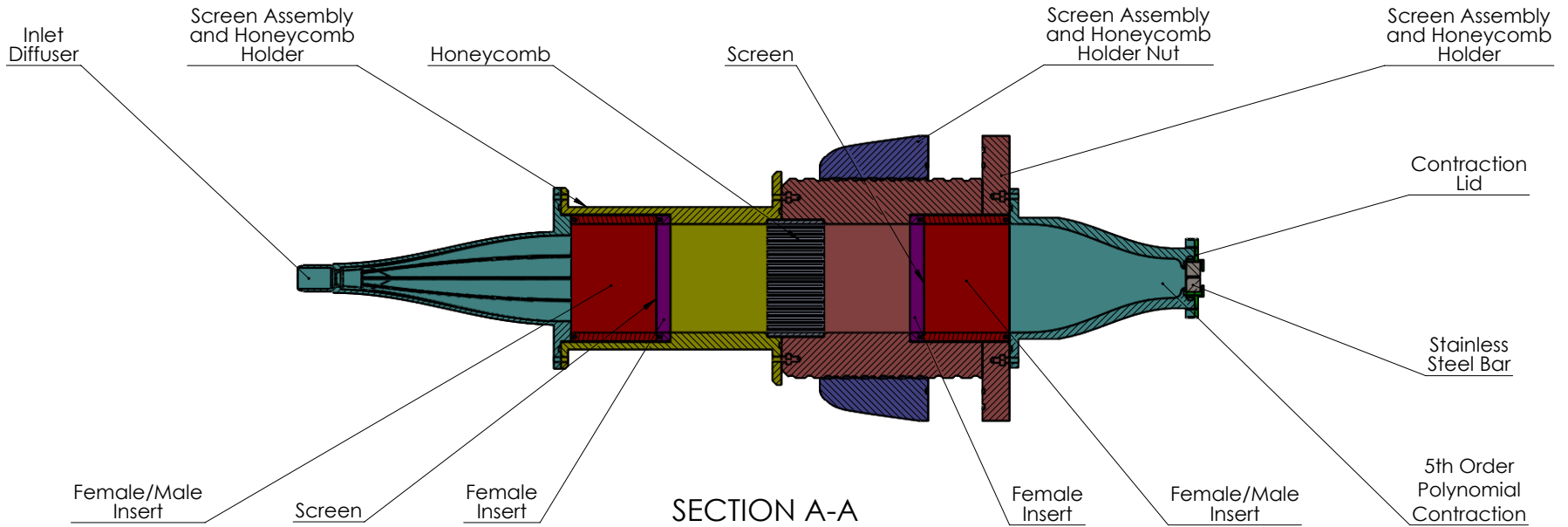
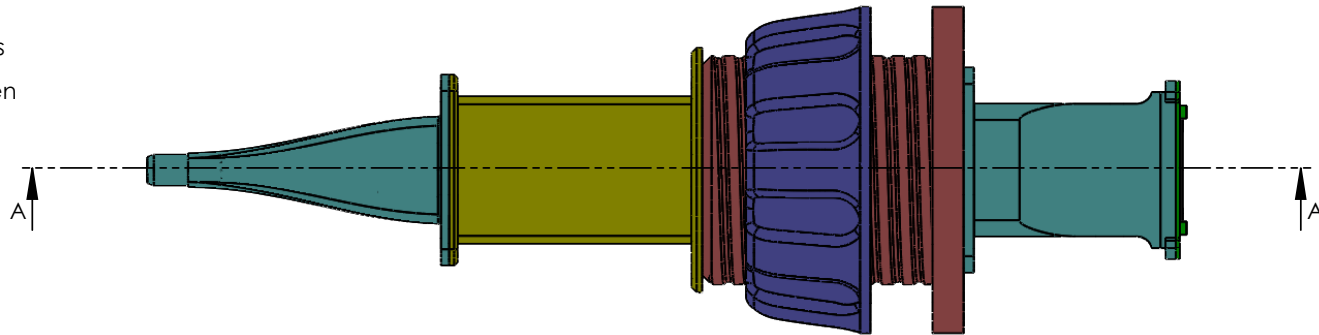


Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE: Flow Conditioning Assembly	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		SIZE: B	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6		Reviewed By:		Project: M.Sc.	
Comments:		MATERIAL:		MecE Job #		REV	
PRINTER NAME: ULTIMAKER 2+		SM File Name: Total Experiment Assembly		Speed code		SCALE: 1:4	
DO NOT SCALE DRAWING		Thursday, September 19, 2019 12:47:04 PM Tuesday, January 23, 2018 11:45:20 AM		Contraction Lid		Mass: SHEET 1 OF 3	



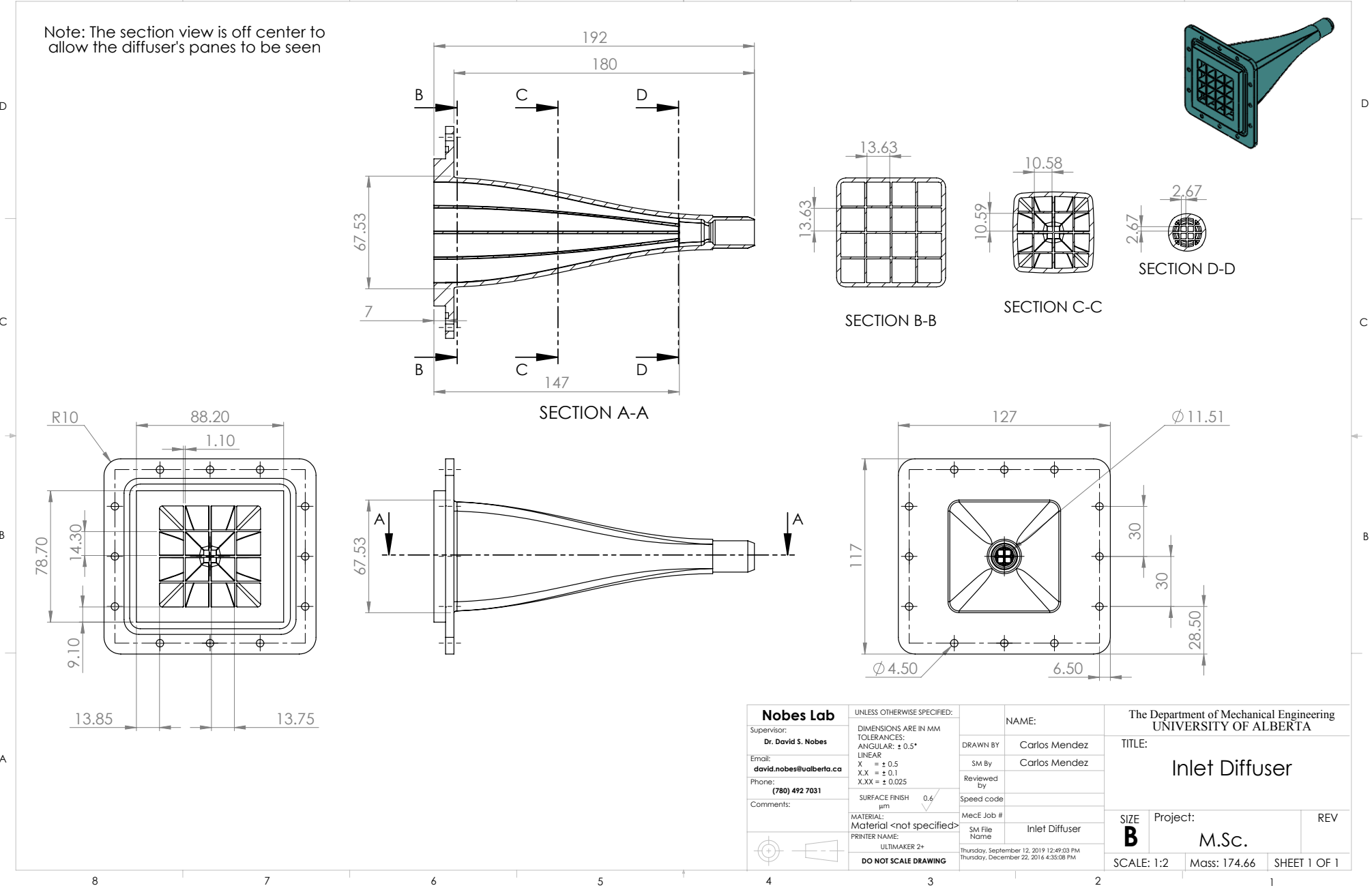
Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE: Flow Conditioning Assembly	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		SIZE B	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:		Project: M.Sc.	
Comments:		MATERIAL:		MecE Job #		REV	
PRINTER NAME: ULTIMAKER 2+		PRINTER NAME: ULTIMAKER 2+		SM File Name Total Experiment Assembly		SCALE: 1:3	
DO NOT SCALE DRAWING		Thursday, September 19, 2019 12:47:04 PM Tuesday, January 23, 2018 11:45:20 AM		Mass:		SHEET 2 OF 3	

Note: The section view is off center to allow the diffuser's panes to be seen

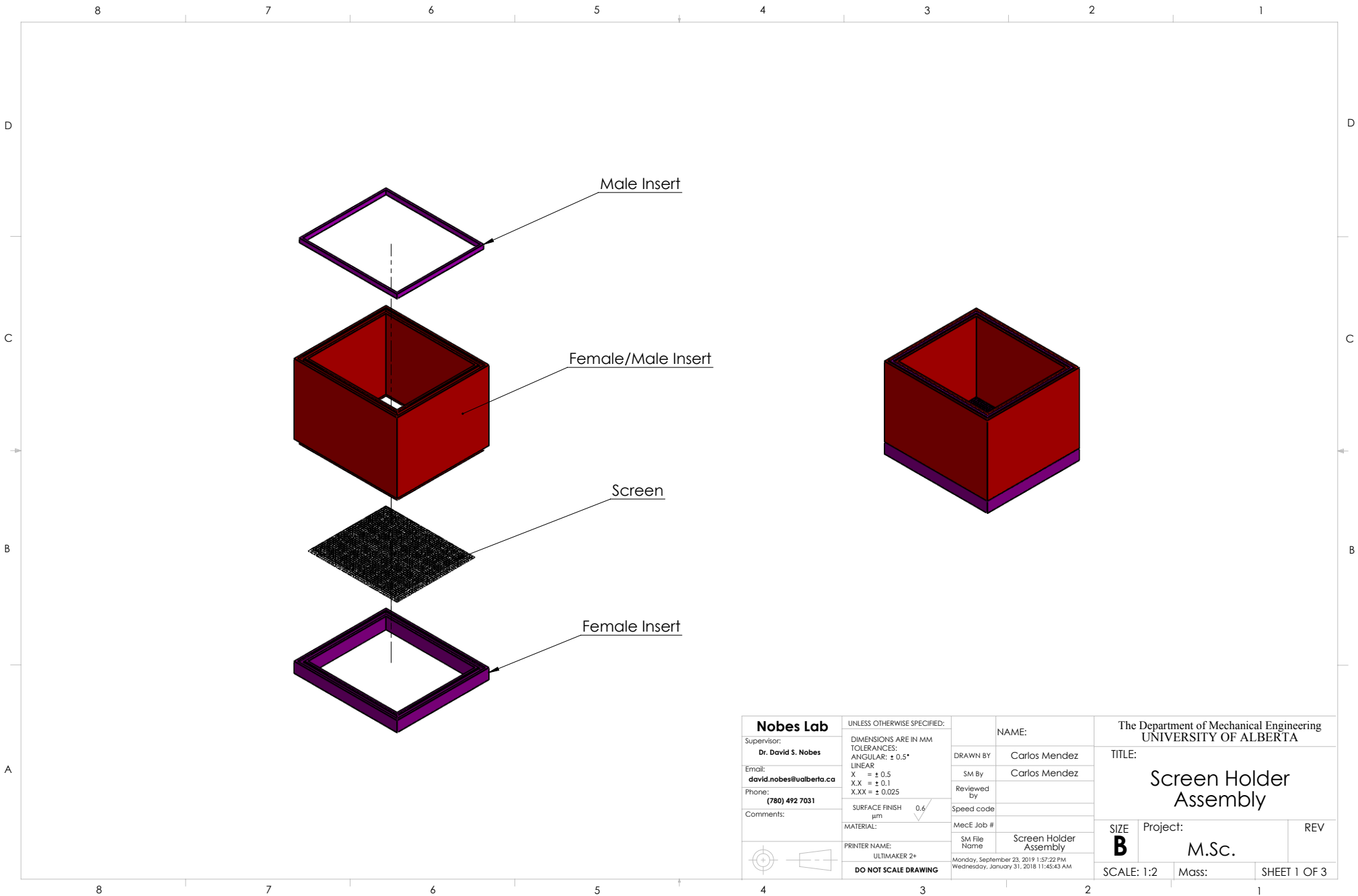


Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM		DRAWN BY: Carlos Mendez		TITLE: Flow Conditioning Assembly	
Email: david.nobes@ualberta.ca		TOLERANCES: ANGULAR: $\pm 0.5^\circ$		SM By: Carlos Mendez		Project: M.Sc.	
Phone: (780) 492 7031		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		Reviewed By:		REV	
Comments:		SURFACE FINISH μm 0.6		Speed code		SIZE B	
		MATERIAL:		MecE Job #		Mass:	
		PRINTER NAME: ULTIMAKER 2+		SM File Name Total Experiment Assembly		SHEET 3 OF 3	
		DO NOT SCALE DRAWING		Thursday, September 19, 2019 12:47:04 PM Tuesday, January 23, 2018 11:45:20 AM		SCALE: 1:3	

Note: The section view is off center to allow the diffuser's panes to be seen



Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE: Inlet Diffuser	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		SIZE B	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:		Project: M.Sc.	
Comments:		MATERIAL: Material <not specified>		MecE Job #		REV	
		PRINTER NAME: ULTIMAKER 2+		SM File Name Inlet Diffuser		SCALE: 1:2	
		DO NOT SCALE DRAWING		Thursday, September 12, 2019 12:49:03 PM Thursday, December 22, 2016 4:35:08 PM		Mass: 174.66	
						SHEET 1 OF 1	

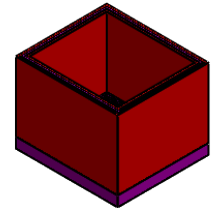


Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes	DIMENSIONS ARE IN MM	TOLERANCES:		DRAWN BY:	Carlos Mendez	TITLE:	
Email: david.nobes@ualberta.ca	ANGULAR: $\pm 0.5^\circ$	LINEAR:		SM By:	Carlos Mendez	Screen Holder Assembly	
Phone: (780) 492 7031	X = ± 0.5	X.X = ± 0.1		Reviewed By:		SIZE	Project:
Comments:	X.XX = ± 0.025	SURFACE FINISH μm		Speed code		B	M.Sc.
		0.6 <input checked="" type="checkbox"/>		MecE Job #		SCALE: 1:2	REV
		MATERIAL:		SM File Name	Screen Holder Assembly	Mass:	SHEET 1 OF 3
		PRINTER NAME:		Monday, September 23, 2019 1:57:22 PM			
		ULTIMAKER 2+		Wednesday, January 31, 2018 11:45:43 AM			
		DO NOT SCALE DRAWING					

8 7 6 5 4 3 2 1

D

D



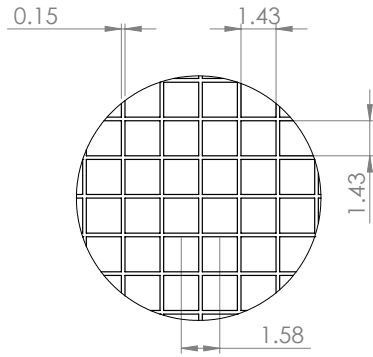
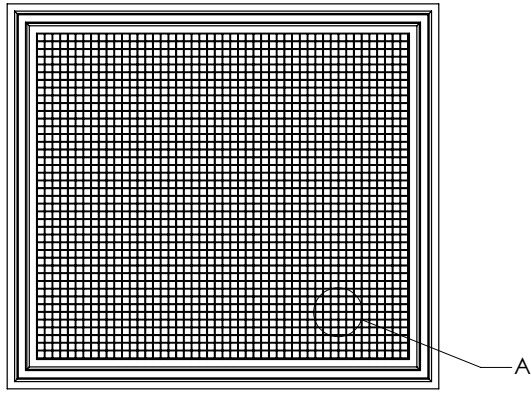
C

B

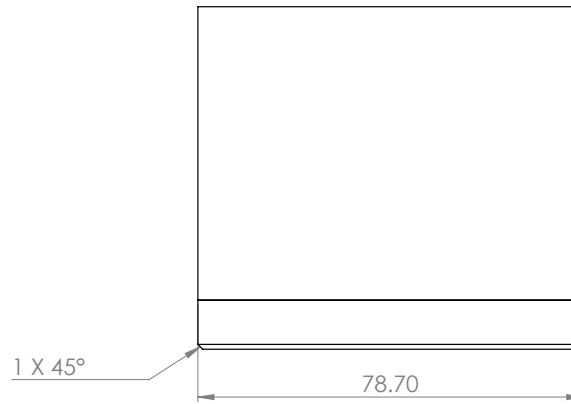
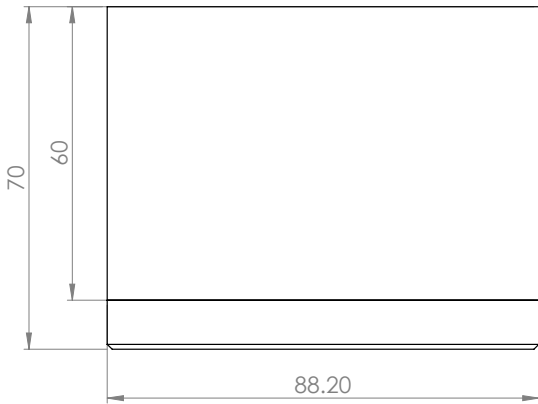
C

B

A

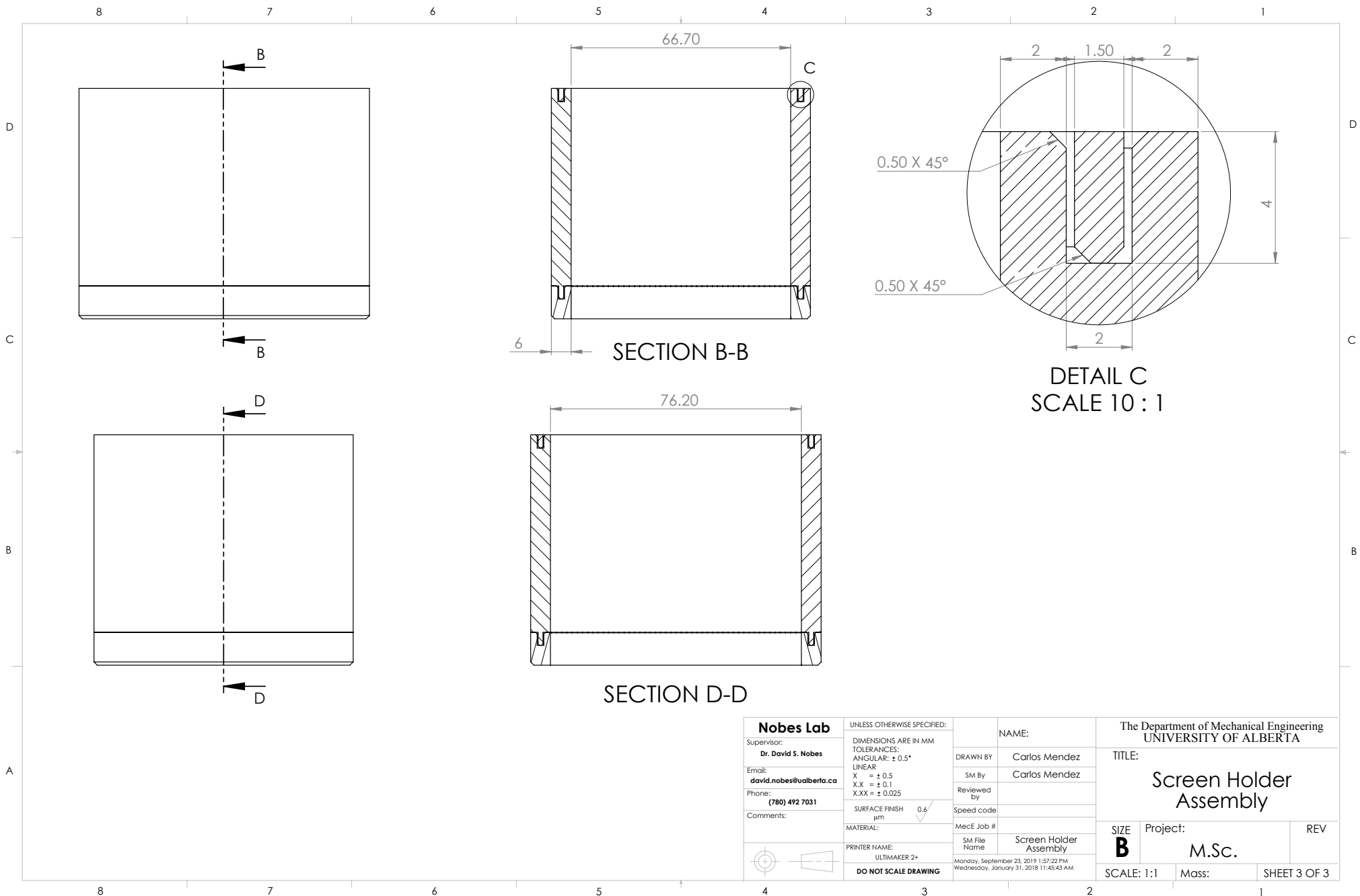


DETAIL A
SCALE 5 : 1

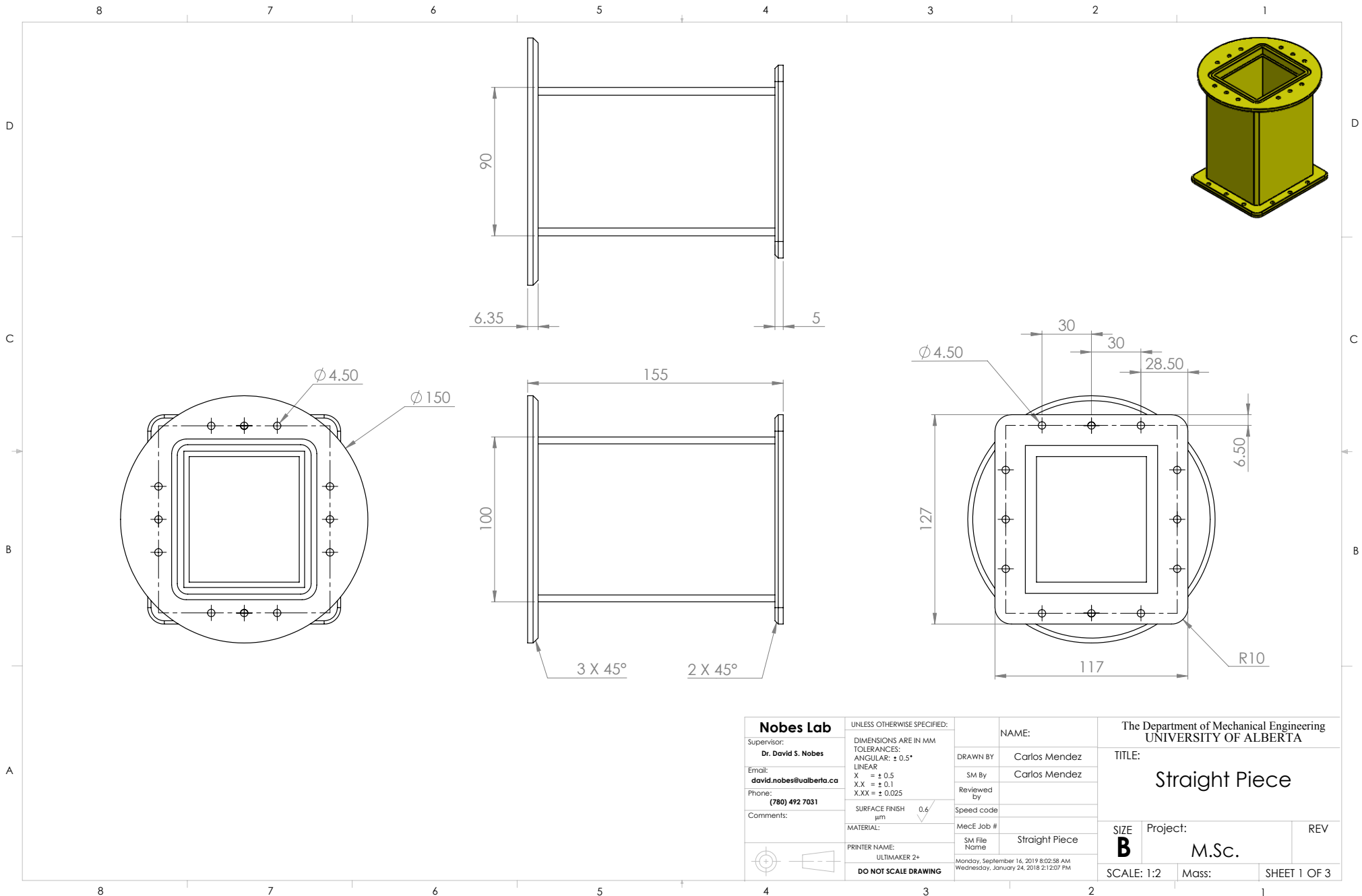


8 7 6 5 4 3 2 1

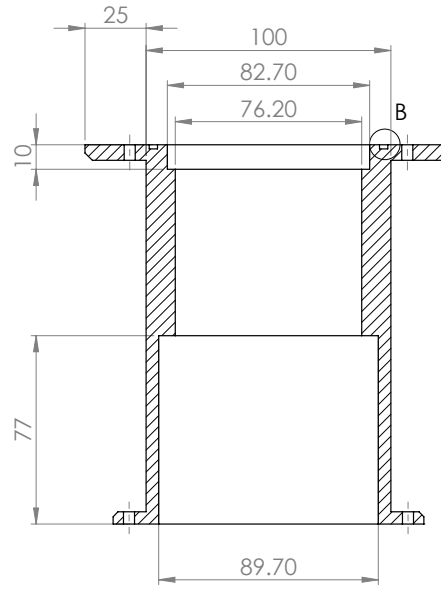
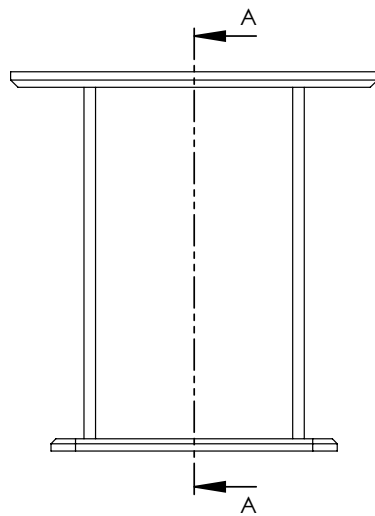
Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Screen Holder Assembly	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:		SIZE	
Comments:		MATERIAL:		Speed code:		Project:	
PRINTER NAME: ULTIMAKER 2+		MecE Job #		SM File Name: Screen Holder Assembly		REV	
DO NOT SCALE DRAWING		Monday, September 23, 2019 1:57:22 PM Wednesday, January 31, 2018 11:45:43 AM		SCALE: 1:1		Mass:	
						SHEET 2 OF 3	



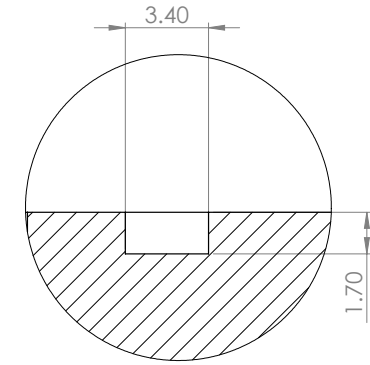
Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE: Screen Holder Assembly	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		SIZE B	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6		Reviewed By:		Project: M.Sc.	
Comments:		MATERIAL:		Speed code:		REV	
		PRINTER NAME: ULTIMAKER 2+		SM File Name: Screen Holder Assembly		SCALE: 1:1	
		DO NOT SCALE DRAWING		Monday, September 23, 2019 1:57:22 PM Wednesday, January 31, 2018 11:45:43 AM		Mass: SHEET 3 OF 3	



Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE: Straight Piece	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		SIZE: B	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6		Reviewed By:		Project: M.Sc.	
Comments:		MATERIAL:		MecE Job #		REV	
		PRINTER NAME: ULTIMAKER 2+		SM File Name: Straight Piece		SCALE: 1:2	
		DO NOT SCALE DRAWING		Monday, September 16, 2019 8:02:58 AM Wednesday, January 24, 2018 2:12:07 PM		Mass: SHEET 1 OF 3	

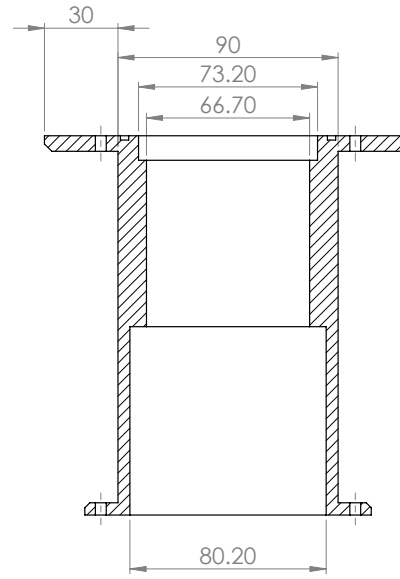
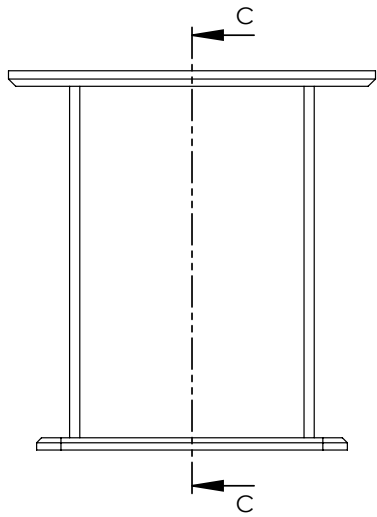


SECTION A-A



DETAIL B
SCALE 5 : 1

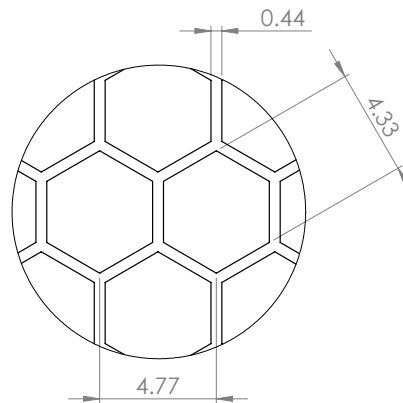
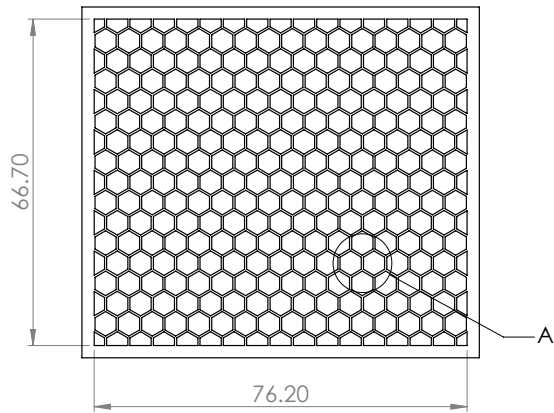
Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes	DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$	DRAWN BY: Carlos Mendez		TITLE:		Straight Piece	
Email: david.nobes@ualberta.ca	LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025	SM By: Carlos Mendez	Reviewed By:	SIZE		Project:	REV
Phone: (780) 492 7031	SURFACE FINISH μm 0.6	MecE Job #	Speed code	B		M.Sc.	
Comments:	MATERIAL:	SM File Name	Straight Piece	SCALE: 1:2		Mass:	SHEET 2 OF 3
	PRINTER NAME: ULTIMAKER 2+	Monday, September 16, 2019 8:02:58 AM Wednesday, January 24, 2018 2:12:07 PM					
DO NOT SCALE DRAWING							



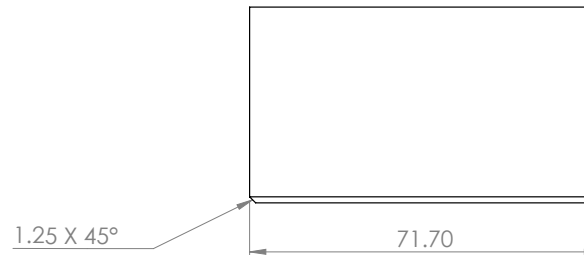
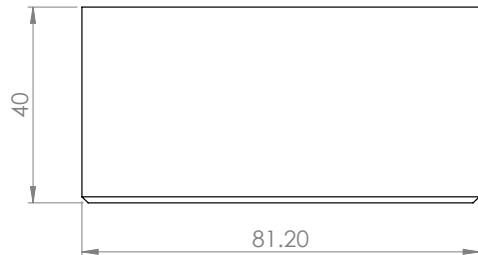
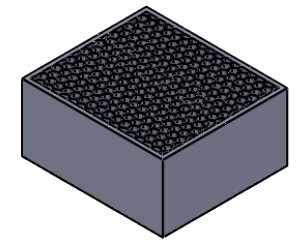
SECTION C-C

Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes	DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$	DRAWN BY: Carlos Mendez		TITLE:		Straight Piece	
Email: david.nobes@ualberta.ca	LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025	SM By: Carlos Mendez	Reviewed By:	SIZE		Project:	REV
Phone: (780) 492 7031	SURFACE FINISH μm 0.6	Speed code	MecE Job #	B		M.Sc.	
Comments:	MATERIAL:	SM File Name	Straight Piece	SCALE: 1:2		Mass:	SHEET 3 OF 3
	PRINTER NAME: ULTIMAKER 2+	Monday, September 16, 2019 8:02:58 AM Wednesday, January 24, 2018 2:12:07 PM					
	DO NOT SCALE DRAWING						

8 7 6 5 4 3 2 1



DETAIL A
SCALE 5 : 1



D

C

B

A

D

C

B

8 7 6 5 4 3 2 1

Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE: Honeycomb	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		REV	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:		SIZE B	
Comments:		MATERIAL: Material <not specified>		MecE Job #		Project: M.Sc.	
		PRINTER NAME: ULTIMAKER 2+		SM File Name Honeycomb - Carlos		SCALE: 1:1	
		DO NOT SCALE DRAWING		Monday, November 05, 2018 3:35:09 PM Thursday, November 16, 2017 9:32:15 PM		Mass: 64.84	
						SHEET 1 OF 1	

8 7 6 5 4 3 2 1

D

C

B

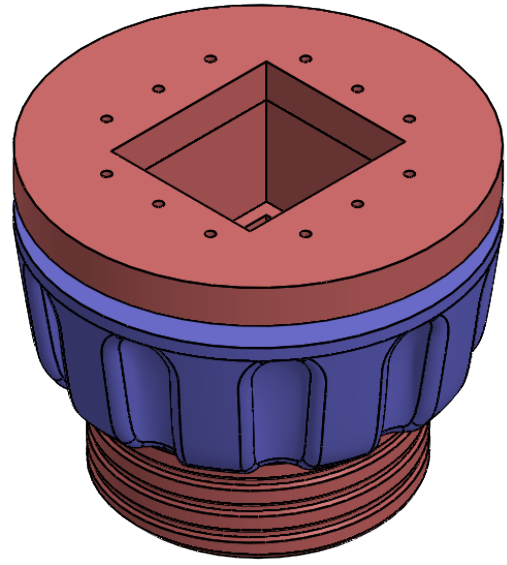
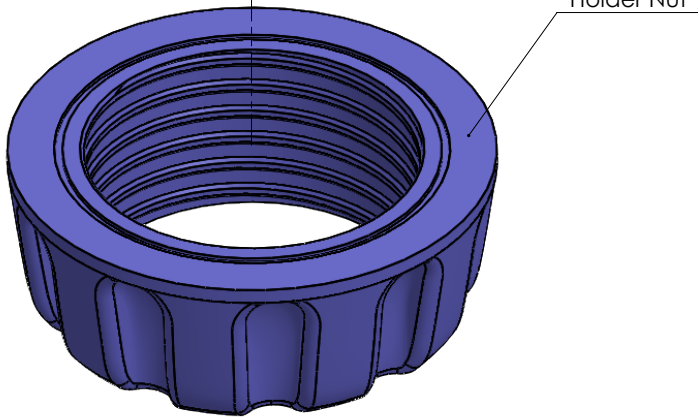
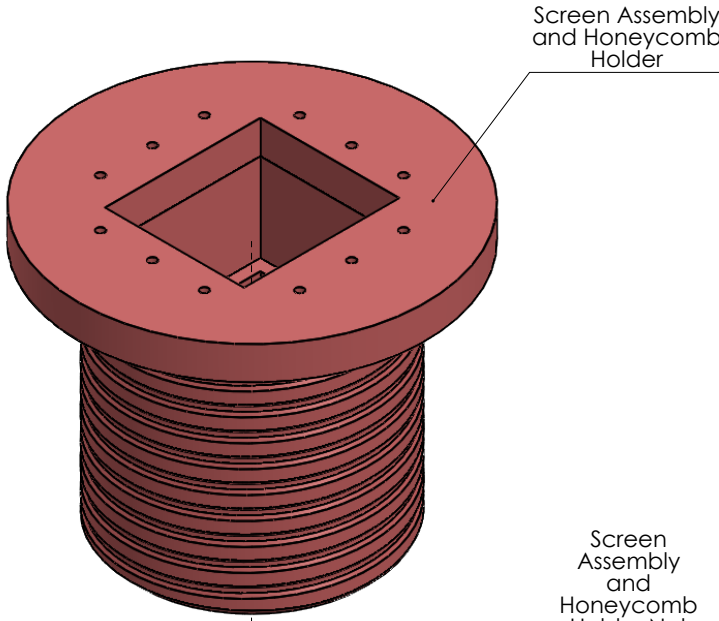
A

D

C

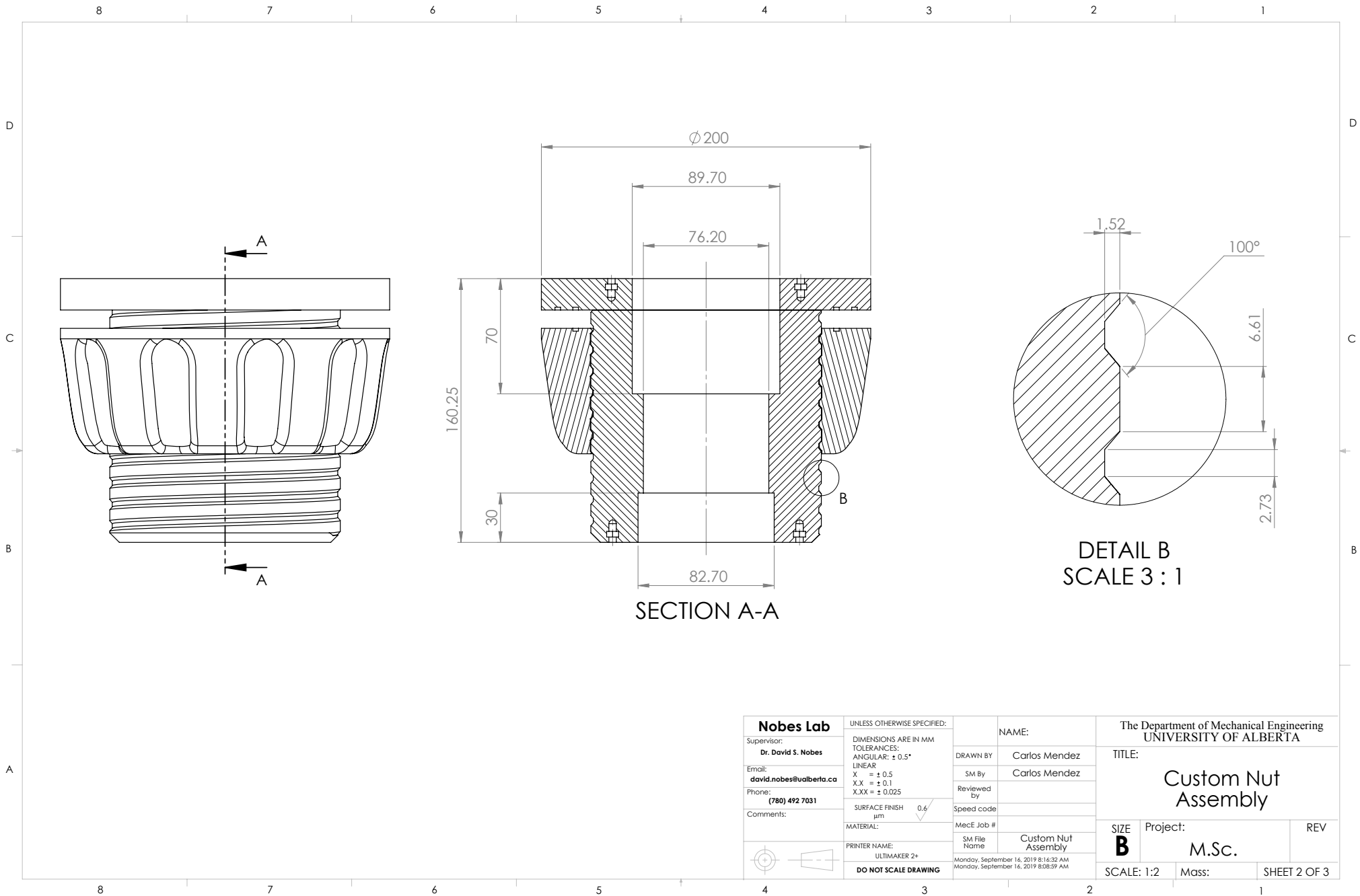
B

A



Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Custom Nut Assembly	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6		Reviewed By:			
Comments:		MATERIAL:		MecE Job #		SIZE	
PRINTER NAME: ULTIMAKER 2+		MATERIAL:		SM File Name: Custom Nut Assembly		Project:	
DO NOT SCALE DRAWING		MATERIAL:		Monday, September 16, 2019 8:14:32 AM Monday, September 16, 2019 8:08:59 AM		REV	
		SCALE: 1:2		Mass:		SHEET 1 OF 3	

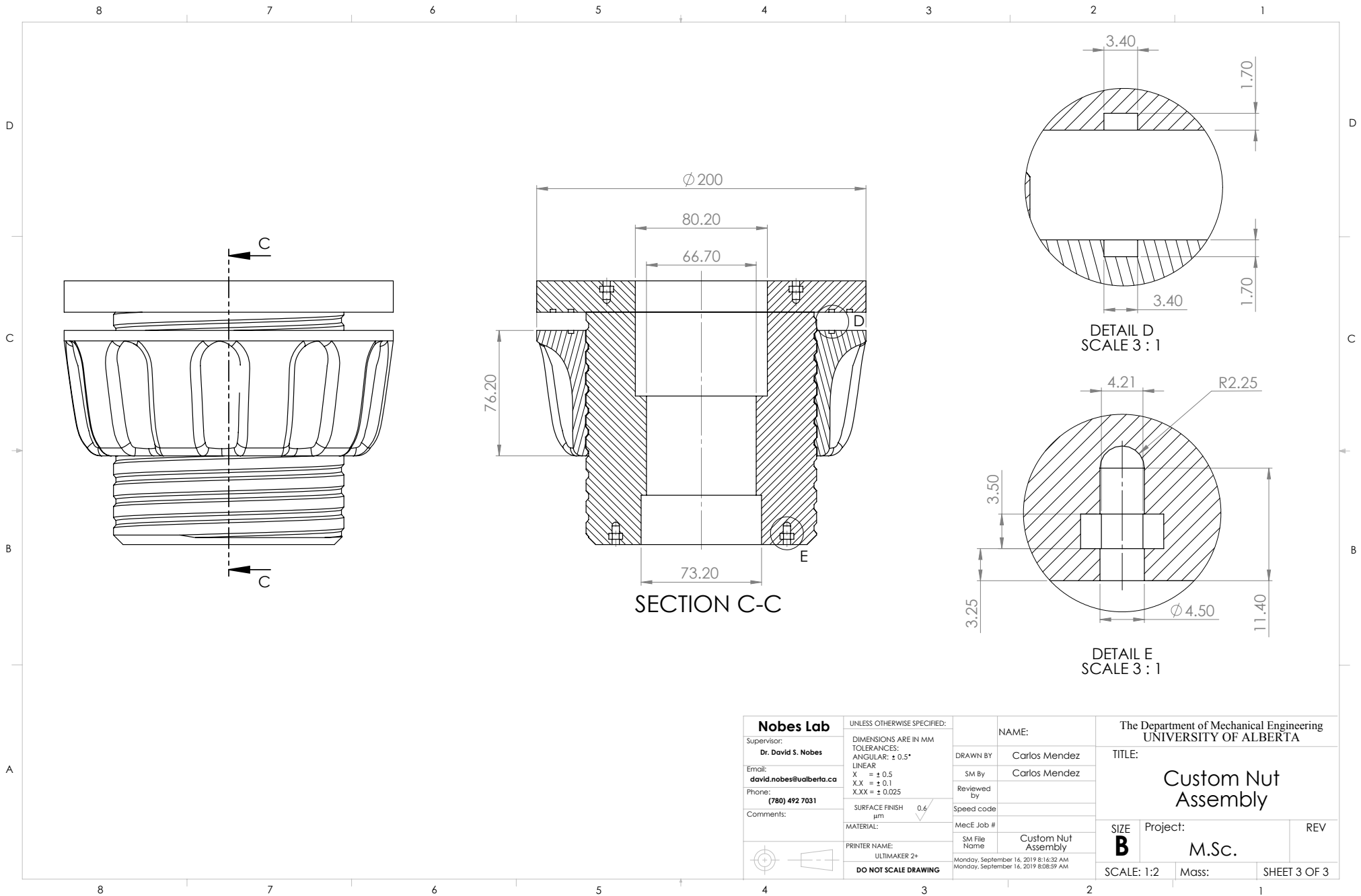
8 7 6 5 4 3 2 1



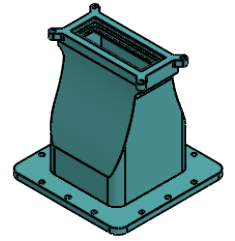
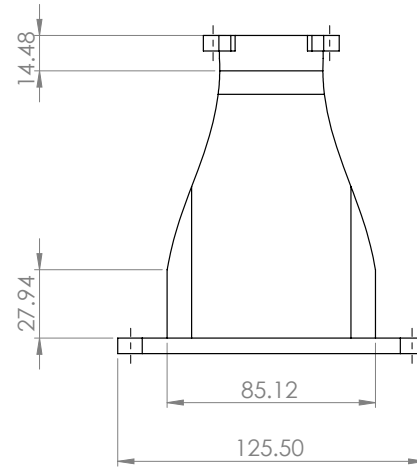
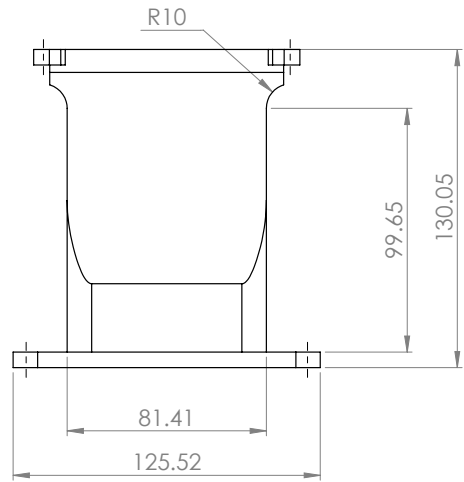
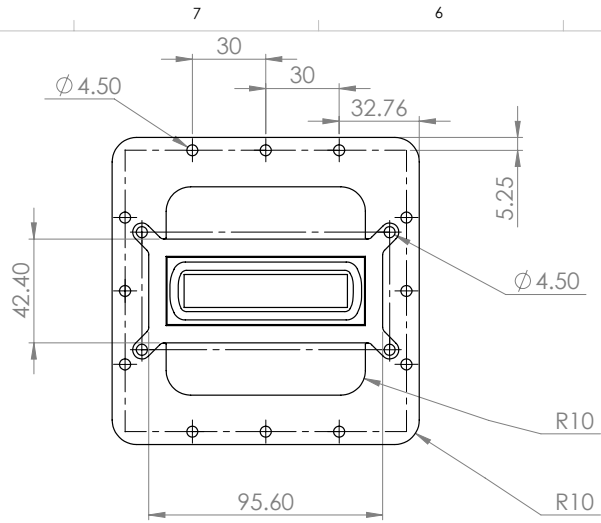
SECTION A-A

DETAIL B
SCALE 3 : 1

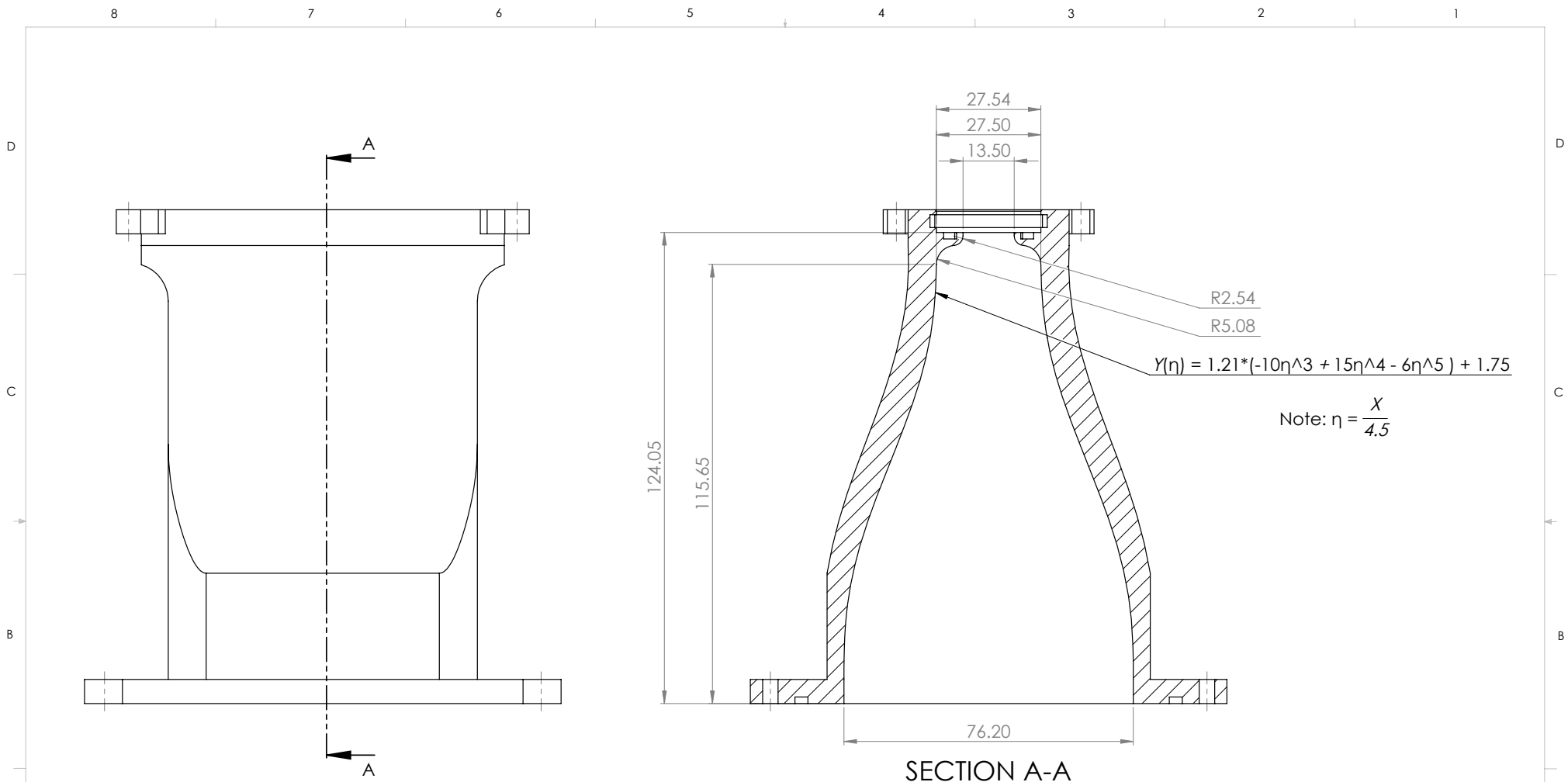
Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Custom Nut Assembly	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:			
Comments:		MATERIAL:		MecE Job #		SIZE	
PRINTER NAME: ULTIMAKER 2+		PRINTER NAME: ULTIMAKER 2+		SM File Name Custom Nut Assembly		Project: M.Sc.	
DO NOT SCALE DRAWING		Monday, September 16, 2019 8:14:32 AM Monday, September 16, 2019 8:08:59 AM		Speed code		REV	
				SCALE: 1:2		Mass: SHEET 2 OF 3	



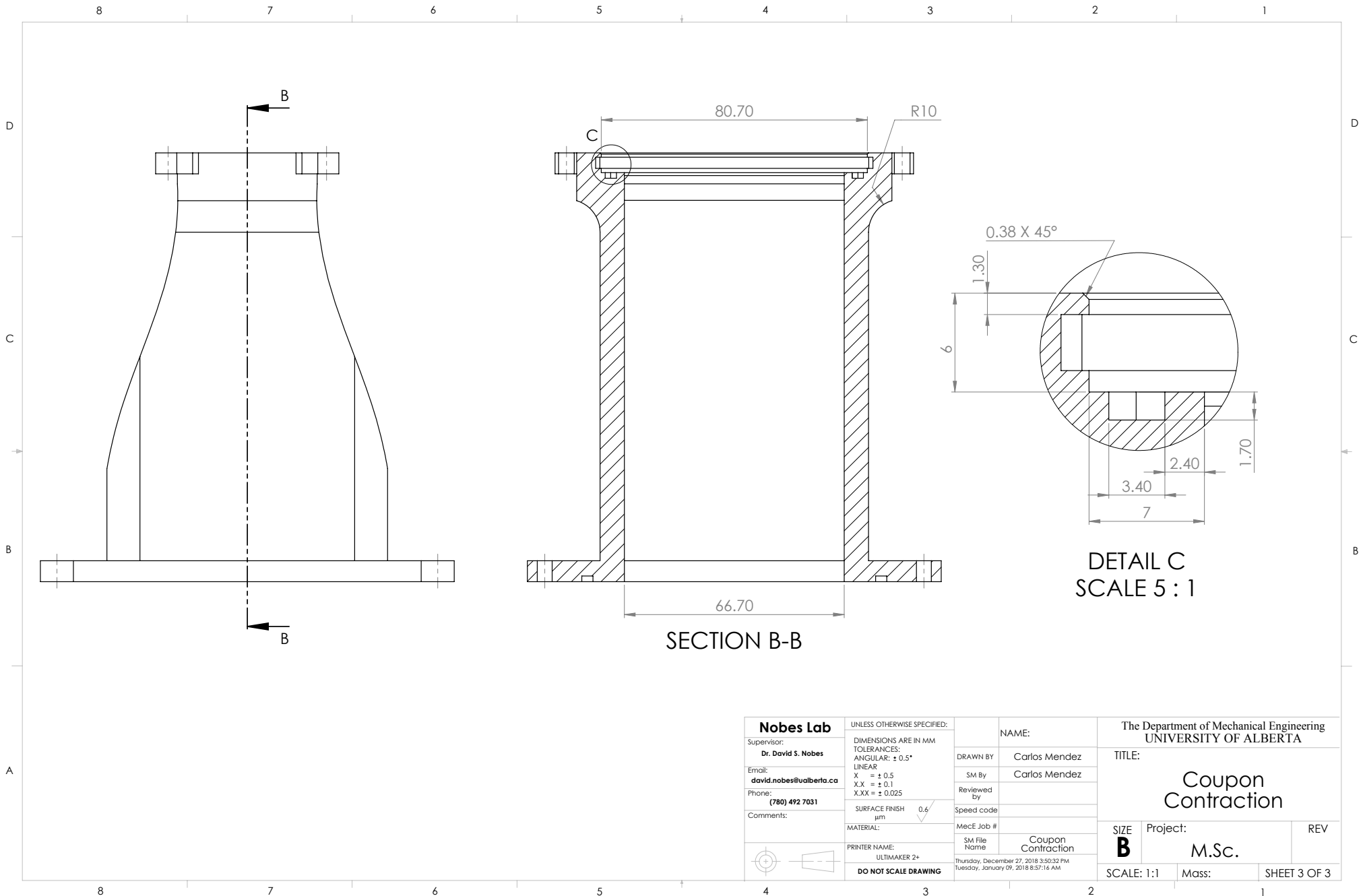
Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Custom Nut Assembly	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:			
Comments:		MATERIAL:		MecE Job #		SIZE	
PRINTER NAME: ULTIMAKER 2+		PRINTER NAME: ULTIMAKER 2+		SM File Name Custom Nut Assembly		Project: M.Sc.	
DO NOT SCALE DRAWING		Monday, September 16, 2019 8:14:32 AM Monday, September 16, 2019 8:08:59 AM		Speed code		REV	
				SCALE: 1:2		Mass: SHEET 3 OF 3	



Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Coupon Contraction	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6 ✓		Reviewed By:			
Comments:		MATERIAL:		MecE Job #		SIZE	
PRINTER NAME: ULTIMAKER 2+		SM File Name Coupon Contraction		Speed code		Project:	
DO NOT SCALE DRAWING		Thursday, December 27, 2018 3:50:32 PM Tuesday, January 09, 2018 8:57:14 AM		SM File Name Coupon Contraction		M.Sc.	
				SCALE: 1:2		Mass:	
						REV	
						SHEET 1 OF 3	



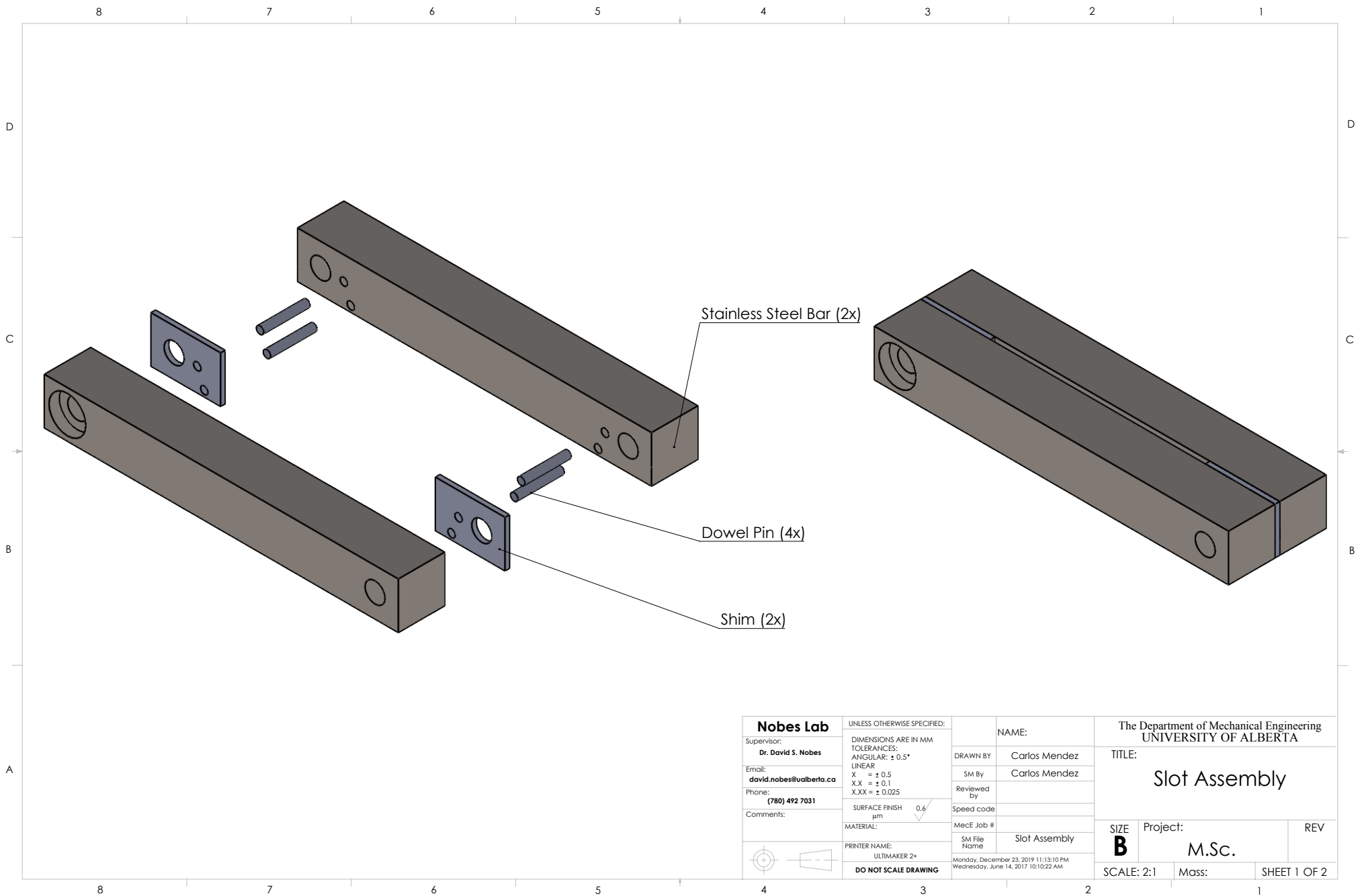
Nobes Lab Supervisor: Dr. David S. Nobes Email: david.nobes@ualberta.ca Phone: (780) 492 7031 Comments:	UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$ LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025	NAME:	The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
		DRAWN BY:	Carlos Mendez	TITLE:
		SM By:	Carlos Mendez	
		Reviewed By:		
		Speed code:		
	SURFACE FINISH μm 0.6	MecE Job #:		
	MATERIAL:	SM File Name:	Coupon Contraction	
	PRINTER NAME: ULTIMAKER 2+	Thursday, December 27, 2018 3:50:32 PM Tuesday, January 09, 2018 8:57:14 AM		
	DO NOT SCALE DRAWING	SCALE: 1:1	Mass:	SHEET 2 OF 3



SECTION B-B

DETAIL C
SCALE 5 : 1

Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Coupon Contraction	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6		Reviewed by:			
Comments:		MATERIAL:		Speed code:		SIZE	
PRINTER NAME: ULTIMAKER 2+		MecE Job #		SM File Name Coupon Contraction		Project: M.Sc.	
DO NOT SCALE DRAWING		Thursday, December 27, 2018 3:50:32 PM Tuesday, January 09, 2018 8:57:14 AM		SCALE: 1:1		REV	
						Mass:	
						SHEET 3 OF 3	

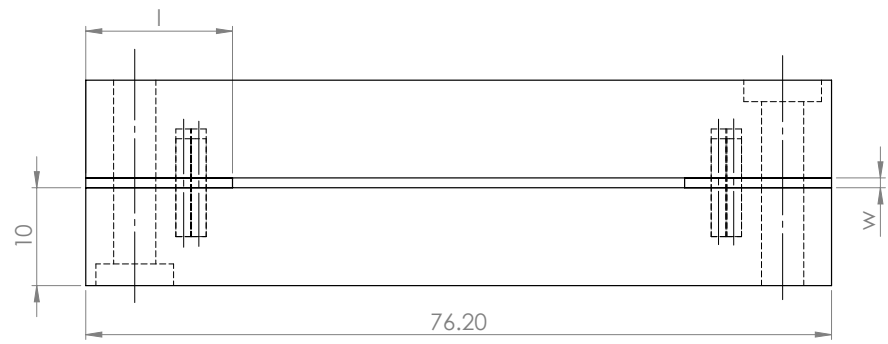
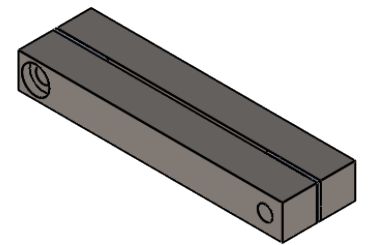


Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE: Slot Assembly	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		REV	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6 <input checked="" type="checkbox"/>		Reviewed By:		SIZE B	
Comments:		MATERIAL:		MecE Job #		Project: M.Sc.	
		PRINTER NAME: ULTIMAKER 2+		SM File Name Slot Assembly		SCALE: 2:1	
		DO NOT SCALE DRAWING		Monday, December 23, 2019 11:13:10 PM Wednesday, June 14, 2017 10:10:22 AM		Mass: SHEET 1 OF 2	

8 7 6 5 4 3 2 1

D

D

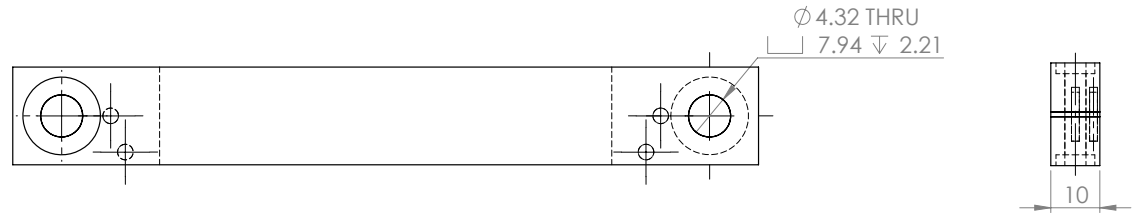


C

C

B

B



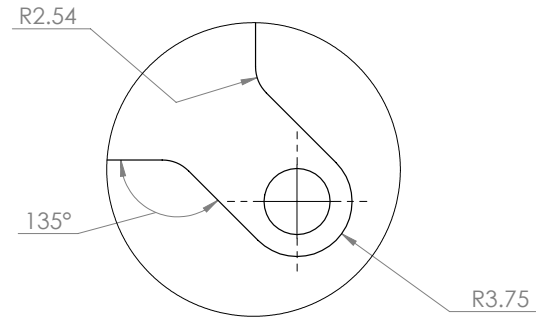
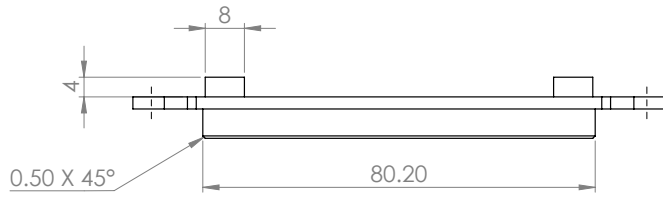
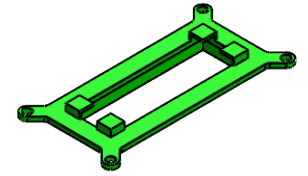
Note: l and w varied during experimentation and had the following values:

l	w
20.78	0.35
16.89	0.28
13.61	0.25
13.00	0.24

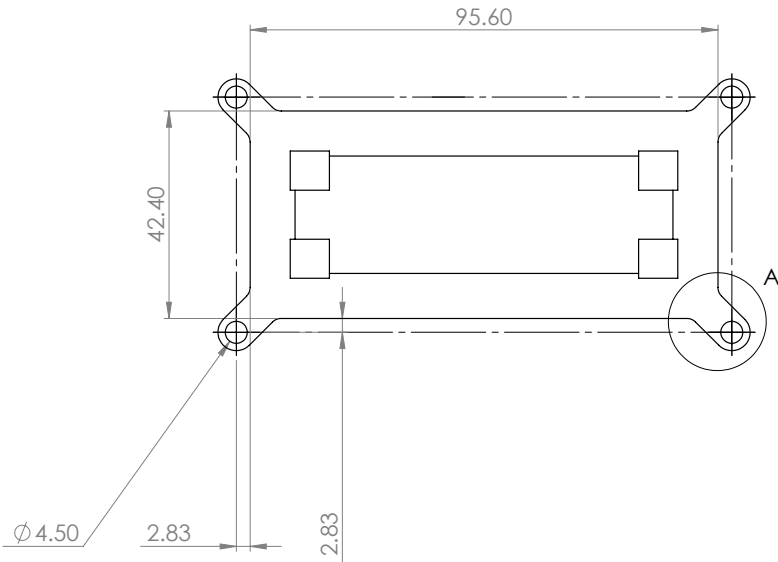
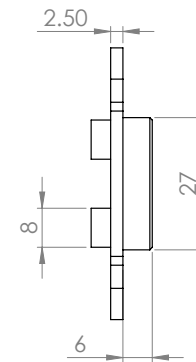
A

Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: $\pm 0.5^\circ$		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Slot Assembly	
Phone: (780) 492 7031		SURFACE FINISH μm 0.6		Reviewed By:		SIZE	
Comments:		MATERIAL:		Speed code		Project:	
PRINTER NAME: ULTIMAKER 2+		MecE Job #		SM File Name: Slot Assembly		REV	
DO NOT SCALE DRAWING		Monday, December 23, 2019 11:13:10 PM Wednesday, June 14, 2017 10:10:22 AM		SCALE: 2:1		Mass:	

8 7 6 5 4 3 2 1



DETAIL A
SCALE 3 : 1



Nobes Lab		UNLESS OTHERWISE SPECIFIED:		NAME:		The Department of Mechanical Engineering UNIVERSITY OF ALBERTA	
Supervisor: Dr. David S. Nobes		DIMENSIONS ARE IN MM TOLERANCES: ANGULAR: ± 0.5°		DRAWN BY: Carlos Mendez		TITLE:	
Email: david.nobes@ualberta.ca		LINEAR: X = ± 0.5 X.X = ± 0.1 X.XX = ± 0.025		SM By: Carlos Mendez		Coupon Contraction Lid	
Phone: (780) 492 7031		SURFACE FINISH µm 0.6		Reviewed By:			
Comments:		MATERIAL:		MecE Job #		SIZE	
PRINTER NAME: ULTIMAKER 2+		SM File Name: Coupon Contraction Lid		Speed code		Project:	
DO NOT SCALE DRAWING		Tuesday, September 17, 2019 10:13:33 AM Thursday, February 01, 2018 9:03:58 AM		SM File Name: Coupon Contraction Lid		M.Sc.	
				SCALE: 1:1		REV	
				Mass:		SHEET 1 OF 1	

B MATLAB Codes for 2D-PIV Post-Processing

This section contains the MATLAB codes that were developed for further post-processing of the 2D-PIV data. Specifically, the codes include those that were used in the stitching process, and those that were used to produce the figures presented within this study.

```

function get2Dfiles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% get2Dfiles Extract 2D-PIV files from their locations and save to
% structure array.
%
% get2Dfiles works to get the average velocity and average rms
% velocity results files for each aspect ratio, cam, and flow rate
% for the 2D-PIV experiments. The results are loaded into a
% structure array, which has the following fields: AvgV, AvgV_rms,
% AR, Camera, Flowrate, and DataOrigin.
%
% Who : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Getting overall results folder. Location is saved as output
% variable, if needed later. Directory listing structure is made,
% and first two rows are deleted as they are non-physical listings.
OverallDataFolderloc = uigetdir;
listing_info = dir(OverallDataFolderloc);
listing_info(~[listing_info.isdir]) = []; %get rid of non folders
listing_info(ismember({listing_info.name}, {'.', '..'})) = []; %get
rid of . and ..
subdir_names = fullfile(OverallDataFolderloc, {listing_info.name});
%construct list of names
ARFolderNames = subdir_names.';

TwoDFilesArray(48,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'Camera', [], ...
'FlowRate', [], 'DataOrigin', []);

% Loop to go through overall results folder, get files and save
% them in FilesArray output variable.
structloc = 1;
% Going through AR
for i = 1:4
    string = ARFolderNames(i);
    string = num2str(cell2mat(string));
    ARFolderName = string;
    ARListing_info = dir(ARFolderName);
    ARListing_info(~[ARListing_info.isdir]) = []; %get rid of non
folders
    ARListing_info(ismember({ARListing_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    ARFsubdir_names = fullfile(ARFolderName, {ARListing_info.name});
    %construct list of names
    FlowFolderNames = ARFsubdir_names(1,1:6).'; %first 6 folders are
flow folders

    % Going through Flow Folders.
    for j = 1:6

```

```

    string2 = FlowFolderNames(j);
    string2 = num2str(cell2mat(string2));
    FlowFolderName = string2;

    FFlisting_info = dir(FlowFolderName);
    FFlisting_info(~[FFlisting_info.isdir]) = []; %get rid of non
folders
    FFlisting_info(ismember({FFlisting_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    FFsubdir_names = fullfile(FlowFolderName,
{FFlisting_info.name}); %construct list of names
    CamerasFolder = FFsubdir_names(1,1:2).';

    % Going through camera folders.
    for k = 1:2
        string3 = CamerasFolder(k);
        string3 = num2str(cell2mat(string3));
        CameraFolderName = string3;

        % Storing the AvgV, AvgV_rms, AR, FlowRate, PlaneLoc, and
        % DataOrigin in the FileArray structure in their
        % respective fields.
        AvgVelString = [CameraFolderName '\B00001_AvgV.vc7'];
        TwoDFilesArray(structloc,1).AvgV = loadvec(AvgVelString);
        AvgVelRMSString = [CameraFolderName '\B00002_StdevV.vc7'];
        TwoDFilesArray(structloc,1).AvgV_rms =
loadvec(AvgVelRMSString);
        TwoDFilesArray(structloc,1).DataOrigin = CameraFolderName;

        revFFN = fliplr(CameraFolderName);
        [token,remain] = strtok(revFFN,'\');
        Camera = fliplr(token);

        [token2,remain2] = strtok(remain,'\');
        FlowRate = fliplr(token2);

        [token2,remain3] = strtok(remain2,'\');
        AR = fliplr(token2);

        TwoDFilesArray(structloc,1).AR = strtrim(AR);
        TwoDFilesArray(structloc,1).Camera = strtrim(Camera);
        TwoDFilesArray(structloc,1).FlowRate = FlowRate;

        structloc = structloc + 1;
    end
end
end

% Save output structure
save('W:\Carlos Mendez\Matlab Results Folder
\2DFiles', 'TwoDFilesArray');
end

```

Published with MATLAB® R2019a

```

function [maxVel] = maxCLsearch(VecField)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% maxCLsearch Function for finding the maximum velocity value
% in a vector field
%
%   maxCLsearch(VecField) %% Function for finding the maximum
%   velocity value in a vector field. This is for 2D data flowing
%   from bottom to top. Only the first quarter of data is used
%   as this is where the highest velocity should be located.
%
% Who   : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% A new vector field made only of the first quarter of the original
% vector field is made, as the maximum velocity should be here.
VF = VecField;
[r,c] = size(VF);
newC = ceil(c/4);
VF2 = VF(1:r,1:newC);

% From the 2D data, the 10 highest values in each column are
% found. This is essentially looking at each velocity profile
% downstream and getting the 10 highest values.
TopTenRows = maxk(VF2,10);

% Matrix array of size (10x5) is made.
TopFive = zeros(10,5);

% The the 5 highest values in each row is found. This way most
% of the values in the potential core region should be obtained
for i = 1:10
    CurTopTenRow = TopTenRows(i,:);
    TopFive(i,:) = maxk(CurTopTenRow,5);
end

% Reshaping the matrix into a (1x50) array so that only outliers can
% be removed. It was in matrix form, the entire column containing the
% outlier would be removed. Matlab removes any value above 3 standard
% deviations of the mean.
TopFive = reshape(TopFive,[1,50]);
OutRemTopFive = rmoutliers(TopFive);

% With the outliers removed, the maximum value is taken, as this
% should be the actual maximum value.
maxVel = max(OutRemTopFive);

end

```

Published with MATLAB® R2019a

```

function get2Dfiles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% get2Dfiles Extract 2D-PIV files from their locations and save to
% structure array.
%
% get2Dfiles works to get the average velocity and average rms
% velocity results files for each aspect ratio, cam, and flow rate
% for the 2D-PIV experiments. The results are loaded into a
% structure array, which has the following fields: AvgV, AvgV_rms,
% AR, Camera, Flowrate, and DataOrigin.
%
% Who : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Getting overall results folder. Location is saved as output
% variable, if needed later. Directory listing structure is made,
% and first two rows are deleted as they are non-physical listings.
OverallDataFolderloc = uigetdir;
listing_info = dir(OverallDataFolderloc);
listing_info(~[listing_info.isdir]) = []; %get rid of non folders
listing_info(ismember({listing_info.name}, {'.', '..'})) = []; %get
rid of . and ..
subdir_names = fullfile(OverallDataFolderloc, {listing_info.name});
%construct list of names
ARFolderNames = subdir_names.';

TwoDFilesArray(48,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'Camera', [], ...
'FlowRate', [], 'DataOrigin', []);

% Loop to go through overall results folder, get files and save
% them in FilesArray output variable.
structloc = 1;
% Going through AR
for i = 1:4
    string = ARFolderNames(i);
    string = num2str(cell2mat(string));
    ARFolderName = string;
    ARListing_info = dir(ARFolderName);
    ARListing_info(~[ARListing_info.isdir]) = []; %get rid of non
folders
    ARListing_info(ismember({ARListing_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    ARFsubdir_names = fullfile(ARFolderName, {ARListing_info.name});
    %construct list of names
    FlowFolderNames = ARFsubdir_names(1,1:6).'; %first 6 folders are
flow folders

    % Going through Flow Folders.
    for j = 1:6

```

```

    string2 = FlowFolderNames(j);
    string2 = num2str(cell2mat(string2));
    FlowFolderName = string2;

    FFlisting_info = dir(FlowFolderName);
    FFlisting_info(~[FFlisting_info.isdir]) = []; %get rid of non
folders
    FFlisting_info(ismember({FFlisting_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    FFsubdir_names = fullfile(FlowFolderName,
{FFlisting_info.name}); %construct list of names
    CamerasFolder = FFsubdir_names(1,1:2).';

    % Going through camera folders.
    for k = 1:2
        string3 = CamerasFolder(k);
        string3 = num2str(cell2mat(string3));
        CameraFolderName = string3;

        % Storing the AvgV, AvgV_rms, AR, FlowRate, PlaneLoc, and
        % DataOrigin in the FileArray structure in their
        % respective fields.
        AvgVelString = [CameraFolderName '\B00001_AvgV.vc7'];
        TwoDFilesArray(structloc,1).AvgV = loadvec(AvgVelString);
        AvgVelRMSString = [CameraFolderName '\B00002_StdevV.vc7'];
        TwoDFilesArray(structloc,1).AvgV_rms =
loadvec(AvgVelRMSString);
        TwoDFilesArray(structloc,1).DataOrigin = CameraFolderName;

        revFFN = fliplr(CameraFolderName);
        [token,remain] = strtok(revFFN,'\');
        Camera = fliplr(token);

        [token2,remain2] = strtok(remain,'\');
        FlowRate = fliplr(token2);

        [token2,remain3] = strtok(remain2,'\');
        AR = fliplr(token2);

        TwoDFilesArray(structloc,1).AR = strtrim(AR);
        TwoDFilesArray(structloc,1).Camera = strtrim(Camera);
        TwoDFilesArray(structloc,1).FlowRate = FlowRate;

        structloc = structloc + 1;
    end
end
end

% Save output structure
save('W:\Carlos Mendez\Matlab Results Folder
\2DFiles', 'TwoDFilesArray');
end

```

Published with MATLAB® R2019a

```

function workonTwoDrawfiles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% workonTwoDrawfiles stitches all of the 2D-PIV data
%
% workonTwoDrawfiles takes the upstream velocity field
% data (V1 & V1rms) along with the downstream velocity field
% data (V2 & V2rms) and sends them to be stitched them using the
% stitcher2 function. The outputs from the function are saved to
% the new structures.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading the 2D-PIV velocity structures.
load('W:\Carlos Mendez\Matlab Results Folder
\2DFiles','TwoDFilesArray');

% Making structures for the stitched PIV data to be saved to. Note:
% different methods of stitching were used, so multiple result
% structures were made, according to the method used. The stitching
% involved stitching with respect to decay, half-width, turbulence
% intensity, and all three values (discussed in the thesis).
StitchedTwoDFiles(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

StitchedTwoDFiles2(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

StitchedTwoDFiles3(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

StitchedTwoDFiles4(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

% Sending the 2D-PIV data to be stitched using the stitcher2 function
% and saving the output to the corresponding structure.
% StitchedTwoDFiles - stitched wrt to all three variables;
% StitchedTwoDFiles2 - stitched wrt to decay
% StitchedTwoDFiles2 - stitched wrt to velocity half width
% StitchedTwoDFiles3 - stitched wrt to velocity turbulence intensity
for i = 1:2:47
    cam1_VEC_temp = TwoDFilesArray(i).AvgV;
    cam2_VEC_temp = TwoDFilesArray(i+1).AvgV;

    cam1_VEC_temp_rms = TwoDFilesArray(i).AvgV_rms;
    cam2_VEC_temp_rms = TwoDFilesArray(i+1).AvgV_rms;

```

```

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

    position = 0.5*(i+1);

    StitchedTwoDFiles(position).AvgV = stitchVel;
    StitchedTwoDFiles(position).AvgV_rms = stitchVelrms;

    StitchedTwoDFiles(position).Cam1AvgV = newV1;
    StitchedTwoDFiles(position).Cam1AvgV_rms = newV1rms;

    StitchedTwoDFiles(position).Cam2AvgV = newV2;
    StitchedTwoDFiles(position).Cam2AvgV_rms = newV2rms;

    StitchedTwoDFiles(position).AR = TwoDFilesArray(i).AR;
    StitchedTwoDFiles(position).FlowRate = TwoDFilesArray(i).FlowRate;

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

    StitchedTwoDFiles2(position).AvgV = stitchVel;
    StitchedTwoDFiles2(position).AvgV_rms = stitchVelrms;

    StitchedTwoDFiles2(position).Cam1AvgV = newV1;
    StitchedTwoDFiles2(position).Cam1AvgV_rms = newV1rms;

    StitchedTwoDFiles2(position).Cam2AvgV = newV2;
    StitchedTwoDFiles2(position).Cam2AvgV_rms = newV2rms;

    StitchedTwoDFiles2(position).AR = TwoDFilesArray(i).AR;
    StitchedTwoDFiles2(position).FlowRate = TwoDFilesArray(i).FlowRate;

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

    StitchedTwoDFiles3(position).AvgV = stitchVel;
    StitchedTwoDFiles3(position).AvgV_rms = stitchVelrms;

    StitchedTwoDFiles3(position).Cam1AvgV = newV1;
    StitchedTwoDFiles3(position).Cam1AvgV_rms = newV1rms;

    StitchedTwoDFiles3(position).Cam2AvgV = newV2;
    StitchedTwoDFiles3(position).Cam2AvgV_rms = newV2rms;

    StitchedTwoDFiles3(position).AR = TwoDFilesArray(i).AR;
    StitchedTwoDFiles3(position).FlowRate = TwoDFilesArray(i).FlowRate;

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

```

```
StitchedTwoDFiles4(position).AvgV = stitchVel;  
StitchedTwoDFiles4(position).AvgV_rms = stitchVelrms;  
  
StitchedTwoDFiles4(position).Cam1AvgV = newV1;  
StitchedTwoDFiles4(position).Cam1AvgV_rms = newV1rms;  
  
StitchedTwoDFiles4(position).Cam2AvgV = newV2;  
StitchedTwoDFiles4(position).Cam2AvgV_rms = newV2rms;  
  
StitchedTwoDFiles4(position).AR = TwoDFilesArray(i).AR;  
StitchedTwoDFiles4(position).FlowRate = TwoDFilesArray(i).FlowRate;  
end  
  
% Saving all four structures  
save('W:\Carlos Mendez\Matlab Results Folder  
\Stitched2DFiles', 'StitchedTwoDFiles');  
save('W:\Carlos Mendez\Matlab Results Folder  
\Stitched2DFilesCL', 'StitchedTwoDFiles2');  
save('W:\Carlos Mendez\Matlab Results Folder  
\Stitched2DFilesYhlf', 'StitchedTwoDFiles3');  
save('W:\Carlos Mendez\Matlab Results Folder  
\Stitched2DFilesTurb', 'StitchedTwoDFiles4');  
end
```

Published with MATLAB® R2019a

```

function [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] =
    stitcher2(V1,V1rms,V2,V2rms,num)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% stitcher2 This is a function to stitch together the near exit flow
% field and the downstream flow field.
%
%   stitcher2(V1,V1rms,V2,V2rms) takes the upstream velocity field
%   data (V1 & V1rms) along with the downstream velocity field
%   data (V2 & V2rms) and stitches them according to the procedure
%   described in the thesis. The outputs are the stitched velocity
%   fields, along with the transformed initial velocity fields
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vrms1 = V1rms;
Vave2 = V2;
Vrms2 = V2rms;

% First, the CLs of both fields are found, and straightened if needed.
% The data is also centered about the CLs. Camera 2's data is flipped
% as it was located on the other side of the laser sheet.
[Vave1,Vrms1] = TwoDCL(Vave1,Vrms1);
Vave2 = flipf(Vave2,'x');
Vrms2 = flipf(Vrms2,'x');

% Here the downstream vector field is interpolated so that it has the
% same spatial resolution as the upstream vector field.
[Vave2,Vrms2] = TwoDCL(Vave2,Vrms2);
[Vave2,Vrms2] = TwoDInterpField(Vave1,Vave2,Vrms2);
[Vave2,Vrms2] = TwoDCL(Vave2,Vrms2);

% The location downstream where camera 1's data is cut-off is found.
if num == 1
    [ib1,ib12] = TwoDGetCutOffs2(Vave1,Vave2,Vrms1,Vrms2);
elseif num == 2
    [ib1,ib12] = TwoDGetCutOffs3(Vave1,Vave2,Vrms1,Vrms2);
elseif num == 3
    [ib1,ib12] = TwoDGetCutOffs4(Vave1,Vave2,Vrms1,Vrms2);
elseif num == 4
    [ib1,ib12] = TwoDGetCutOffs5(Vave1,Vave2,Vrms1,Vrms2);
end

% Both vector fields are trimmed where they should be.
[Vave1,Vrms1] = TwoDCrop1(Vave1,Vrms1,ib1);
[Vave2,Vrms2] = TwoDCrop2(Vave2,Vrms2,ib12);

% Here the structure array for the stiched velocity fields are made by
% copying the structure arrays of the downstream velocity fields

```

```

stitchVel = Vave2;
stitchVelrms = Vrms2;
size1 = size(Vave1.vy);
size2 = size(Vave2.vy);

% Making new Y vector array.
delY = Vave1.y(1,2) - Vave1.y(1,1);
downstreamY = zeros(1,size2(1,2));
downstreamY(1,1) = Vave1.y(1,end)+delY;
for i = 2:size2(1,2)
    downstreamY(1,i) = downstreamY(1,i-1)+delY;
end
newY = horzcat(Vave1.y,downstreamY);
stitchVel.y = newY;
stitchVelrms.y = newY;

% Making new vx array. First, the data from camera 1 is padded with
% zeros so it is the same width as the data from camera 2.
EmptySpace = size2(1,1) - size1(1,1);
ZeroPad = zeros(EmptySpace/2,size1(1,2));
newVave1_vx = vertcat(ZeroPad,Vave1.vx,ZeroPad);
newVrms1_vx = vertcat(ZeroPad,Vrms1.vx,ZeroPad);
stitched_Vave_vx = horzcat(newVave1_vx,Vave2.vx);
stitched_Vrms_vx = horzcat(newVrms1_vx,Vrms2.vx);
stitchVel.vx = stitched_Vave_vx;
stitchVelrms.vx = stitched_Vrms_vx;

% Making new vy array. First, the data from camera 1 is padded with
% zeros so it is the same width as the data from camera 2.
newVave1_vy = vertcat(ZeroPad,Vave1.vy,ZeroPad);
newVrms1_vy = vertcat(ZeroPad,Vrms1.vy,ZeroPad);
stitched_Vave_vy = horzcat(newVave1_vy,Vave2.vy);
stitched_Vrms_vy = horzcat(newVrms1_vy,Vrms2.vy);
stitchVel.vy = stitched_Vave_vy;
stitchVelrms.vy = stitched_Vrms_vy;

% Saving the new velocity fields to the output variables.
newV1 = Vave1;
newV1rms = Vrms1;
newV2 = Vave2;
newV2rms = Vrms2;
end

```

Published with MATLAB® R2019a

```

function [new_Vave,newV_rms] = TwoDCL(V_ave,V_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDCL This is a function to find the centerline of a 2D-PIV
% velocity field and center the data.
%
% TwoDCL(V_ave,V_rms) takes the velocity field data (V_ave,V_rms)
% and finds the centerline, straightens it, and centers it
% according to the procedure described in the thesis. The outputs
% are the transformed velocity fields.
%
% Who : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Vave = V_ave;
Vrms = V_rms;

% Here, the location of the highest velocity in the X-direction is
% determined using the maxCLsearch function, and its location
% within the velocity field is then determined.
Umax = maxCLsearch(Vave.vy);
[vmax_y,vmax_x] = find(Vave.vy == Umax);

% Here a center line row vector is made and the location of the
% highest velocity, which should be on the CL is determined and
% stored.
Y = Vave.x; % In physical coordinates, x array is in Y-direction
X = Vave.y; % In physical coordinates, y array is in X-direction
rc = size(V_ave.vy);
CL_matrix = zeros(1,rc(1,2));
for i = 1:rc(1,2)
    U_ave_vec = Vave.vy(1:end,i);
    U_ave_vec_max = max(max(U_ave_vec));
    loc = find(U_ave_vec == U_ave_vec_max);
    CL_matrix(1,i) = Y(1,loc(1,1));
end

% Outliers are looked for a deleted. This should be near the edges of
% the velocity vector field.
TF = isoutlier(CL_matrix);
rc2 = size(TF);
DeleteArray = zeros(1,1);
j = 1;

for i = 1:rc2(1,2)
    criteria_check = TF(1,i);
    if criteria_check == 1
        DeleteArray(j) = i;
        j = j + 1;
    end
end
end

```

```

% Deleting any columns that shouldnt be within the array
if DeleteArray(1,1) > 0
    revDeleteArray = fliplr(DeleteArray);
    rc3 = size(DeleteArray);
    for k = 1:rc3(1,2)
        CL_matrix(:,revDeleteArray(k)) = [];
        X(:,revDeleteArray(k)) = [];
    end
end

% Here a line of best fit is made for the centerline points
last_x = X(1,end);
Fit = polyfit(X,CL_matrix,1);
last_y = (Fit(1,1)*last_x) + Fit(1,2);

% Here the first and last X position is are compared to see if the
% line is straight. If not, the angle of rotation is determined
% and the centerline is rotated. The fields are centered and saved
% to the new field as output.
if last_y ~= Y(1,vmax_y)
    delta_x = abs(last_x-0);
    delta_y = (Y(1,vmax_y)-last_y);
    div = delta_y/delta_x;
    angle = atan(div);
    NewVave = rotatef(Vave,angle,X(1,vmax_x),Y(1,vmax_y));
    NewVrms = rotatef(Vrms,angle,X(1,vmax_x),Y(1,vmax_y));

    rc = size(NewVave.vy);
    newCL_matrix = zeros(1,rc(1,2));
    for i = 1:rc(1,2)
        U_ave_vec = NewVave.vy(1:end,i);
        U_ave_vec_max = max(max(U_ave_vec));
        loc = find(U_ave_vec == U_ave_vec_max);
        newCL_matrix(1,i) = NewVave.x(1,loc(1,1));
    end

    zeroloc = mode(newCL_matrix);
    new_Vave = NewVave;
    newV_rms = NewVrms;
    new_Vave.x = NewVave.x - zeroloc;
    newV_rms.x = NewVrms.x - zeroloc;

elseif last_y == Y(1,vmax_y)
    zeroloc = mode(CL_matrix);
    new_Vave = Vave;
    newV_rms = Vrms;
    new_Vave.x = Vave.x - zeroloc;
    newV_rms.x = Vrms.x - zeroloc;
end

end

```

Published with MATLAB® R2019a

```

function [maxVel] = maxCLsearch(VecField)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% maxCLsearch Function for finding the maximum velocity value
% in a vector field
%
%   maxCLsearch(VecField) %% Function for finding the maximum
%   velocity value in a vector field. This is for 2D data flowing
%   from bottom to top. Only the first quarter of data is used
%   as this is where the highest velocity should be located.
%
% Who   : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% A new vector field made only of the first quarter of the original
% vector field is made, as the maximum velocity should be here.
VF = VecField;
[r,c] = size(VF);
newC = ceil(c/4);
VF2 = VF(1:r,1:newC);

% From the 2D data, the 10 highest values in each column are
% found. This is essentially looking at each velocity profile
% downstream and getting the 10 highest values.
TopTenRows = maxk(VF2,10);

% Matrix array of size (10x5) is made.
TopFive = zeros(10,5);

% The the 5 highest values in each row is found. This way most
% of the values in the potential core region should be obtained
for i = 1:10
    CurTopTenRow = TopTenRows(i,:);
    TopFive(i,:) = maxk(CurTopTenRow,5);
end

% Reshaping the matrix into a (1x50) array so that only outliers can
% be removed. It was in matrix form, the entire column containing the
% outlier would be removed. Matlab removes any value above 3 standard
% deviations of the mean.
TopFive = reshape(TopFive,[1,50]);
OutRemTopFive = rmoutliers(TopFive);

% With the outliers removed, the maximum value is taken, as this
% should be the actual maximum value.
maxVel = max(OutRemTopFive);

end

```

Published with MATLAB® R2019a

```

function [newVField2,newVField2_rms] =
    TwoDInterpField(VField1,VField2,VField2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDInterpField Function that interpolates the larger downstream
% vector field so that it has the same spatial resolution as the
% smaller upstream vector field
%
%   TwoDInterpField(VField1,VField2,VField2_rms) takes the upstream
%   velocity field data (VField1), determines its resolution and
%   then makes new downstream fields for the old downstream data
%   (V2 & V2rms) to be put into.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Vec_ave1 is the vector field with the smaller grid resolution.
Vec_ave1 = VField1;
Vec_ave2 = VField2;
Vec_rms2 = VField2_rms;

% Creating new structures by copying old structures. The X,
% Y, U, and V values will be replaced accordingly.
newVec_ave2 = Vec_ave2;
newVec_rms2 = Vec_rms2;

% Determining current X and Y resolutions for each data set.
delX = diff(Vec_ave1.x(1,1:2));
delY = diff(Vec_ave1.y(1,1:2));
delX2 = diff(Vec_ave2.x(1,1:2));
delY2 = diff(Vec_ave2.y(1,1:2));

% Making X and Y arrays that have the same size as the vector
% field, and making gridded interpolant objects for the velocity
% data so the new data fields may be made using the objects.
[X,Y] = ndgrid(Vec_ave2.x(1,1):delX2:Vec_ave2.x(1,end),...
    Vec_ave2.y(1,1):delY2:Vec_ave2.y(1,end));
vx = griddedInterpolant(X,Y,Vec_ave2.vx,'cubic');
vy = griddedInterpolant(X,Y,Vec_ave2.vy,'cubic');
vx_rms = griddedInterpolant(X,Y,Vec_rms2.vx,'cubic');
vy_rms = griddedInterpolant(X,Y,Vec_rms2.vy,'cubic');

% Making new vector arrays with new gridding with proper resolution.
[X2,Y2] = ndgrid(Vec_ave2.x(1,1):delX:Vec_ave2.x(1,end),...
    Vec_ave2.y(1,1):delY:Vec_ave2.y(1,end));
newVx = vx(X2,Y2);
newVy = vy(X2,Y2);
newVx_rms = vx_rms(X2,Y2);
newVy_rms = vy_rms(X2,Y2);

% Updating structures with proper array values.
newVec_ave2.x = X2(1:end,1).';

```

```
newVec_ave2.y = Y2(1,1:end);
newVec_ave2.vx = newVx;
newVec_ave2.vy = newVy;

newVec_rms2.x = X2(1:end,1) .';
newVec_rms2.y = Y2(1,1:end);
newVec_rms2.vx = newVx_rms;
newVec_rms2.vy = newVy_rms;

% Saving results to outputs
newVField2 = newVec_ave2;
newVField2_rms = newVec_rms2;

end
```

Published with MATLAB® R2019a

```

function [ib4,ib44] = TwoDGetCutOffs2(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs2 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs2(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between decay values is determined in the overlap region
% and its location is taken as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.
pointsleft1 = sizeVel1(1,2) - ib1;
DS1 = Vave1.vy(r1,ib1:end);

```

```

DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then the location of where the velocity field needs to be cut-off
% is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1 = DS1(1,1:ib2);
DS2 = DS2(1,1:ib2);

% Here each point within DS1 is compared to DS2 and the absolute
% difference between them is determined.
DiffDS = zeros(ib2,ib2);
for i = 1:ib2
    DiffDS(:,i) = abs(DS2 - DS1(1,i));
end
overallDiff = DiffDS;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;
ib44 = c2 + ib33 - 1;
end

```

```

function [ib4,ib44] = TwoDGetCutOffs3(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs3 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs3(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between half width values is determined in the overlap region
% and its location is taken as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);
sizeVel2 = size(Vave2.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.
pointsleft1 = sizeVel1(1,2) - ib1;

```

```

DS1 = Vave1.vy(r1,ib1:end);
DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then where the velocity field needs to be cut-off is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1 = DS1(1,1:ib2);
DS2 = DS2(1,1:ib2);

% Here the velocity half-widths are found and stored in 2 arrays.
velhlf1 = zeros(2,ib2);
velhlf2 = zeros(2,ib2);
for i = 1:ib2
    currentVel1 = DS1(1,i);
    currentVel2 = DS2(1,i);
    [r3,c3] = find(Vave1.vy == currentVel1);
    [r4,c4] = find(Vave2.vy == currentVel2);
    velVec1 = Vave1.vy(:,c3);
    velVec2 = Vave2.vy(:,c4);

    % Getting half width location in negative direction for DS1
    for j = (r3-1):-1:1
        criteria = velVec1(j,1)/currentVel1;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave1.x(1,j);
            velhlf1(1,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave1.x(1,j-1);

```

```

        hlf_phys_loc2 = Vave1.x(1,j);
        V_1 = velVec1(j-1,1)/currentVel1;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf1(1,i) = interp_hlf;
        break
    end
end

% Getting half width location in positive direction for DS1
for j = (r3+1):sizeVel1(1,1)
    criteria = velVec1(j,1)/currentVel1;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave1.x(1,j);
        velhlf1(2,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave1.x(1,j-1);
        hlf_phys_loc2 = Vave1.x(1,j);
        V_1 = velVec1(j-1,1)/currentVel1;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf1(2,i) = interp_hlf;
        break
    end
end

% Getting half width location in negative direction for DS2
for j = (r4-1):-1:1
    criteria = velVec2(j,1)/currentVel2;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave2.x(1,j);
        velhlf2(1,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave2.x(1,j-1);
        hlf_phys_loc2 = Vave2.x(1,j);
        V_1 = velVec2(j-1,1)/currentVel2;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf2(1,i) = interp_hlf;
        break
    end
end
end

```

```

    % Getting half width location in positive direction for DS2
    for j = (r4+1):sizeVel2(1,1)
        criteria = velVec2(j,1)/currentVel2;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave2.x(1,j);
            velhlf2(2,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave2.x(1,j-1);
            hlf_phys_loc2 = Vave2.x(1,j);
            V_1 = velVec2(j-1,1)/currentVel2;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf2(2,i) = interp_hlf;
            break
        end
    end
end

% Here the velocity half widths of DS1 are compared to the
% velocity half widths of DS2. The absolute differences between
% them is determined and stored within two arrays.
DiffVelhlf1 = zeros(ib2,ib2);
DiffVelhlf2 = zeros(ib2,ib2);
for i = 1:ib2
    DiffVelhlf1(:,i) = abs(velhlf2(1,:) - velhlf1(1,i));
    DiffVelhlf2(:,i) = abs(velhlf2(2,:) - velhlf1(2,i));
end

% The average of the velocity half widths differences at each
% location is determined.
aveDiffVelhlf = (DiffVelhlf1+DiffVelhlf2)/2;
overallDiff = aveDiffVelhlf;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;
ib44 = c2 + ib33 - 1;
end

```

```

function [ib4,ib44] = TwoDGetCutOffs4(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs4 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs4(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between turbulence values is determined in the overlap region
% and its location is taken as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;
Vrms1 = V1_rms;
Vrms2 = V2_rms;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.

```

```

pointsleft1 = sizeVel1(1,2) - ib1;
DS1 = Vave1.vy(r1,ib1:end);
DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));
DS1rms1 = Vrms1.vy(r1,ib1:end);
DS2rms2 = Vrms2.vy(r2,c2:(c2+pointsleft1));
DS1rms = DS1rms1./DS1;
DS2rms = DS2rms2./DS2;

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then where the velocity field needs to be cut-off is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1rms = DS1rms(1,1:ib2);
DS2rms = DS2rms(1,1:ib2);

% Here each point within DS1rms is compared to DS2rms
% and the absolute difference between them is determined.
DiffDSrms = zeros(ib2,ib2);
for i = 1:ib2
    DiffDSrms(:,i) = abs(DS2rms - DS1rms(1,i));
end
overallDiff = DiffDSrms;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;

```

```
ib44 = c2 + ib33 - 1;  
end
```

Published with MATLAB® R2019a

```

function [ib4,ib44] = TwoDGetCutOffs5(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs5 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs5(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between the decay, half width, and turbulence values is
% determined in the overlap region and its location is taken
% as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;
Vrms1 = V1_rms;
Vrms2 = V2_rms;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);
sizeVel2 = size(Vave2.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end
end

```

```

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.
pointsleft1 = sizeVell(1,2) - ib1;
DS1 = Vave1.vy(r1,ib1:end);
DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));
DS1rms1 = Vrms1.vy(r1,ib1:end);
DS2rms2 = Vrms2.vy(r2,c2:(c2+pointsleft1));
DS1rms = DS1rms1./DS1;
DS2rms = DS2rms2./DS2;

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then where the velocity field needs to be cut-off is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1 = DS1(1,1:ib2);
DS2 = DS2(1,1:ib2);
DS1rms = DS1rms(1,1:ib2);
DS2rms = DS2rms(1,1:ib2);

% Here each point within DS1 is compared to DS2 and the absolute
% difference between them is determined. The same is done for
% DS1rms and DS2rms.
DiffDS = zeros(ib2,ib2);
DiffDSrms = zeros(ib2,ib2);
for i = 1:ib2
    DiffDS(:,i) = abs(DS2 - DS1(1,i));
    DiffDSrms(:,i) = abs(DS2rms - DS1rms(1,i));
end

% Here the velocity half-widths are found and stored in 2 arrays.
velhlf1 = zeros(2,ib2);
velhlf2 = zeros(2,ib2);

```

```

for i = 1:ib2
    currentVel1 = DS1(1,i);
    currentVel2 = DS2(1,i);
    [r3,c3] = find(Vave1.vy == currentVel1);
    [r4,c4] = find(Vave2.vy == currentVel2);
    velVec1 = Vave1.vy(:,c3);
    velVec2 = Vave2.vy(:,c4);

    % Getting half width location in negative direction for DS1
    for j = (r3-1):-1:1
        criteria = velVec1(j,1)/currentVel1;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave1.x(1,j);
            velhlf1(1,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave1.x(1,j-1);
            hlf_phys_loc2 = Vave1.x(1,j);
            V_1 = velVec1(j-1,1)/currentVel1;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf1(1,i) = interp_hlf;
            break
        end
    end
end

% Getting half width location in positive direction for DS1
for j = (r3+1):sizeVel1(1,1)
    criteria = velVec1(j,1)/currentVel1;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave1.x(1,j);
        velhlf1(2,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave1.x(1,j-1);
        hlf_phys_loc2 = Vave1.x(1,j);
        V_1 = velVec1(j-1,1)/currentVel1;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf1(2,i) = interp_hlf;
        break
    end
end

% Getting half width location in negative direction for DS2
for j = (r4-1):-1:1

```

```

        criteria = velVec2(j,1)/currentVel2;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave2.x(1,j);
            velhlf2(1,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave2.x(1,j-1);
            hlf_phys_loc2 = Vave2.x(1,j);
            V_1 = velVec2(j-1,1)/currentVel2;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf2(1,i) = interp_hlf;
            break
        end
    end
end

% Getting half width location in positive direction for DS2
for j = (r4+1):sizeVel2(1,1)
    criteria = velVec2(j,1)/currentVel2;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave2.x(1,j);
        velhlf2(2,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave2.x(1,j-1);
        hlf_phys_loc2 = Vave2.x(1,j);
        V_1 = velVec2(j-1,1)/currentVel2;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf2(2,i) = interp_hlf;
        break
    end
end
end

% The absolute differences between the half widths is determined
% and stored within two arrays. The average of the two is then
% determined at each location.
DiffVelhlf1 = zeros(ib2,ib2);
DiffVelhlf2 = zeros(ib2,ib2);
for i = 1:ib2
    DiffVelhlf1(:,i) = abs(velhlf2(1,:) - velhlf1(1,i));
    DiffVelhlf2(:,i) = abs(velhlf2(2,:) - velhlf1(2,i));
end
end

```

```
aveDiffVelhlf = (DiffVelhlf1+DiffVelhlf2)/2;

% The overall difference in terms of decay, velocity half width
% and turbulence is determined.
overallDiff = (DiffDS + aveDiffVelhlf+DiffDSrms)/3;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;
ib44 = c2 + ib33 - 1;
end
```

Published with MATLAB® R2019a

```

function [newVecAve,NewVecRMS] = TwoDCrop1(VecAve,VecRMS,ib2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDCrop1 Function to crop the upstream vector field
%
%   TwoDCrop1(VecAve,VecRMS,ib2) takes the upstream velocity
%   field data (VecAve,VecRMS), crops the edge near the nozzle
%   exit and on the sides parallel to the centerline. The end
%   of the jet is cropped using ib2.
%
% Who   : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave = VecAve;
Vrms = VecRMS;
cutoff = ib2;

% Cropping off excess points where vec2 will be stitched onto.
rc = size(Vave.vy);
for i = rc(1,2):-1:(cutoff+1)
    Vave.vx(:,i) = [];
    Vave.vy(:,i) = [];
    Vave.y(:,i) = [];
    Vrms.vx(:,i) = [];
    Vrms.vy(:,i) = [];
    Vrms.y(:,i) = [];
end

% Finding the closest data point to zero. The mask was
% extended to be slightly before the nozzle exit in an
% to help with processing.
rc = size(Vave.vy);
for i = 1:rc(1,2)
    criteria = Vave.y(1,i);
    if criteria < 0
        continue
    elseif criteria == 0
        zero_loc = i;
        break
    elseif criteria > 0
        zero_loc = i;
        break
    end
end

% Setting the zero value point
zeroval = Vave.y(1,zero_loc);

% Deleting any before the zero value point location
for i = (zero_loc-1):-1:1

```

```

    Vave.vx(:,i) = [];
    Vave.vy(:,i) = [];
    Vave.y(:,i) = [];

    Vrms.vx(:,i) = [];
    Vrms.vy(:,i) = [];
    Vrms.y(:,i) = [];
end

% Positioning the vector fields in the X-direction wrt to
% the zero location.
Vave.y = Vave.y - zeroval;
Vrms.y = Vrms.y - zeroval;

% Cropping the sides parallel to the jet flow. 5% of the
% shortest distance from the centerline to the edges is
% cut off so that the jet is centered.

% Determining shortest distance between the two values.
rc2 = find(Vave.x == 0);
dist1 = Vave.x(1,rc2) - Vave.x(1,1);
dist2 = Vave.x(1,rc(1,1)) - Vave.x(1,rc2);
if dist1 < dist2
    dist = dist1;
elseif dist1 > dist2
    dist = dist2;
end

% Getting the five percent distance in terms of mm.
FivePercDist = dist*0.05;

% Determining where in the Y array on the right side the
% five percent distance is located.
criteria = dist - FivePercDist;
for i = 1:rc(1,1)
    check = Vave.x(1,i);
    if check < criteria
        continue
    elseif check == criteria
        cut_loc = i;
        break
    elseif check > criteria
        cut_loc = i;
        break
    end
end

% Cutting off the right side first as this wont affect
% the indices of the left side in doing so.
for i = rc(1,1):-1:cut_loc
    Vave.vx(i,:) = [];
    Vave.vy(i,:) = [];
    Vave.x(:,i) = [];

```

```

        Vrms.vx(i,:) = [];
        Vrms.vy(i,:) = [];
        Vrms.x(:,i) = [];
    end

    % Determining where in the Y array on the left side the
    % five percent distance is located.
    criteria = FivePercDist - dist;

    for i = 1:rc(1,1)
        check = Vave.x(1,i);
        if check < criteria
            continue
        elseif check == criteria
            cut_loc = i;
            break
        elseif check > criteria
            cut_loc = i-1;
            break
        end
    end

    % Cutting off the left side now.
    for i = cut_loc:-1:1
        Vave.vx(i,:) = [];
        Vave.vy(i,:) = [];
        Vave.x(:,i) = [];

        Vrms.vx(i,:) = [];
        Vrms.vy(i,:) = [];
        Vrms.x(:,i) = [];
    end

    % Saving the cut down vector fields to the output variables.
    newVecAve = Vave;
    NewVecRMS = Vrms;
end

```

Published with MATLAB® R2019a

```

function [newVecAve,NewVecRMS] = TwoDCrop2 (VecAve,VecRMS,ib22)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDCrop2 Function to crop the downstream vector field
%
%   TwoDCrop2(VecAve,VecRMS,ib22) takes the downstream velocity
%   field data (VecAve,VecRMS), crops the downstream edge and
%   the sides parallel to the centerline. The edge near the slot
%   exit is cropped using ib22.
%
% Who   : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave = VecAve;
Vrms = VecRMS;
cutoff = ib22;

% Cropping off excess points where vec1 will be stitched onto.
zeroval = Vave.y(1,cutoff);
for i = (cutoff-1):-1:1
    Vave.vx(:,i) = [];
    Vave.vy(:,i) = [];
    Vave.y(:,i) = [];
    Vrms.vx(:,i) = [];
    Vrms.vy(:,i) = [];
    Vrms.y(:,i) = [];
end

% Positioning the vector fields in the X-direction wrt to
% the zero location.
Vave.y = Vave.y - zeroval;
Vrms.y = Vrms.y - zeroval;
rc = size(Vave.vy);

% Cropping the last 6 points in the data. Visually, these
% points have been found to be outliers in data sets.
Vave.vx(:, (rc(1,2)-6):rc(1,2)) = [];
Vave.vy(:, (rc(1,2)-6):rc(1,2)) = [];
Vave.y(:, (rc(1,2)-6):rc(1,2)) = [];
Vrms.vx(:, (rc(1,2)-6):rc(1,2)) = [];
Vrms.vy(:, (rc(1,2)-6):rc(1,2)) = [];
Vrms.y(:, (rc(1,2)-6):rc(1,2)) = [];

% Cropping the sides parallel to the jet flow. 2.5% of the
% shortest distance from the centerline to the edges is
% cut off so that the jet is centered.

% Determining shortest distance between the two values.
rc2 = find(Vave.x == 0);
dist1 = Vave.x(1,rc2) - Vave.x(1,1);
dist2 = Vave.x(1,rc(1,1)) - Vave.x(1,rc2);

```

```

if dist1 < dist2
    dist = dist1;
elseif dist1 > dist2
    dist = dist2;
end

% Getting the two-point-five percent distance in terms of mm.
FivePercDist = dist*0.025;

% Determining where in the Y array on the right side the
% two-point-five percent distance is located.
criteria = dist - FivePercDist;
for i = 1:rc(1,1)
    check = Vave.x(1,i);
    if check < criteria
        continue
    elseif check == criteria
        cut_loc = i;
        break
    elseif check > criteria
        cut_loc = i;
        break
    end
end

% Cutting off the right side first as this wont affect
% the indices of the left side in doing so.
for i = rc(1,1):-1:cut_loc
    Vave.vx(i,:) = [];
    Vave.vy(i,:) = [];
    Vave.x(:,i) = [];
    Vrms.vx(i,:) = [];
    Vrms.vy(i,:) = [];
    Vrms.x(:,i) = [];
end

% Determining where in the Y array on the left side the
% two-point-five percent distance is located.
criteria = FivePercDist - dist;

for i = 1:rc(1,1)
    check = Vave.x(1,i);
    if check < criteria
        continue
    elseif check == criteria
        cut_loc = i;
        break
    elseif check > criteria
        cut_loc = i-1;
        break
    end
end

% Cutting off the left side now.

```

```
for i = cut_loc:-1:1
    Vave.vx(i,:) = [];
    Vave.vy(i,:) = [];
    Vave.x(:,i) = [];

    Vrms.vx(i,:) = [];
    Vrms.vy(i,:) = [];
    Vrms.x(:,i) = [];
end

% Saving the cut down vector fields to the output variables.
newVecAve = Vave;
NewVecRMS = Vrms;
end
```

Published with MATLAB® R2019a

```

function [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] =
    stitcher2(V1,V1rms,V2,V2rms,num)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% stitcher2 This is a function to stitch together the near exit flow
% field and the downstream flow field.
%
%   stitcher2(V1,V1rms,V2,V2rms) takes the upstream velocity field
%   data (V1 & V1rms) along with the downstream velocity field
%   data (V2 & V2rms) and stitches them according to the procedure
%   described in the thesis. The outputs are the stitched velocity
%   fields, along with the transformed initial velocity fields
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vrms1 = V1rms;
Vave2 = V2;
Vrms2 = V2rms;

% First, the CLs of both fields are found, and straightened if needed.
% The data is also centered about the CLs. Camera 2's data is flipped
% as it was located on the other side of the laser sheet.
[Vave1,Vrms1] = TwoDCL(Vave1,Vrms1);
Vave2 = flipf(Vave2,'x');
Vrms2 = flipf(Vrms2,'x');

% Here the downstream vector field is interpolated so that it has the
% same spatial resolution as the upstream vector field.
[Vave2,Vrms2] = TwoDCL(Vave2,Vrms2);
[Vave2,Vrms2] = TwoDInterpField(Vave1,Vave2,Vrms2);
[Vave2,Vrms2] = TwoDCL(Vave2,Vrms2);

% The location downstream where camera 1's data is cut-off is found.
if num == 1
    [ib1,ib12] = TwoDGetCutOffs2(Vave1,Vave2,Vrms1,Vrms2);
elseif num == 2
    [ib1,ib12] = TwoDGetCutOffs3(Vave1,Vave2,Vrms1,Vrms2);
elseif num == 3
    [ib1,ib12] = TwoDGetCutOffs4(Vave1,Vave2,Vrms1,Vrms2);
elseif num == 4
    [ib1,ib12] = TwoDGetCutOffs5(Vave1,Vave2,Vrms1,Vrms2);
end

% Both vector fields are trimmed where they should be.
[Vave1,Vrms1] = TwoDCrop1(Vave1,Vrms1,ib1);
[Vave2,Vrms2] = TwoDCrop2(Vave2,Vrms2,ib12);

% Here the structure array for the stiched velocity fields are made by
% copying the structure arrays of the downstream velocity fields

```

```

stitchVel = Vave2;
stitchVelrms = Vrms2;
size1 = size(Vave1.vy);
size2 = size(Vave2.vy);

% Making new Y vector array.
delY = Vave1.y(1,2) - Vave1.y(1,1);
downstreamY = zeros(1,size2(1,2));
downstreamY(1,1) = Vave1.y(1,end)+delY;
for i = 2:size2(1,2)
    downstreamY(1,i) = downstreamY(1,i-1)+delY;
end
newY = horzcat(Vave1.y,downstreamY);
stitchVel.y = newY;
stitchVelrms.y = newY;

% Making new vx array. First, the data from camera 1 is padded with
% zeros so it is the same width as the data from camera 2.
EmptySpace = size2(1,1) - size1(1,1);
ZeroPad = zeros(EmptySpace/2,size1(1,2));
newVave1_vx = vertcat(ZeroPad,Vave1.vx,ZeroPad);
newVrms1_vx = vertcat(ZeroPad,Vrms1.vx,ZeroPad);
stitched_Vave_vx = horzcat(newVave1_vx,Vave2.vx);
stitched_Vrms_vx = horzcat(newVrms1_vx,Vrms2.vx);
stitchVel.vx = stitched_Vave_vx;
stitchVelrms.vx = stitched_Vrms_vx;

% Making new vy array. First, the data from camera 1 is padded with
% zeros so it is the same width as the data from camera 2.
newVave1_vy = vertcat(ZeroPad,Vave1.vy,ZeroPad);
newVrms1_vy = vertcat(ZeroPad,Vrms1.vy,ZeroPad);
stitched_Vave_vy = horzcat(newVave1_vy,Vave2.vy);
stitched_Vrms_vy = horzcat(newVrms1_vy,Vrms2.vy);
stitchVel.vy = stitched_Vave_vy;
stitchVelrms.vy = stitched_Vrms_vy;

% Saving the new velocity fields to the output variables.
newV1 = Vave1;
newV1rms = Vrms1;
newV2 = Vave2;
newV2rms = Vrms2;
end

```

Published with MATLAB® R2019a

```

function [new_Vave,newV_rms] = TwoDCL(V_ave,V_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDCL This is a function to find the centerline of a 2D-PIV
% velocity field and center the data.
%
% TwoDCL(V_ave,V_rms) takes the velocity field data (V_ave,V_rms)
% and finds the centerline, straightens it, and centers it
% according to the procedure described in the thesis. The outputs
% are the transformed velocity fields.
%
% Who : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Vave = V_ave;
Vrms = V_rms;

% Here, the location of the highest velocity in the X-direction is
% determined using the maxCLsearch function, and its location
% within the velocity field is then determined.
Umax = maxCLsearch(Vave.vy);
[vmax_y,vmax_x] = find(Vave.vy == Umax);

% Here a center line row vector is made and the location of the
% highest velocity, which should be on the CL is determined and
% stored.
Y = Vave.x; % In physical coordinates, x array is in Y-direction
X = Vave.y; % In physical coordinates, y array is in X-direction
rc = size(V_ave.vy);
CL_matrix = zeros(1,rc(1,2));
for i = 1:rc(1,2)
    U_ave_vec = Vave.vy(1:end,i);
    U_ave_vec_max = max(max(U_ave_vec));
    loc = find(U_ave_vec == U_ave_vec_max);
    CL_matrix(1,i) = Y(1,loc(1,1));
end

% Outliers are looked for a deleted. This should be near the edges of
% the velocity vector field.
TF = isoutlier(CL_matrix);
rc2 = size(TF);
DeleteArray = zeros(1,1);
j = 1;

for i = 1:rc2(1,2)
    criteria_check = TF(1,i);
    if criteria_check == 1
        DeleteArray(j) = i;
        j = j + 1;
    end
end
end

```

```

% Deleting any columns that shouldnt be within the array
if DeleteArray(1,1) > 0
    revDeleteArray = fliplr(DeleteArray);
    rc3 = size(DeleteArray);
    for k = 1:rc3(1,2)
        CL_matrix(:,revDeleteArray(k)) = [];
        X(:,revDeleteArray(k)) = [];
    end
end

% Here a line of best fit is made for the centerline points
last_x = X(1,end);
Fit = polyfit(X,CL_matrix,1);
last_y = (Fit(1,1)*last_x) + Fit(1,2);

% Here the first and last X position is are compared to see if the
% line is straight. If not, the angle of rotation is determined
% and the centerline is rotated. The fields are centered and saved
% to the new field as output.
if last_y ~= Y(1,vmax_y)
    delta_x = abs(last_x-0);
    delta_y = (Y(1,vmax_y)-last_y);
    div = delta_y/delta_x;
    angle = atan(div);
    NewVave = rotatef(Vave,angle,X(1,vmax_x),Y(1,vmax_y));
    NewVrms = rotatef(Vrms,angle,X(1,vmax_x),Y(1,vmax_y));

    rc = size(NewVave.vy);
    newCL_matrix = zeros(1,rc(1,2));
    for i = 1:rc(1,2)
        U_ave_vec = NewVave.vy(1:end,i);
        U_ave_vec_max = max(max(U_ave_vec));
        loc = find(U_ave_vec == U_ave_vec_max);
        newCL_matrix(1,i) = NewVave.x(1,loc(1,1));
    end

    zeroloc = mode(newCL_matrix);
    new_Vave = NewVave;
    newV_rms = NewVrms;
    new_Vave.x = NewVave.x - zeroloc;
    newV_rms.x = NewVrms.x - zeroloc;

elseif last_y == Y(1,vmax_y)
    zeroloc = mode(CL_matrix);
    new_Vave = Vave;
    newV_rms = Vrms;
    new_Vave.x = Vave.x - zeroloc;
    newV_rms.x = Vrms.x - zeroloc;
end

end

```

Published with MATLAB® R2019a

```

function [newVecAve,NewVecRMS] = TwoDCrop1(VecAve,VecRMS,ib2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDCrop1 Function to crop the upstream vector field
%
%   TwoDCrop1(VecAve,VecRMS,ib2) takes the upstream velocity
%   field data (VecAve,VecRMS), crops the edge near the nozzle
%   exit and on the sides parallel to the centerline. The end
%   of the jet is cropped using ib2.
%
% Who   : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave = VecAve;
Vrms = VecRMS;
cutoff = ib2;

% Cropping off excess points where vec2 will be stitched onto.
rc = size(Vave.vy);
for i = rc(1,2):-1:(cutoff+1)
    Vave.vx(:,i) = [];
    Vave.vy(:,i) = [];
    Vave.y(:,i) = [];
    Vrms.vx(:,i) = [];
    Vrms.vy(:,i) = [];
    Vrms.y(:,i) = [];
end

% Finding the closest data point to zero. The mask was
% extended to be slightly before the nozzle exit in an
% to help with processing.
rc = size(Vave.vy);
for i = 1:rc(1,2)
    criteria = Vave.y(1,i);
    if criteria < 0
        continue
    elseif criteria == 0
        zero_loc = i;
        break
    elseif criteria > 0
        zero_loc = i;
        break
    end
end

% Setting the zero value point
zeroval = Vave.y(1,zero_loc);

% Deleting any before the zero value point location
for i = (zero_loc-1):-1:1

```

```

    Vave.vx(:,i) = [];
    Vave.vy(:,i) = [];
    Vave.y(:,i) = [];

    Vrms.vx(:,i) = [];
    Vrms.vy(:,i) = [];
    Vrms.y(:,i) = [];
end

% Positioning the vector fields in the X-direction wrt to
% the zero location.
Vave.y = Vave.y - zeroval;
Vrms.y = Vrms.y - zeroval;

% Cropping the sides parallel to the jet flow. 5% of the
% shortest distance from the centerline to the edges is
% cut off so that the jet is centered.

% Determining shortest distance between the two values.
rc2 = find(Vave.x == 0);
dist1 = Vave.x(1,rc2) - Vave.x(1,1);
dist2 = Vave.x(1,rc(1,1)) - Vave.x(1,rc2);
if dist1 < dist2
    dist = dist1;
elseif dist1 > dist2
    dist = dist2;
end

% Getting the five percent distance in terms of mm.
FivePercDist = dist*0.05;

% Determining where in the Y array on the right side the
% five percent distance is located.
criteria = dist - FivePercDist;
for i = 1:rc(1,1)
    check = Vave.x(1,i);
    if check < criteria
        continue
    elseif check == criteria
        cut_loc = i;
        break
    elseif check > criteria
        cut_loc = i;
        break
    end
end

% Cutting off the right side first as this wont affect
% the indices of the left side in doing so.
for i = rc(1,1):-1:cut_loc
    Vave.vx(i,:) = [];
    Vave.vy(i,:) = [];
    Vave.x(:,i) = [];

```

```

        Vrms.vx(i,:) = [];
        Vrms.vy(i,:) = [];
        Vrms.x(:,i) = [];
    end

    % Determining where in the Y array on the left side the
    % five percent distance is located.
    criteria = FivePercDist - dist;

    for i = 1:rc(1,1)
        check = Vave.x(1,i);
        if check < criteria
            continue
        elseif check == criteria
            cut_loc = i;
            break
        elseif check > criteria
            cut_loc = i-1;
            break
        end
    end

    % Cutting off the left side now.
    for i = cut_loc:-1:1
        Vave.vx(i,:) = [];
        Vave.vy(i,:) = [];
        Vave.x(:,i) = [];

        Vrms.vx(i,:) = [];
        Vrms.vy(i,:) = [];
        Vrms.x(:,i) = [];
    end

    % Saving the cut down vector fields to the output variables.
    newVecAve = Vave;
    NewVecRMS = Vrms;
end

```

Published with MATLAB® R2019a

```

function [newVecAve,NewVecRMS] = TwoDCrop2 (VecAve,VecRMS,ib22)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDCrop2 Function to crop the downstream vector field
%
% TwoDCrop2(VecAve,VecRMS,ib22) takes the downstream velocity
% field data (VecAve,VecRMS), crops the downstream edge and
% the sides parallel to the centerline. The edge near the slot
% exit is cropped using ib22.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave = VecAve;
Vrms = VecRMS;
cutoff = ib22;

% Cropping off excess points where vec1 will be stitched onto.
zeroval = Vave.y(1,cutoff);
for i = (cutoff-1):-1:1
    Vave.vx(:,i) = [];
    Vave.vy(:,i) = [];
    Vave.y(:,i) = [];
    Vrms.vx(:,i) = [];
    Vrms.vy(:,i) = [];
    Vrms.y(:,i) = [];
end

% Positioning the vector fields in the X-direction wrt to
% the zero location.
Vave.y = Vave.y - zeroval;
Vrms.y = Vrms.y - zeroval;
rc = size(Vave.vy);

% Cropping the last 6 points in the data. Visually, these
% points have been found to be outliers in data sets.
Vave.vx(:, (rc(1,2)-6):rc(1,2)) = [];
Vave.vy(:, (rc(1,2)-6):rc(1,2)) = [];
Vave.y(:, (rc(1,2)-6):rc(1,2)) = [];
Vrms.vx(:, (rc(1,2)-6):rc(1,2)) = [];
Vrms.vy(:, (rc(1,2)-6):rc(1,2)) = [];
Vrms.y(:, (rc(1,2)-6):rc(1,2)) = [];

% Cropping the sides parallel to the jet flow. 2.5% of the
% shortest distance from the centerline to the edges is
% cut off so that the jet is centered.

% Determining shortest distance between the two values.
rc2 = find(Vave.x == 0);
dist1 = Vave.x(1,rc2) - Vave.x(1,1);
dist2 = Vave.x(1,rc(1,1)) - Vave.x(1,rc2);

```

```

if dist1 < dist2
    dist = dist1;
elseif dist1 > dist2
    dist = dist2;
end

% Getting the two-point-five percent distance in terms of mm.
FivePercDist = dist*0.025;

% Determining where in the Y array on the right side the
% two-point-five percent distance is located.
criteria = dist - FivePercDist;
for i = 1:rc(1,1)
    check = Vave.x(1,i);
    if check < criteria
        continue
    elseif check == criteria
        cut_loc = i;
        break
    elseif check > criteria
        cut_loc = i;
        break
    end
end

% Cutting off the right side first as this wont affect
% the indices of the left side in doing so.
for i = rc(1,1):-1:cut_loc
    Vave.vx(i,:) = [];
    Vave.vy(i,:) = [];
    Vave.x(:,i) = [];
    Vrms.vx(i,:) = [];
    Vrms.vy(i,:) = [];
    Vrms.x(:,i) = [];
end

% Determining where in the Y array on the left side the
% two-point-five percent distance is located.
criteria = FivePercDist - dist;

for i = 1:rc(1,1)
    check = Vave.x(1,i);
    if check < criteria
        continue
    elseif check == criteria
        cut_loc = i;
        break
    elseif check > criteria
        cut_loc = i-1;
        break
    end
end

% Cutting off the left side now.

```

```
for i = cut_loc:-1:1
    Vave.vx(i,:) = [];
    Vave.vy(i,:) = [];
    Vave.x(:,i) = [];

    Vrms.vx(i,:) = [];
    Vrms.vy(i,:) = [];
    Vrms.x(:,i) = [];
end

% Saving the cut down vector fields to the output variables.
newVecAve = Vave;
NewVecRMS = Vrms;
end
```

Published with MATLAB® R2019a

```

function [ib4,ib44] = TwoDGetCutOffs2(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs2 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs2(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between decay values is determined in the overlap region
% and its location is taken as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.
pointsleft1 = sizeVel1(1,2) - ib1;
DS1 = Vave1.vy(r1,ib1:end);

```

```

DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then the location of where the velocity field needs to be cut-off
% is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1 = DS1(1,1:ib2);
DS2 = DS2(1,1:ib2);

% Here each point within DS1 is compared to DS2 and the absolute
% difference between them is determined.
DiffDS = zeros(ib2,ib2);
for i = 1:ib2
    DiffDS(:,i) = abs(DS2 - DS1(1,i));
end
overallDiff = DiffDS;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;
ib44 = c2 + ib33 - 1;
end

```

```

function [ib4,ib44] = TwoDGetCutOffs3(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs3 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs3(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between half width values is determined in the overlap region
% and its location is taken as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);
sizeVel2 = size(Vave2.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.
pointsleft1 = sizeVel1(1,2) - ib1;

```

```

DS1 = Vave1.vy(r1,ib1:end);
DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then where the velocity field needs to be cut-off is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1 = DS1(1,1:ib2);
DS2 = DS2(1,1:ib2);

% Here the velocity half-widths are found and stored in 2 arrays.
velhlf1 = zeros(2,ib2);
velhlf2 = zeros(2,ib2);
for i = 1:ib2
    currentVel1 = DS1(1,i);
    currentVel2 = DS2(1,i);
    [r3,c3] = find(Vave1.vy == currentVel1);
    [r4,c4] = find(Vave2.vy == currentVel2);
    velVec1 = Vave1.vy(:,c3);
    velVec2 = Vave2.vy(:,c4);

    % Getting half width location in negative direction for DS1
    for j = (r3-1):-1:1
        criteria = velVec1(j,1)/currentVel1;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave1.x(1,j);
            velhlf1(1,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave1.x(1,j-1);

```

```

        hlf_phys_loc2 = Vave1.x(1,j);
        V_1 = velVec1(j-1,1)/currentVel1;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf1(1,i) = interp_hlf;
        break
    end
end

% Getting half width location in positive direction for DS1
for j = (r3+1):sizeVel1(1,1)
    criteria = velVec1(j,1)/currentVel1;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave1.x(1,j);
        velhlf1(2,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave1.x(1,j-1);
        hlf_phys_loc2 = Vave1.x(1,j);
        V_1 = velVec1(j-1,1)/currentVel1;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf1(2,i) = interp_hlf;
        break
    end
end

% Getting half width location in negative direction for DS2
for j = (r4-1):-1:1
    criteria = velVec2(j,1)/currentVel2;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave2.x(1,j);
        velhlf2(1,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave2.x(1,j-1);
        hlf_phys_loc2 = Vave2.x(1,j);
        V_1 = velVec2(j-1,1)/currentVel2;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf2(1,i) = interp_hlf;
        break
    end
end
end

```

```

    % Getting half width location in positive direction for DS2
    for j = (r4+1):sizeVel2(1,1)
        criteria = velVec2(j,1)/currentVel2;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave2.x(1,j);
            velhlf2(2,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave2.x(1,j-1);
            hlf_phys_loc2 = Vave2.x(1,j);
            V_1 = velVec2(j-1,1)/currentVel2;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf2(2,i) = interp_hlf;
            break
        end
    end
end

% Here the velocity half widths of DS1 are compared to the
% velocity half widths of DS2. The absolute differences between
% them is determined and stored within two arrays.
DiffVelhlf1 = zeros(ib2,ib2);
DiffVelhlf2 = zeros(ib2,ib2);
for i = 1:ib2
    DiffVelhlf1(:,i) = abs(velhlf2(1,:) - velhlf1(1,i));
    DiffVelhlf2(:,i) = abs(velhlf2(2,:) - velhlf1(2,i));
end

% The average of the velocity half widths differences at each
% location is determined.
aveDiffVelhlf = (DiffVelhlf1+DiffVelhlf2)/2;
overallDiff = aveDiffVelhlf;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;
ib44 = c2 + ib33 - 1;
end

```

```

function [ib4,ib44] = TwoDGetCutOffs4(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs4 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs4(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between turbulence values is determined in the overlap region
% and its location is taken as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;
Vrms1 = V1_rms;
Vrms2 = V2_rms;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.

```

```

pointsleft1 = sizeVel1(1,2) - ib1;
DS1 = Vave1.vy(r1,ib1:end);
DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));
DS1rms1 = Vrms1.vy(r1,ib1:end);
DS2rms2 = Vrms2.vy(r2,c2:(c2+pointsleft1));
DS1rms = DS1rms1./DS1;
DS2rms = DS2rms2./DS2;

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then where the velocity field needs to be cut-off is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1rms = DS1rms(1,1:ib2);
DS2rms = DS2rms(1,1:ib2);

% Here each point within DS1rms is compared to DS2rms
% and the absolute difference between them is determined.
DiffDSrms = zeros(ib2,ib2);
for i = 1:ib2
    DiffDSrms(:,i) = abs(DS2rms - DS1rms(1,i));
end
overallDiff = DiffDSrms;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;

```

```
ib44 = c2 + ib33 - 1;  
end
```

Published with MATLAB® R2019a

```

function [ib4,ib44] = TwoDGetCutOffs5(V1,V2,V1_rms,V2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDGetCutOffs5 Function to get the locations of where the two
% velocity matrices should be cut-off for stitching. The best
% cut-off location is found in terms of velocity decay.
%
% TwoDGetCutOffs5(V1,V2,V1_rms,V2_rms) takes the upstream
% velocity field data (V1 & V1rms) along with the downstream
% velocity field data (V2 & V2rms), finds where they begin to
% overlap in terms of velocity values. The smallest difference
% between the decay, half width, and turbulence values is
% determined in the overlap region and its location is taken
% as the cut-off location.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
Vave1 = V1;
Vave2 = V2;
Vrms1 = V1_rms;
Vrms2 = V2_rms;

% Determining the maximum velocity in both velocity fields.
maxVel1 = maxCLsearch(Vave1.vy);
maxVel2 = maxCLsearch(Vave2.vy);

% Finding the locations of the maximum velocities in each
% velocity field and then obtaining each field's center line.
[r1,c1] = find(Vave1.vy == maxVel1);
CL1 = Vave1.vy(r1,:);
[r2,c2] = find(Vave2.vy == maxVel2);
sizeVel1 = size(Vave1.vy);
sizeVel2 = size(Vave2.vy);

% Finding out where the first data set starts to have overlap
% with the second data set. ib1 is the point right before this
% happens. This is accomplished by evaluating centerline values.
for i = c1:sizeVel1(1,2)
    criteria = CL1(1,i);
    if criteria > maxVel2
        continue
    elseif criteria == maxVel2
        ib1 = i-1;
        break
    elseif criteria < maxVel2
        ib1 = i-1;
        break
    end
end
end

```

```

% Making two arrays with the overlapping data points in the upstream
% and downstream arrays.
pointsleft1 = sizeVell(1,2) - ib1;
DS1 = Vave1.vy(r1,ib1:end);
DS2 = Vave2.vy(r2,c2:(c2+pointsleft1));
DS1rms1 = Vrms1.vy(r1,ib1:end);
DS2rms2 = Vrms2.vy(r2,c2:(c2+pointsleft1));
DS1rms = DS1rms1./DS1;
DS2rms = DS2rms2./DS2;

% The first data set should have points that are higher in value
% than the second one. If theres any data that need to be cut-off,
% it will be lower than the values in the second data set.
Diff = DS2 - DS1;
Diff2 = (Diff > 0);
Diff3 = find(Diff2);

% If this array is empty, then this means that all of the remaining
% data points from the first data set overlap with the data from
% the second data set, so it doesn't need to be cut-off. If not,
% then where the velocity field needs to be cut-off is determined.
if isempty(Diff3) == 1
    ib2 = pointsleft1 + 1;
elseif isempty(Diff3) == 0
    for i = (pointsleft1+1):-1:1
        criteria = Diff(1,i);
        if criteria > 0
            continue
        elseif criteria < 0
            ib2 = i;
            break
        end
    end
end
end

% The remaining values on the centerline are updated.
DS1 = DS1(1,1:ib2);
DS2 = DS2(1,1:ib2);
DS1rms = DS1rms(1,1:ib2);
DS2rms = DS2rms(1,1:ib2);

% Here each point within DS1 is compared to DS2 and the absolute
% difference between them is determined. The same is done for
% DS1rms and DS2rms.
DiffDS = zeros(ib2,ib2);
DiffDSrms = zeros(ib2,ib2);
for i = 1:ib2
    DiffDS(:,i) = abs(DS2 - DS1(1,i));
    DiffDSrms(:,i) = abs(DS2rms - DS1rms(1,i));
end

% Here the velocity half-widths are found and stored in 2 arrays.
velhlf1 = zeros(2,ib2);
velhlf2 = zeros(2,ib2);

```

```

for i = 1:ib2
    currentVel1 = DS1(1,i);
    currentVel2 = DS2(1,i);
    [r3,c3] = find(Vave1.vy == currentVel1);
    [r4,c4] = find(Vave2.vy == currentVel2);
    velVec1 = Vave1.vy(:,c3);
    velVec2 = Vave2.vy(:,c4);

    % Getting half width location in negative direction for DS1
    for j = (r3-1):-1:1
        criteria = velVec1(j,1)/currentVel1;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave1.x(1,j);
            velhlf1(1,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave1.x(1,j-1);
            hlf_phys_loc2 = Vave1.x(1,j);
            V_1 = velVec1(j-1,1)/currentVel1;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf1(1,i) = interp_hlf;
            break
        end
    end
end

    % Getting half width location in positive direction for DS1
    for j = (r3+1):sizeVel1(1,1)
        criteria = velVec1(j,1)/currentVel1;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave1.x(1,j);
            velhlf1(2,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave1.x(1,j-1);
            hlf_phys_loc2 = Vave1.x(1,j);
            V_1 = velVec1(j-1,1)/currentVel1;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf1(2,i) = interp_hlf;
            break
        end
    end
end

    % Getting half width location in negative direction for DS2
    for j = (r4-1):-1:1

```

```

        criteria = velVec2(j,1)/currentVel2;
        if criteria > 0.5
            continue
        elseif criteria == 0.5
            hlf_phys_loc = Vave2.x(1,j);
            velhlf2(1,i) = hlf_phys_loc;
            break
        elseif criteria < 0.5
            hlf_phys_loc1 = Vave2.x(1,j-1);
            hlf_phys_loc2 = Vave2.x(1,j);
            V_1 = velVec2(j-1,1)/currentVel2;
            interp_ratio = (0.5 - V_1)/(criteria - V_1);
            interp_hlf = hlf_phys_loc1 + ...
                (interp_ratio * (hlf_phys_loc2 - ...
                    hlf_phys_loc1));
            velhlf2(1,i) = interp_hlf;
            break
        end
    end
end

% Getting half width location in positive direction for DS2
for j = (r4+1):sizeVel2(1,1)
    criteria = velVec2(j,1)/currentVel2;
    if criteria > 0.5
        continue
    elseif criteria == 0.5
        hlf_phys_loc = Vave2.x(1,j);
        velhlf2(2,i) = hlf_phys_loc;
        break
    elseif criteria < 0.5
        hlf_phys_loc1 = Vave2.x(1,j-1);
        hlf_phys_loc2 = Vave2.x(1,j);
        V_1 = velVec2(j-1,1)/currentVel2;
        interp_ratio = (0.5 - V_1)/(criteria - V_1);
        interp_hlf = hlf_phys_loc1 + ...
            (interp_ratio * (hlf_phys_loc2 - ...
                hlf_phys_loc1));
        velhlf2(2,i) = interp_hlf;
        break
    end
end
end

% The absolute differences between the half widths is determined
% and stored within two arrays. The average of the two is then
% determined at each location.
DiffVelhlf1 = zeros(ib2,ib2);
DiffVelhlf2 = zeros(ib2,ib2);
for i = 1:ib2
    DiffVelhlf1(:,i) = abs(velhlf2(1,:) - velhlf1(1,i));
    DiffVelhlf2(:,i) = abs(velhlf2(2,:) - velhlf1(2,i));
end
end

```

```
aveDiffVelhlf = (DiffVelhlf1+DiffVelhlf2)/2;

% The overall difference in terms of decay, velocity half width
% and turbulence is determined.
overallDiff = (DiffDS + aveDiffVelhlf+DiffDSrms)/3;

% The minimum value location between the data sets is determined
% and then its location is determined.
[minoverallDiffM1,minoverallDiffI1] = min(overallDiff);
[minoverallDiffM2,minoverallDiffI2] = min(minoverallDiffM1);

% The cut-off locations are saved to the output variables.
ib3 = minoverallDiffI2;
ib33 = minoverallDiffI1(minoverallDiffI2);
ib4 = ib1 + ib3 - 1;
ib44 = c2 + ib33 - 1;
end
```

Published with MATLAB® R2019a

```

function [newVField2,newVField2_rms] =
    TwoDInterpField(VField1,VField2,VField2_rms)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDInterpField Function that interpolates the larger downstream
% vector field so that it has the same spatial resolution as the
% smaller upstream vector field
%
%   TwoDInterpField(VField1,VField2,VField2_rms) takes the upstream
%   velocity field data (VField1), determines its resolution and
%   then makes new downstream fields for the old downstream data
%   (V2 & V2rms) to be put into.
%
% Who   : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Vec_ave1 is the vector field with the smaller grid resolution.
Vec_ave1 = VField1;
Vec_ave2 = VField2;
Vec_rms2 = VField2_rms;

% Creating new structures by copying old structures. The X,
% Y, U, and V values will be replaced accordingly.
newVec_ave2 = Vec_ave2;
newVec_rms2 = Vec_rms2;

% Determining current X and Y resolutions for each data set.
delX = diff(Vec_ave1.x(1,1:2));
delY = diff(Vec_ave1.y(1,1:2));
delX2 = diff(Vec_ave2.x(1,1:2));
delY2 = diff(Vec_ave2.y(1,1:2));

% Making X and Y arrays that have the same size as the vector
% field, and making gridded interpolant objects for the velocity
% data so the new data fields may be made using the objects.
[X,Y] = ndgrid(Vec_ave2.x(1,1):delX2:Vec_ave2.x(1,end),...
    Vec_ave2.y(1,1):delY2:Vec_ave2.y(1,end));
vx = griddedInterpolant(X,Y,Vec_ave2.vx,'cubic');
vy = griddedInterpolant(X,Y,Vec_ave2.vy,'cubic');
vx_rms = griddedInterpolant(X,Y,Vec_rms2.vx,'cubic');
vy_rms = griddedInterpolant(X,Y,Vec_rms2.vy,'cubic');

% Making new vector arrays with new gridding with proper resolution.
[X2,Y2] = ndgrid(Vec_ave2.x(1,1):delX:Vec_ave2.x(1,end),...
    Vec_ave2.y(1,1):delY:Vec_ave2.y(1,end));
newVx = vx(X2,Y2);
newVy = vy(X2,Y2);
newVx_rms = vx_rms(X2,Y2);
newVy_rms = vy_rms(X2,Y2);

% Updating structures with proper array values.
newVec_ave2.x = X2(1:end,1).';

```

```
newVec_ave2.y = Y2(1,1:end);
newVec_ave2.vx = newVx;
newVec_ave2.vy = newVy;

newVec_rms2.x = X2(1:end,1) .';
newVec_rms2.y = Y2(1,1:end);
newVec_rms2.vx = newVx_rms;
newVec_rms2.vy = newVy_rms;

% Saving results to outputs
newVField2 = newVec_ave2;
newVField2_rms = newVec_rms2;

end
```

Published with MATLAB® R2019a

```

function workonTwoDrawfiles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% workonTwoDrawfiles stitches all of the 2D-PIV data
%
% workonTwoDrawfiles takes the upstream velocity field
% data (V1 & V1rms) along with the downstream velocity field
% data (V2 & V2rms) and sends them to be stitched them using the
% stitcher2 function. The outputs from the function are saved to
% the new structures.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading the 2D-PIV velocity structures.
load('W:\Carlos Mendez\Matlab Results Folder
\2DFiles','TwoDFilesArray');

% Making structures for the stitched PIV data to be saved to. Note:
% different methods of stitching were used, so multiple result
% structures were made, according to the method used. The stitching
% involved stitching with respect to decay, half-width, turbulence
% intensity, and all three values (discussed in the thesis).
StitchedTwoDFiles(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

StitchedTwoDFiles2(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

StitchedTwoDFiles3(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

StitchedTwoDFiles4(24,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR',
[], 'FlowRate', [], ...
    'Cam1AvgV', [], 'Cam1AvgV_rms', [], 'Cam2AvgV', [], 'Cam2AvgV_rms', []);

% Sending the 2D-PIV data to be stitched using the stitcher2 function
% and saving the output to the corresponding structure.
% StitchedTwoDFiles - stitched wrt to all three variables;
% StitchedTwoDFiles2 - stitched wrt to decay
% StitchedTwoDFiles2 - stitched wrt to velocity half width
% StitchedTwoDFiles3 - stitched wrt to velocity turbulence intensity
for i = 1:2:47
    cam1_VEC_temp = TwoDFilesArray(i).AvgV;
    cam2_VEC_temp = TwoDFilesArray(i+1).AvgV;

    cam1_VEC_temp_rms = TwoDFilesArray(i).AvgV_rms;
    cam2_VEC_temp_rms = TwoDFilesArray(i+1).AvgV_rms;

```

```

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

    position = 0.5*(i+1);

    StitchedTwoDFiles(position).AvgV = stitchVel;
    StitchedTwoDFiles(position).AvgV_rms = stitchVelrms;

    StitchedTwoDFiles(position).Cam1AvgV = newV1;
    StitchedTwoDFiles(position).Cam1AvgV_rms = newV1rms;

    StitchedTwoDFiles(position).Cam2AvgV = newV2;
    StitchedTwoDFiles(position).Cam2AvgV_rms = newV2rms;

    StitchedTwoDFiles(position).AR = TwoDFilesArray(i).AR;
    StitchedTwoDFiles(position).FlowRate = TwoDFilesArray(i).FlowRate;

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

    StitchedTwoDFiles2(position).AvgV = stitchVel;
    StitchedTwoDFiles2(position).AvgV_rms = stitchVelrms;

    StitchedTwoDFiles2(position).Cam1AvgV = newV1;
    StitchedTwoDFiles2(position).Cam1AvgV_rms = newV1rms;

    StitchedTwoDFiles2(position).Cam2AvgV = newV2;
    StitchedTwoDFiles2(position).Cam2AvgV_rms = newV2rms;

    StitchedTwoDFiles2(position).AR = TwoDFilesArray(i).AR;
    StitchedTwoDFiles2(position).FlowRate = TwoDFilesArray(i).FlowRate;

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

    StitchedTwoDFiles3(position).AvgV = stitchVel;
    StitchedTwoDFiles3(position).AvgV_rms = stitchVelrms;

    StitchedTwoDFiles3(position).Cam1AvgV = newV1;
    StitchedTwoDFiles3(position).Cam1AvgV_rms = newV1rms;

    StitchedTwoDFiles3(position).Cam2AvgV = newV2;
    StitchedTwoDFiles3(position).Cam2AvgV_rms = newV2rms;

    StitchedTwoDFiles3(position).AR = TwoDFilesArray(i).AR;
    StitchedTwoDFiles3(position).FlowRate = TwoDFilesArray(i).FlowRate;

    [newV1,newV1rms,newV2,newV2rms, stitchVel, stitchVelrms] = ...

stitcher2(cam1_VEC_temp,cam1_VEC_temp_rms,cam2_VEC_temp,cam2_VEC_temp.

```

```
StitchedTwoDFiles4(position).AvgV = stitchVel;
StitchedTwoDFiles4(position).AvgV_rms = stitchVelrms;

StitchedTwoDFiles4(position).Cam1AvgV = newV1;
StitchedTwoDFiles4(position).Cam1AvgV_rms = newV1rms;

StitchedTwoDFiles4(position).Cam2AvgV = newV2;
StitchedTwoDFiles4(position).Cam2AvgV_rms = newV2rms;

StitchedTwoDFiles4(position).AR = TwoDFilesArray(i).AR;
StitchedTwoDFiles4(position).FlowRate = TwoDFilesArray(i).FlowRate;
end

% Saving all four structures
save('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFiles', 'StitchedTwoDFiles');
save('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFilesCL', 'StitchedTwoDFiles2');
save('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFilesYhlf', 'StitchedTwoDFiles3');
save('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFilesTurb', 'StitchedTwoDFiles4');
end
```

Published with MATLAB® R2019a

```

function MakeTwoDVectorFields
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MakeTwoDVectorFields This is a function to make whole-field plots
% for the 2D-PIV data.
%
% MakeTwoDVectorFields loads in the 2D-PIV data, and plots the
% velocity vector field, colormap, and colormap with the vector
% field. The resulting images are then saved to their folders.
%
% Who   : Carlos Mendez
% Where: University of Alberta
% When: Fall 2019
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading in the data
load('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFiles','StitchedTwoDFiles');
load('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFilesCL','StitchedTwoDFiles2');
load('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFilesYhlf','StitchedTwoDFiles3');
load('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFilesTurb','StitchedTwoDFiles4');
De = 2*sqrt(12/pi);

for i = 1:24
    % Loading in the vector arrays from each folder.
    currentVecArray1 = StitchedTwoDFiles(i).AvgV;
    currentVecArray2 = StitchedTwoDFiles2(i).AvgV;
    currentVecArray3 = StitchedTwoDFiles3(i).AvgV;
    currentVecArray4 = StitchedTwoDFiles4(i).AvgV;

    % Cutting down the vector field if its longer than 7De.
    Xarray1 = currentVecArray1.x/De;
    lastX1 = Xarray1(1,end);

    if lastX1 > 7
        [~,I] = min(abs(abs(Xarray1)-7));
        currentVecArray1.x = currentVecArray1.x(1,1:I);
        currentVecArray1.vx = currentVecArray1.vx(1:I,:);
        currentVecArray1.vy = currentVecArray1.vy(1:I,:);
    end

    Xarray2 = currentVecArray2.x/De;
    lastX2 = Xarray2(1,end);

    if lastX2 > 7
        [~,I] = min(abs(abs(Xarray2)-7));
        currentVecArray2.x = currentVecArray2.x(1,1:I);
        currentVecArray2.vx = currentVecArray2.vx(1:I,:);
        currentVecArray2.vy = currentVecArray2.vy(1:I,:);

```

```

end

Xarray3 = currentVecArray3.x/De;
lastX3 = Xarray3(1,end);

if lastX3 > 7
    [~,I] = min(abs(abs(Xarray3)-7));
    currentVecArray3.x = currentVecArray3.x(1,1:I);
    currentVecArray3.vx = currentVecArray3.vx(1:I,:);
    currentVecArray3.vy = currentVecArray3.vy(1:I,:);
end

Xarray4 = currentVecArray4.x/De;
lastX4 = Xarray4(1,end);

if lastX4 > 7
    [~,I] = min(abs(abs(Xarray4)-7));
    currentVecArray4.x = currentVecArray4.x(1,1:I);
    currentVecArray4.vx = currentVecArray4.vx(1:I,:);
    currentVecArray4.vy = currentVecArray4.vy(1:I,:);
end

% Making three plots - velocity field, velocity field with arrows,
% and arrow field for each data scenario (only one scenario
actually
% used).

VectorMapandArrows2(currentVecArray1);

experiment_string1 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Emf Version\' ...
    StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];
experiment_string2 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Eps Version\' ...
    StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];
experiment_string3 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Mat Version\' ...
    StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];
experiment_string4 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Png Version\' ...
    StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];

savename1 = [experiment_string1 '_VelField11'];
savename2 = [experiment_string2 '_VelField11'];
savename3 = [experiment_string3 '_VelField11'];
savename4 = [experiment_string4 '_VelField11'];

saveas(figure(1),savename1,'emf');
saveas(figure(1),savename2,'eps');
saveas(figure(1),savename3,'m');
saveas(figure(1),savename4,'png');

savename1 = [experiment_string1 '_VelField12'];
savename2 = [experiment_string2 '_VelField12'];

```

```

savename3 = [experiment_string3 '_VelField12'];
savename4 = [experiment_string4 '_VelField12'];

saveas(figure(2),savename1,'emf');
saveas(figure(2),savename2,'eps');
saveas(figure(2),savename3,'m');
saveas(figure(2),savename4,'png');

savename1 = [experiment_string1 '_VelField13'];
savename2 = [experiment_string2 '_VelField13'];
savename3 = [experiment_string3 '_VelField13'];
savename4 = [experiment_string4 '_VelField13'];

saveas(figure(3),savename1,'emf');
saveas(figure(3),savename2,'eps');
saveas(figure(3),savename3,'m');
saveas(figure(3),savename4,'png');

close all

VectorMapandArrows2(currentVecArray2);
experiment_string1 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Emf Version\' ...
    StitchedTwoDFiles2(i).AR StitchedTwoDFiles2(i).FlowRate];
experiment_string2 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Eps Version\' ...
    StitchedTwoDFiles2(i).AR StitchedTwoDFiles2(i).FlowRate];
experiment_string3 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Mat Version\' ...
    StitchedTwoDFiles2(i).AR StitchedTwoDFiles2(i).FlowRate];
experiment_string4 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Png Version\' ...
    StitchedTwoDFiles2(i).AR StitchedTwoDFiles2(i).FlowRate];

savename1 = [experiment_string1 '_VelField21'];
savename2 = [experiment_string2 '_VelField21'];
savename3 = [experiment_string3 '_VelField21'];
savename4 = [experiment_string4 '_VelField21'];

saveas(figure(1),savename1,'emf');
saveas(figure(1),savename2,'eps');
saveas(figure(1),savename3,'m');
saveas(figure(1),savename4,'png');

savename1 = [experiment_string1 '_VelField22'];
savename2 = [experiment_string2 '_VelField22'];
savename3 = [experiment_string3 '_VelField22'];
savename4 = [experiment_string4 '_VelField22'];

saveas(figure(2),savename1,'emf');
saveas(figure(2),savename2,'eps');
saveas(figure(2),savename3,'m');
saveas(figure(2),savename4,'png');

```

```

savename1 = [experiment_string1 '_VelField23'];
savename2 = [experiment_string2 '_VelField23'];
savename3 = [experiment_string3 '_VelField23'];
savename4 = [experiment_string4 '_VelField23'];

saveas (figure (3), savename1, 'emf');
saveas (figure (3), savename2, 'eps');
saveas (figure (3), savename3, 'm');
saveas (figure (3), savename4, 'png');

close all

VectorMapandArrows2 (currentVecArray3);
experiment_string1 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Emf Version\' ...
    StitchedTwoDFiles3(i).AR StitchedTwoDFiles3(i).FlowRate];
experiment_string2 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Eps Version\' ...
    StitchedTwoDFiles3(i).AR StitchedTwoDFiles3(i).FlowRate];
experiment_string3 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Mat Version\' ...
    StitchedTwoDFiles3(i).AR StitchedTwoDFiles3(i).FlowRate];
experiment_string4 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Png Version\' ...
    StitchedTwoDFiles3(i).AR StitchedTwoDFiles3(i).FlowRate];

savename1 = [experiment_string1 '_VelField31'];
savename2 = [experiment_string2 '_VelField31'];
savename3 = [experiment_string3 '_VelField31'];
savename4 = [experiment_string4 '_VelField31'];

saveas (figure (1), savename1, 'emf');
saveas (figure (1), savename2, 'eps');
saveas (figure (1), savename3, 'm');
saveas (figure (1), savename4, 'png');

savename1 = [experiment_string1 '_VelField32'];
savename2 = [experiment_string2 '_VelField32'];
savename3 = [experiment_string3 '_VelField32'];
savename4 = [experiment_string4 '_VelField32'];

saveas (figure (2), savename1, 'emf');
saveas (figure (2), savename2, 'eps');
saveas (figure (2), savename3, 'm');
saveas (figure (2), savename4, 'png');

savename1 = [experiment_string1 '_VelField33'];
savename2 = [experiment_string2 '_VelField33'];
savename3 = [experiment_string3 '_VelField33'];
savename4 = [experiment_string4 '_VelField33'];

saveas (figure (3), savename1, 'emf');
saveas (figure (3), savename2, 'eps');
saveas (figure (3), savename3, 'm');

```

```

saveas (figure (3) , savename4 , 'png' );

close all

VectorMapandArrows2 (currentVecArray4) ;
experiment_string1 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Emf Version\' ...
    StitchedTwoDFiles4(i).AR StitchedTwoDFiles4(i).FlowRate] ;
experiment_string2 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Eps Version\' ...
    StitchedTwoDFiles4(i).AR StitchedTwoDFiles4(i).FlowRate] ;
experiment_string3 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Mat Version\' ...
    StitchedTwoDFiles4(i).AR StitchedTwoDFiles4(i).FlowRate] ;
experiment_string4 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Vector Field Plots\Png Version\' ...
    StitchedTwoDFiles4(i).AR StitchedTwoDFiles4(i).FlowRate] ;

savename1 = [experiment_string1 '_VelField41'] ;
savename2 = [experiment_string2 '_VelField41'] ;
savename3 = [experiment_string3 '_VelField41'] ;
savename4 = [experiment_string4 '_VelField41'] ;

saveas (figure (1) , savename1 , 'emf' ) ;
saveas (figure (1) , savename2 , 'eps' ) ;
saveas (figure (1) , savename3 , 'm' ) ;
saveas (figure (1) , savename4 , 'png' ) ;

savename1 = [experiment_string1 '_VelField42'] ;
savename2 = [experiment_string2 '_VelField42'] ;
savename3 = [experiment_string3 '_VelField42'] ;
savename4 = [experiment_string4 '_VelField42'] ;

saveas (figure (2) , savename1 , 'emf' ) ;
saveas (figure (2) , savename2 , 'eps' ) ;
saveas (figure (2) , savename3 , 'm' ) ;
saveas (figure (2) , savename4 , 'png' ) ;

savename1 = [experiment_string1 '_VelField43'] ;
savename2 = [experiment_string2 '_VelField43'] ;
savename3 = [experiment_string3 '_VelField43'] ;
savename4 = [experiment_string4 '_VelField43'] ;

saveas (figure (3) , savename1 , 'emf' ) ;
saveas (figure (3) , savename2 , 'eps' ) ;
saveas (figure (3) , savename3 , 'm' ) ;
saveas (figure (3) , savename4 , 'png' ) ;

close all

end

```

end

Published with MATLAB® R2019a

```

function TwoDPlots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TwoDPlots This is a function to make X-Y plots.
%
% TwoDPlots loads in the 2D-PIV data, and plots the velocity decay,
% turbulence intensity, and lateral velocity half width. These
% individual plots are then saved in various formats.
%
% Who    : Carlos Mendez
% Where: University of Alberta
% When:  Fall 2019
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading in the data
load('W:\Carlos Mendez\Matlab Results Folder
\Stitched2DFiles','StitchedTwoDFiles');
for i = 1:24
    currentVave = StitchedTwoDFiles(i).AvgV;
    currentVave_rms = StitchedTwoDFiles(i).AvgV_rms;

    % getting data values
    De = 2*sqrt(12/pi);
    [r,c] = find(currentVave.x == 0);
    CL = currentVave.vy(c,:);
    CL_rms = currentVave_rms.vy(c,:);
    maxvel = maxCLsearch(currentVave.vy);
    nondimCL = CL/maxvel;
    nondimCL_rms = CL_rms./CL;
    nondimX = currentVave.y/De;

    % prepping save location string
    experiment_string1 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Individual Line Plots\Emf Version\' ...
        StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];
    experiment_string2 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Individual Line Plots\Mat Version\' ...
        StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];
    experiment_string3 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Individual Line Plots\Png Version\' ...
        StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];
    experiment_string4 = ['W:\Carlos Mendez\Matlab Results Folder\2D
Individual Line Plots\Eps Version\' ...
        StitchedTwoDFiles(i).AR StitchedTwoDFiles(i).FlowRate];

    % Plotting data and saving to results folder %

    % Plotting Centerline Decay %
    f = figure(1);
    f.Position = ([14,374,1893,623]);
    ax = gca;
    hold on

```

```

scatter(nondimX(1:5:end),nondimCL(1:5:end),'k');
ax.XScale = 'log';
ax.YScale = 'log';
xlim([10^-1 10]);
ylim([10^-1 1.1]);
ax.XLabel.String = 'X/D_e';
ax.XLabel.FontAngle = 'italic';
ax.YLabel.String = 'U_{cl}/U_{0}';
ax.YLabel.FontAngle = 'italic';
ax.FontName = 'Times New Roman';
ax.FontSize = 12;
hold off
savename1 = [experiment_string1 '_UclU0'];
savename2 = [experiment_string2 '_UclU0'];
savename3 = [experiment_string3 '_UclU0'];
savename4 = [experiment_string4 '_UclU0'];

saveas(gcf,savename1,'emf');
saveas(gcf,savename2,'m');
saveas(gcf,savename3,'png');
saveas(gcf,savename4,'eps');

[nondimVel_hlf,nondimVel_hlf2] = TwoDHalfWidth2(currentVave);

nondimVel_hlf_2 =
[nondimVel_hlf(1:5:end),nondimVel_hlf2(1:5:end)];

nondimX_2 = [nondimX(1:5:end),nondimX(1:5:end)];

% Plotting Velocity Half-Width %
f = figure(2);
f.Position = ([14,374,1893,623]);
ax = gca;
hold on
scatter(nondimX_2,nondimVel_hlf_2,'k');
xlim([0 8.25]);
ylim([-1.5 1.5]);
ax.XLabel.String = 'X/D_e';
ax.XLabel.FontAngle = 'italic';
ax.YLabel.String = 'Y_{1/2}/D_e';
ax.YLabel.FontAngle = 'italic';
ax.FontName = 'Times New Roman';
ax.FontSize = 12;
hold off

savename1 = [experiment_string1 '_YYhlf'];
savename2 = [experiment_string2 '_YYhlf'];
savename3 = [experiment_string3 '_YYhlf'];
savename4 = [experiment_string4 '_YYhlf'];
saveas(gcf,savename1,'emf');
saveas(gcf,savename2,'m');
saveas(gcf,savename3,'png');
saveas(gcf,savename4,'eps');

```

```

% Plotting streamwise turbulence intensity %
f = figure(3);
f.Position = ([14,374,1893,623]);
ax = gca;
hold on
scatter(nondimX(1:5:end),nondimCL_rms(1:5:end),'k');
xlim([0 8.25]);
ylim([0 0.4]);
ax.XLabel.String = 'X/D_e';
ax.XLabel.FontAngle = 'italic';
ax.YLabel.Interpreter = 'Latex';
ax.YLabel.String = '$\sqrt{\overline{u'^2}}/U_{cl}$';
ax.FontName = 'Times New Roman';
ax.FontSize = 12;
hold off

savename1 = [experiment_string1 '_uUcl'];
savename2 = [experiment_string2 '_uUcl'];
savename3 = [experiment_string3 '_uUcl'];
savename4 = [experiment_string4 '_uUcl'];
saveas(gcf,savename1,'emf');
saveas(gcf,savename2,'m');
saveas(gcf,savename3,'png');
saveas(gcf,savename4,'eps');

close all
end
end

```

Published with MATLAB® R2019a

C MATLAB Codes for Stereo-PIV Post-Processing

This section contains the MATLAB codes that were developed for further post-processing of the stereo-PIV data. Specifically, the codes include those that were used to locate the jet boundary, and produce the figures presented within this study.

```

function getstereofiles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% getstereofiles Extract stereo-PIV files from their locations and
% save to structure array.
%
% getstereofiles works to get the average velocity and average rms
% velocity results files for each aspect ratio, plane, and flow rate
% for the stereo-PIV experiments. The results are loaded into a
% structure array, which has the following fields: AvgV, AvgV_rms,
% AR, FlowRate, PlaneLoc, and DataOrigin.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Getting overall results folder. Location is saved as output
% variable, if needed later. Directory listing structure is made,
% and the first two rows are deleted as they are non-physical
% listings.
OverallDataFolderloc = uigetdir;
listing_info = dir(OverallDataFolderloc);
listing_info(~[listing_info.isdir]) = []; %get rid of non folders
listing_info(ismember({listing_info.name}, {'.', '..'})) = []; %get
% rid of . and ..
subdir_names = fullfile(OverallDataFolderloc, {listing_info.name});
%construct list of names
ARFolderNames = subdir_names.';

FilesArray(72,1) = struct('AvgV', [], 'AvgV_rms', [], 'AR', [], 'Plane',
[], ...
'FlowRate', [], 'DataOrigin', []);

% Loop to go through overall results folder, get files and save
% them in FilesArray output variable.

structloc = 1;
% Going through AR & Planes Folder.
for i = 1:12
    string = ARFolderNames(i);
    string = num2str(cell2mat(string));
    ARFolderName = string;
    ARFlisting_info = dir(ARFolderName);
    ARFlisting_info(~[ARFlisting_info.isdir]) = []; %get rid of non
% folders
    ARFlisting_info(ismember({ARFlisting_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    ARFsubdir_names = fullfile(ARFolderName, {ARFlisting_info.name});
    %construct list of names
    FlowFolderNames = ARFsubdir_names(1,1:6).'; %first 6 folders are
% flow folders

```

```

% Going through Flow Folders.
for j = 1:6
    string2 = FlowFolderNames(j);
    string2 = num2str(cell2mat(string2));
    FlowFolderName = string2;

    FFlisting_info = dir(FlowFolderName);
    FFlisting_info(~[FFlisting_info.isdir]) = []; %get rid of non
folders
    FFlisting_info(ismember({FFlisting_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    FFsubdir_names = fullfile(FlowFolderName,
{FFlisting_info.name}); %construct list of names
    ResultsFolder = FFsubdir_names(1,1).';

    RFlisting_info = dir(cell2mat(ResultsFolder));
    RFlisting_info(~[RFlisting_info.isdir]) = []; %get rid of non
folders
    RFlisting_info(ismember({RFlisting_info.name}, {'.', '..'})) =
[]; %get rid of . and ..
    RFsubdir_names = fullfile(ResultsFolder,
{RFlisting_info.name}); %construct list of names
    AveVelResultsFolder = RFsubdir_names(1,1).';
    AveVelResultsFolder = cell2mat(AveVelResultsFolder);

    % Storing the AvgV, AvgV_rms, AR, FlowRate, PlaneLoc, and
% DataOrigin in the FileArray structure in their
% respective fields.
    AvgVelString = [AveVelResultsFolder '\B00001_AvgV.vc7'];
    FileArray(structloc,1).AvgV = loadvec(AvgVelString);
    AvgVelRMSString = [AveVelResultsFolder '\B00002_StdevV.vc7'];
    FileArray(structloc,1).AvgV_rms = loadvec(AvgVelRMSString);
    FileArray(structloc,1).DataOrigin = AveVelResultsFolder;

    revFFN = fliplr(FlowFolderName);
    [token,remain] = strtok(revFFN,'\');
    FlowRate = fliplr(token);

    [token2,remain2] = strtok(remain,'\');
    intermstring = fliplr(token2);

    [newStr,matches] = split(intermstring,'-');
    AR = num2str(cell2mat(newStr(2)));
    Plane = num2str(cell2mat(newStr(3)));

    FileArray(structloc,1).AR = strtrim(AR);
    FileArray(structloc,1).Plane = strtrim(Plane);
    FileArray(structloc,1).FlowRate = FlowRate;

    structloc = structloc + 1;
end
end

% Save output structure

```

```
save('W:\Carlos Mendez\Matlab Results Folder  
\StereoFiles', 'FilesArray');  
end
```

Published with MATLAB® R2019a

```

function [JetArea,BPs,NewVecField,NewrmsVecField] =
    InterArea2(VecField,...
        rmsVecField)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% InterArea2 centers the stereo-PIV velocity field, determines
% its boundary points based on a 5% criteria, and calculates its
% area based on the boundary points via MATLAB's polyarea function
%
% InterArea2(VecField,rmsVecField) takes the velocity field
% data(AvgV and AvgV_rms), finds its centerline, centers it,
% finds its boundary points, and then uses MATLAB's polyarea
% function to get the jet area.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity fields to new variables
VEC_temp = VecField;
VEC_temp_rms = rmsVecField;
BC = 0.05;

% Locating the jet centerline, and centering the data sets.
[VEC_temp,AOR] = RotateStereo(VEC_temp);
VEC_temp_rms = rotatef(VEC_temp_rms,-AOR);
[r,c] = size(VEC_temp.vz);
X_vec = VEC_temp.x;

% Locating the approximate zero location in Z.
for i = 1:r
    criteria = X_vec(1,i);
    if criteria < 0
        continue
    elseif criteria == 0
        Zero_Location = i;
        break
    elseif criteria > 0
        prev_Zero_Location = X_vec(1,i-1);
        if abs(prev_Zero_Location) > criteria
            Zero_Location = i;
        elseif abs(prev_Zero_Location) < criteria
            Zero_Location = i-1;
        elseif abs(prev_Zero_Location) == criteria
            Zero_Location = i;
        end
        break
    end
end
end
Origin_Location_row_x = Zero_Location;
CL_crit_origin = X_vec(1,Origin_Location_row_x);
U = VEC_temp.vz;
Y_vec = VEC_temp.y;

```

```

% Locating zero location in Y.
maxvelCL = max(U(Origin_Location_row_x,:));
maxvelCL_Loc = find(U(Origin_Location_row_x,:) == maxvelCL);
CL_location = maxvelCL_Loc;
physical_CL_location = Y_vec(1,CL_location);
VEC_temp.y = VEC_temp.y - physical_CL_location;
Y_vec = VEC_temp.y;
[r4,c4] = find(Y_vec == 0);
Origin_Location_column_y = c4;

% Locating the 4 boundaries at the origin for the horizontal
% and vertical line tests.
Origin_Bound_Matrix = zeros(2,5);

% Locating the boundaries on the centerline.
[y_1,x_1,x_2,int1,int2] =
intercepts2(VEC_temp,Origin_Location_column_y,Origin_Location_row_x,B
Origin_Bound_Matrix(1,:) = [y_1 x_1 x_2 int1 int2];

% Locating the boundaries on the Y-axis.
[y_1,x_1,x_2,int1,int2] = intercepts2(VEC_temp,Origin_Location_row_x,
Origin_Location_column_y,BC,1);
Origin_Bound_Matrix(2,:) = [y_1 x_1 x_2 int1 int2];

% Locating all boundary points doing horizontal tests between the
% vertical boundaries that were just found.
no_horiz_test = Origin_Bound_Matrix(2,2) - Origin_Bound_Matrix(2,3)-2;
horiz_test_matrix = zeros(no_horiz_test, 5);
j = 0;
k = 1;
while j == 0
    first_criteria = Origin_Bound_Matrix(2,2)-1;
    % Going through all positive y-values
    for i = (1 + Origin_Location_column_y):first_criteria
        [y_1,x_1,x_2,int1,int2] =
intercepts2(VEC_temp,i,Origin_Location_row_x,BC,0);
        horiz_test_matrix(k,:) = [y_1 x_1 x_2 int1 int2];
        k = k + 1;
    end

    i = Origin_Location_column_y - 1;
    second_criteria = Origin_Bound_Matrix(2,3)+1;
    % Going through all negative y-values
    for i = (Origin_Location_column_y - 1):-1:second_criteria
        [y_1,x_1,x_2,int1,int2] =
intercepts2(VEC_temp,i,Origin_Location_row_x,BC,0);
        horiz_test_matrix(k,:) = [y_1 x_1 x_2 int1 int2];
        k = k + 1;
    end
    j = 1;
end

% Locating all boundary points doing vertical tests between the

```

```

% horizontal boundaries that were just found.
no_vert_test = Origin_Bound_Matrix(1,2) - Origin_Bound_Matrix(1,3)-2;
vert_test_matrix = zeros(no_vert_test, 5);
i = 1 + Origin_Location_row_x;
j = 0;
k = 1;
while j == 0
    first_criteria = Origin_Bound_Matrix(1,2)-1;
    % Going through all positive y-values
    for i = (1 + Origin_Location_row_x):first_criteria
        [y_1,x_1,x_2,int1,int2] =
intercepts2(VEC_temp,i,Origin_Location_column_y,BC,1);
        vert_test_matrix(k,:) = [y_1 x_1 x_2 int1 int2];
        k = k + 1;
    end

    second_criteria = Origin_Bound_Matrix(1,3)+1;
    % Going through all negative y-values
    for i = (Origin_Location_row_x - 1):-1:second_criteria
        [y_1,x_1,x_2,int1,int2] =
intercepts2(VEC_temp,i,Origin_Location_column_y,BC,1);
        vert_test_matrix(k,:) = [y_1 x_1 x_2 int1 int2];
        k = k + 1;
    end
    j = 1;
end

% Finding and removing all repeat values.
NewHorizMatrix = unique(horiz_test_matrix,'stable','rows');
NewVertMatrix = unique(vert_test_matrix,'stable','rows');

% Find and removing all false values.
[horiz_test_matrix, vert_test_matrix] = FalsePoints(NewHorizMatrix,
NewVertMatrix,Origin_Location_column_y,Origin_Location_row_x);

% Getting the physical values based on interpolation.
[PhysHorizArray, PhysVertArray] =
interpo(VEC_temp,horiz_test_matrix,vert_test_matrix,BC);

% Getting the physical values at the origin based on interpolation.
CL_array = Origin_Bound_Matrix(1,:);
Y_axis_array = Origin_Bound_Matrix(2,:);
[PhysCL_array, PhysY_axis_array] =
interpo(VEC_temp,CL_array,Y_axis_array,BC);

% Sorting the physical values so that they are in CW order starting
% from the positive Y-axis point.
[AllData,PolygonDataX,PolygonDataY] =
Sort2(PhysCL_array,PhysY_axis_array,PhysHorizArray,PhysVertArray);

% Centering the data for the mean velocity field and the rms velocity
% field about the origin.
VEC_temp_rms.x = VEC_temp_rms.x - CL_crit_origin;
VEC_temp_rms.y = VEC_temp_rms.y - physical_CL_location;

```

```

% Finding the approximate center in the x-data and centering the data
% about this point.
[r,c] = size(VEC_temp.x);
for i = 1:c
    criteria = X_vec(1,r);
    if criteria < 0
        continue
    elseif criteria == 0
        Zero_Location = r;
        break
    elseif criteria > 0
        prev_Zero_Location = X_vec(1,r-1);
        if abs(prev_Zero_Location) > criteria
            Zero_Location = r;
        elseif abs(prev_Zero_Location) < criteria
            Zero_Location = r-1;
        end
        break
    end
end
zero_val = VEC_temp.x(1,Zero_Location);
VEC_temp.x = VEC_temp.x - zero_val;
VEC_temp_rms.x = VEC_temp_rms.x - zero_val;
AllData(:,1) = AllData(:,1) - zero_val;
PolygonDataX = PolygonDataX - zero_val;

% Area is calculated and the data values are saved to
% the output variables.
JetArea = polyarea(PolygonDataX,PolygonDataY);
BPs = AllData;
NewVecField = VEC_temp;
NewrmsVecField = VEC_temp_rms;

% Result lengths are divided by the equivalent diameter for
% nondimensionalization. Results are saved to the outputs.
De = 2*sqrt(12/pi);

BPs(:,1) = BPs(:,1)/De;
BPs(:,2) = BPs(:,2)/De;

NewVecField.x = NewVecField.x/De;
NewVecField.y = NewVecField.y/De;

NewrmsVecField.x = NewrmsVecField.x/De;
NewrmsVecField.y = NewrmsVecField.y/De;
end

```

Published with MATLAB® R2019a

```

function [y1,x1,x2,int1,int2] =
    intercepts2(VecField,stationary,moving,BC,HorVer)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% intercepts2 determines where the boundary points criteria is
% fulfilled in the horizontal or vertical direction with respect
% to a specific starting point.
%
% intercepts2(VecField, stationary, moving, BC,HorVer) takes the
% velocity field data(VecField), finds the velocity column or row
% array at the stationary point and moves in the positive and
% negative direction until the boundary point condition is found.
% if interpolation is needed, then int1 and/or int2 are 1.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the input velocity field to a new variable
VEC_temp = VecField;

% Getting average velocity field, finding centerline velocity and
% dividing the field by it
U = VEC_temp.vz;
[r1,c1] = find(VEC_temp.x == 0);
[r2,c2] = find(VEC_temp.y == 0);
Umax = VEC_temp.vz(c1,c2);
V_dim = U./Umax;

% Getting velocity vector at corresponding Z or Y position
Stat_point = stationary;
if HorVer == 0
    V_dim_vec = V_dim(1:end,Stat_point);
elseif HorVer == 1
    V_dim_vec = V_dim(Stat_point,1:end);
    V_dim_vec = V_dim_vec.';
end

Mov_point = moving;
c = 0;

% Moving in the positive direction to determine where in
% the velocity vector the criteria is fulfilled. Results
% are saved to outputs.
while c == 0
    criteria_check = V_dim_vec(Mov_point,1);
    if criteria_check > BC
        Mov_point = Mov_point + 1;
    elseif criteria_check == BC
        c = 1;
        x1 = Mov_point;
        int1 = 0;
    end
end

```

```

    elseif criteria_check < BC
        c = 1;
        x1 = Mov_point;
        int1 = 1;
    end
end

c = 0;
Mov_point = moving;

% Moving in the negative direction to determine where in
% the velocity vector the criteria is fulfilled. Results
% are saved to outputs.
while c == 0
    criteria_check = V_dim_vec(Mov_point,1);
    if criteria_check > BC
        Mov_point = Mov_point - 1;
    elseif criteria_check == BC
        c = 1;
        x2 = Mov_point;
        int2 = 0;
    elseif criteria_check < BC
        c = 1;
        x2 = Mov_point;
        int2 = 1;
    end
end

y1 = Stat_point;
end

```

Published with MATLAB® R2019a

```

function [PhysHorizArray, PhysVertArray] =
    interpo(VecField, HorizArray, VertArray, BC)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% interpo interpolates the points retrieved from the horizontal
% and vertical tests to get the location of the boundary points.
%
% interpo(VecField, HorizArray, VertArray, BC) takes the points
% in from the tests (HorizArray, VertArray), converts them to
% their actual physical values using the vector field data
% (VecField) and interpolates in between points using a linear
% interpolation, if necessary. The physical values of the
% boundary points are the output (PhysHorizArray, PhysVertArray)
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the inputs velocity field and its data to new variables
VEC_temp = VecField;
X = VEC_temp.x.';
Y = VEC_temp.y;
U = VEC_temp.vz;
[r1,c1] = find(VEC_temp.x == 0);
[r2,c2] = find(VEC_temp.y == 0);
V_ave_max = VEC_temp.vz(c1,c2);
V_dim = U/V_ave_max;

% Going through the horizontal test array and converting points
% the points to their physical values. Interpolation is completed
% if it has been specified as being necessary. Results are
% saved to the output array.
[r,c] = size(HorizArray);
PhysHorizArray = zeros(r,c-2);
for i = 1:r
    y_value = HorizArray(i,1);
    y_phys_value = Y(1,y_value);
    PhysHorizArray(i,1) = y_phys_value;
    x_value = HorizArray(i,2);
    x_value2 = HorizArray(i,3);
    int1 = HorizArray(i,4);
    int2 = HorizArray(i,5);

    if (x_value == -1) && (x_value2 == -1)
        PhysHorizArray(i,2) = -1;
        PhysHorizArray(i,3) = -1;
        continue

    elseif (int1 == 0) && (int2 == 0)
        x_phys_value = X(x_value,1);
        PhysHorizArray(i,2) = x_phys_value;
        x_phys_value2 = X(x_value2,1);
        PhysHorizArray(i,3) = x_phys_value2;

```

```

        continue
    end
    if (x_value ~= -1) && (int1 == 1)
        x_1 = x_value - 1;
        x_phys_value = X(x_value,1);
        x_phys_value2 = X(x_1,1);
        v_2 = V_dim(x_value,y_value);
        v_1 = V_dim(x_1, y_value);
        inter_ratio = (BC - v_1)/(v_2 - v_1);
        x_interpol = x_phys_value2 + (inter_ratio * (x_phys_value -
x_phys_value2));
        PhysHorizArray(i,2) = x_interpol;
    end
    if (x_value2 ~= -1) && (int1 == 1)
        x_1 = x_value2 + 1;
        x_phys_value = X(x_value2,1);
        x_phys_value2 = X(x_1,1);
        v_2 = V_dim(x_value2,y_value);
        v_1 = V_dim(x_1, y_value);
        inter_ratio = (BC - v_1)/(v_2 - v_1);
        x_interpol = x_phys_value2 + (inter_ratio * (x_phys_value -
x_phys_value2));
        PhysHorizArray(i,3) = x_interpol;
    end
end
end

% Going through the vertical test array and converting points
% the points to their physical values. Interpolation is completed
% if it has been specified as being necessary. Results are
% saved to the output array.
[r,c] = size(VertArray);
PhysVertArray = zeros(r,c-2);
for i = 1:r
    x_value = VertArray(i,1);
    x_phys_value = X(x_value,1);
    PhysVertArray(i,1) = x_phys_value;
    y_value = VertArray(i,2);
    y_value2 = VertArray(i,3);
    int1 = VertArray(i,4);
    int2 = VertArray(i,5);

    if (y_value == -1) && (y_value2 == -1)
        PhysVertArray(i,2) = -1;
        PhysVertArray(i,3) = -1;
        continue

    elseif (int1 == 0) && (int2 == 0)
        y_phys_value = Y(1,y_value);
        PhysVertArray(i,2) = y_phys_value;
        y_phys_value2 = Y(1,y_value2);
        PhysVertArray(i,3) = y_phys_value2;
        continue
    end
end
if (y_value ~= -1) && (int1 == 1)

```

```

        y_1 = y_value - 1;
        y_phys_value = Y(1,y_value);
        y_phys_value2 = Y(1,y_1);
        v_2 = V_dim(x_value,y_value);
        v_1 = V_dim(x_value, y_1);
        inter_ratio = (BC - v_1)/(v_2 - v_1);
        y_interpol = y_phys_value2 + (inter_ratio * (y_phys_value -
y_phys_value2));
        PhysVertArray(i,2) = y_interpol;
    end
    if (y_value2 ~= -1) && (int1 == 1)
        y_1 = y_value2 + 1;
        y_phys_value = Y(1,y_value2);
        y_phys_value2 = Y(1,y_1);
        v_2 = V_dim(x_value,y_value2);
        v_1 = V_dim(x_value, y_1);
        inter_ratio = (BC - v_1)/(v_2 - v_1);
        y_interpol = y_phys_value2 + (inter_ratio * (y_phys_value -
y_phys_value2));
        PhysVertArray(i,3) = y_interpol;
    end
end

```

Published with MATLAB® R2019a

```

function OverallStereoAnaly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OverallStereoAnaly centers all of the stereo-PIV data and also
% determines each field's boundary points based on a 5% criteria
%
% OverallStereoAnaly takes the velocity field data (AvgV and
% AvgV_rms)and sends them to the InterArea2 function so that
% it can be centered, its boundary points may be found, and its
% area based on the boundary points may be calculated. The
% outputs from the function are saved to a new structure. After
% the stereo-PIV fields in the new structure are trimmed down
% and then replaced in the new structure.
%
% Who    : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear;

% Here a structure containing all of the stereo-PIV experiments
% average and rms velocity structures are loaded in. Note: there
% are 72 experiments, so there are 144 substructures that will be
% worked on.
load('W:\Carlos Mendez\Matlab Results Folder
\StereoFiles','FilesArray');

% Calculated values are saved in a Results structure. The results
% structure has 4 fields: 'Experiment','BPs','MatArea',
% 'ScriptArea', and 'FlowRate'
Results(72,1) = struct('Experiment',[],'BPs',[],'MatArea',[],...
    'ScriptArea',[],'FlowRate',[]);

% AvgV and AvgV_rms are sent to the InterArea2 function so that
% they can be centered, the boundary points may be found, and
% area based on the boundary points may be calculated. The new
% velocity fields replace the old fields in the FilesArray
% structure and the results are added to the Results structure.
for i = 1:72
    [JetArea,BPs,NewVecField,NewrmsVecField] =
    InterArea2(FilesArray(i).AvgV,...
        FilesArray(i).AvgV_rms);

    FilesArray(i).AvgV = NewVecField;
    FilesArray(i).AvgV_rms = NewrmsVecField;

    Results(i).BPs = BPs;
    Results(i).MatArea = JetArea;
end

% The rows with no data are removed.
[FilesArray] = StereoDeleteZeros(FilesArray);

```

```
% Trimming down the stereo-PIV fields and replacing the old
% fields with the new fields in the FilesArray structure.
for i = 1:72
[B1,B2,B3,B4] = NewGetCutOffs(FilesArray(i).AvgV);
[NewVecField] = StereoCutField(FilesArray(i).AvgV,B1,B2,B3,B4);
[NewrmsVecField] = StereoCutField(FilesArray(i).AvgV_rms,B1,B2,B3,B4);
[NewVecField,NewrmsVecField] =
    EvenStereoField(NewVecField,NewrmsVecField);
FilesArray(i).AvgV = NewVecField;
FilesArray(i).AvgV_rms = NewrmsVecField;
end

% Saving the two structures.
save('W:\Carlos Mendez\Matlab Results Folder
\StereoFiles','FilesArray');
save('W:\Carlos Mendez\Matlab Results Folder
\StereoResults','Results');
end
```

Published with MATLAB® R2019a

```

function [NewVecField,AOR] = RotateStereo(VecField)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RotateStereo centers the stereo-PIV velocity field, determines
% its boundary points based on a 5% criteria, and calculates its
% area based on the boundary points via MATLAB's polyarea function
%
% RotateStereo(VecField) takes the velocity field data(VecField),
% finds its centerline, centers it, finds its boundary points,
% and then uses MATLAB's polyarea function to get the jet area.
%
% Who : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the input velocity field to a new variable
VEC_temp = VecField;

% Finding the maximum streamwise velocity and making a delete
% array in case some columns need to be deleted
maxvel = max(max(VEC_temp.vz));
[r,c] = size(VEC_temp.x);
deletearray = zeros(r,c);

% Determining if there are any points at each Z-location that
% fall within the 5% threshold of the maximum jet velocity.
for i = 1:c
    currentVelrow = VEC_temp.vz(i,:);
    isemptycheck = isempty(find(currentVelrow > (0.05*maxvel)));
    if isemptycheck == 0
        deletearray(1,i) = 0;
    elseif isemptycheck == 1
        deletearray(1,i) = i;
    end
end

deletearray = deletearray(deletearray ~= 0);

% Here cropping out the locations that did not meet the
% criteria checked previously.
if any(deletearray)
    revDeleteArray = fliplr(deletearray);
    [r2,c2] = size(revDeleteArray);
    for i = 1:c2
        VEC_temp.vz(revDeleteArray(i),:) = [];
        VEC_temp.x(revDeleteArray(i)) = [];
    end
end

% Repeating the previous steps to ensure all rows that
% don't meet the criteria are removed (found to be
% necessary to do twice after testing)
[r,c] = size(VEC_temp.x);

```

```

deletearray = zeros(r,c);
for i = 1:c
    currentVelrow = VEC_temp.vz(i,:);
    isemptycheck = isempty(find(currentVelrow > (0.05*maxvel)));
    if isemptycheck == 0
        deletearray(1,i) = 0;
    elseif isemptycheck == 1
        deletearray(1,i) = i;
    end
end

deletearray = deletearray(deletearray ~= 0);

% Here cropping out the locations that did not meet the
% criteria checked previously.
if any(deletearray)
    revDeleteArray = fliplr(deletearray);
    [r2,c2] = size(revDeleteArray);
    for i = 1:c2
        VEC_temp.vz(revDeleteArray(i),:) = [];
        VEC_temp.x(revDeleteArray(i)) = [];
    end
end

% Finding max vel locations at each Z-position. This is
% the centerline of the jet in the Z-direction. Any outliers
% are looked for and their location determined if they
% are found to exist.
[M,I] = max(abs(VEC_temp.vz), [], 2);
CL_matrix_y = VEC_temp.y(I);
[CL_matrix_y,deletearray] = rmoutliers(CL_matrix_y);
deletearray = find(deletearray == 1);

% Deleting outliers if they exist.
if any(deletearray)
    revDeleteArray = fliplr(deletearray);
    [r2,c2] = size(revDeleteArray);
    for i = 1:c2
        VEC_temp.vz(revDeleteArray(i),:) = [];
        VEC_temp.x(revDeleteArray(i)) = [];
    end
end

% Making a linear line of best fit and obtaining the slope
% of the line.
p = polyfit(VEC_temp.x,CL_matrix_y,1);
slope = p(1,1);

% If the slope doesn't equal zero then the jet isn't
% horizontal. The angle of rotation to turn the vector field
% is then determined and the vector field is rotated.
if slope ~= 0
    AOR = atand(0) - atand(slope);
    AOR = deg2rad(AOR);
end

```

```

    VEC_temp2 = rotatef(VecField,-AOR);
elseif slope == 0
    VEC_temp2 = VEC_temp;
end

maxvel = max(max(VEC_temp2.vz));

% Determining the origin point in Y and Z.
% Finding the Y location by determining the mode Y
% location of each maximum velocity at each Z position.
[M,I] = max(abs(VEC_temp2.vz), [], 2);
CL_matrix_y = VEC_temp2.y(I);
zeroloc_y = mode(CL_matrix_y);
CL = VEC_temp2.vz(:,find(VEC_temp2.y == zeroloc_y)).';
CL_size = size(CL);

% Determining the Z position by finding the two locations
% on the centerline where the 0.05 criteria is and then
% calculating the midpoint between these two points.
for i = 1:CL_size(1,2)
    criteria = CL(1,i)/maxvel;
    if criteria < 0.05
        continue
    elseif criteria == 0.05
        x_low = VEC_temp2.x(1,i);
        break
    elseif criteria > 0.05
        xprev = VEC_temp2.x(1,i-1);
        xnext = VEC_temp2.x(1,i);
        ratio = (0.05-(CL(1,i-1)/maxvel))/(criteria - (CL(1,i-1)/
maxvel));
        x_low = xprev + (ratio*(xnext-xprev));
        break
    end
end

for i = CL_size(1,2):-1:1
    criteria = CL(1,i)/maxvel;
    if criteria < 0.05
        continue
    elseif criteria == 0.05
        x_up = VEC_temp2.x(1,i);
        break
    elseif criteria > 0.05
        xprev = VEC_temp2.x(1,i+1);
        xnext = VEC_temp2.x(1,i);
        ratio = (0.05-(CL(1,i+1)/maxvel))/(criteria - (CL(1,i+1)/
maxvel));
        x_up = xprev + (ratio*(xnext-xprev));
        break
    end
end

zeroloc_x = (x_up+x_low)/2;

```

```
% Making new velocity field by rotating if necessary, and
% centering the field about the origin. The new field is
% saved to the output variable.
if slope ~= 0
    NewVecField = rotatef(VecField,-AOR);
    NewVecField.x = NewVecField.x - zeroloc_x;
    NewVecField.y = NewVecField.y - zeroloc_y;
elseif slope == 0
    NewVecField = VEC_temp;
    NewVecField.x = NewVecField.x - zeroloc_x;
    NewVecField.y = NewVecField.y - zeroloc_y;
end
end
```

Published with MATLAB® R2019a

```

function [AllData,PolygonDataX,PolygonDataY] = Sort2(CL_array,
  Y_axis_array,HorizArray,VertArray)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sort2 takes boundary points based on a 5% criteria, and sorts them
% into clockwise order starting from the positive Y-axis.
%
% Sort2(CL_array, Y_axis_array,HorizArray,VertArray) takes the
% boundary points from the two tests (HorizArray,VertArray),
% orders them in each of the 4 quadrants, and then puts them
% together with the boundary points on the Y-axis and
% centerline so that they go in the clockwise direction starting
% from the positive Y-axis.
%
% Who : Carlos Mendez
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determining the size of the two arrays. Each array has the Z
% and Y coordinates for each boundary point for each test. This
% means that there are two coordinates in each row.
size1 = size(HorizArray);
size2 = size(VertArray);

% Extracting the Z&Y points from the two tests and putting them
% in new arrays with only two columns rather than four.
newHorizArray1(1:size1(1,1),1) = HorizArray(1:end,2);
newHorizArray1(1:size1(1,1),2) = HorizArray(1:end,1);
newHorizArray2(1:size1(1,1),1) = HorizArray(1:end,3);
newHorizArray2(1:size1(1,1),2) = HorizArray(1:end,1);
newVertArray1(1:size2(1,1),1) = VertArray(1:end,1);
newVertArray1(1:size2(1,1),2) = VertArray(1:end,2);
newVertArray2(1:size2(1,1),1) = VertArray(1:end,1);
newVertArray2(1:size2(1,1),2) = VertArray(1:end,3);

% Putting all the data into one array. The values of each quadrant
% will be extracted and sorted accordingly and then put together.
unsortedData =
  vertcat(newHorizArray1,newVertArray1,newHorizArray2,newVertArray2);

% Searching for the values in the 1st quadrant and sorting them.
rowsX = find(unsortedData(:,1) > Y_axis_array(1,1));
rowsY = find(unsortedData(:,2) > CL_array(1,1));
quadlindices = intersect(rowsX,rowsY);
quad1 = unsortedData(quadlindices,:);
quad1 = sortrows(quad1,1,'ascend');

% Searching for the values in the 2nd quadrant and sorting them.
rowsX = find(unsortedData(:,1) > Y_axis_array(1,1));
rowsY = find(unsortedData(:,2) < CL_array(1,1));
quad2indices = intersect(rowsX,rowsY);
quad2 = unsortedData(quad2indices,:);
quad2 = sortrows(quad2,1,'descend');

```

```

% Searching for the values in the 3rd quadrant and sorting them.
rowsX = find(unsortedData(:,1) < Y_axis_array(1,1));
rowsY = find(unsortedData(:,2) < CL_array(1,1));
quad3indices = intersect(rowsX,rowsY);
quad3 = unsortedData(quad3indices,:);
quad3 = sortrows(quad3,1,'descend');

% Searching for the values in the 4th quadrant and sorting them.
rowsX = find(unsortedData(:,1) < Y_axis_array(1,1));
rowsY = find(unsortedData(:,2) > CL_array(1,1));
quad4indices = intersect(rowsX,rowsY);
quad4 = unsortedData(quad4indices,:);
quad4 = sortrows(quad4,1,'ascend');

% Extracting the points on the y-axis and CL so they can be
% added to the boundary points array.
yaxis1 = [Y_axis_array(1,1),Y_axis_array(1,2)];
yaxis2 = [Y_axis_array(1,1),Y_axis_array(1,3)];
CL1 = [CL_array(1,2),CL_array(1,1)];
CL2 = [CL_array(1,3),CL_array(1,1)];

% Putting all the boundary points together and saving the
% results to the output variables.
AllData = vertcat(yaxis1,quad1,CL1,quad2,yaxis2,quad3,CL2,quad4);
PolygonDataX = AllData(:,1);
PolygonDataY = AllData(:,2);
end

```

Published with MATLAB® R2019a

```

function StereoPlots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% StereoPlots This is a function to make whole-field plots
% for the stereo-PIV data.
%
%   MakeTwoDVectorFields loads in the stereo-PIV data, and 10 whole-
field
%   measurements for each jet configuration, and saves them to their
%   designated folders.
%
% Who   : Carlos Mendez
% Where: University of Alberta
% When: Fall 2019
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading in the vector fields
load('W:\Carlos Mendez\Matlab Results Folder
\StereoFiles','FilesArray')
for i = 1:72

    % Getting the current vector field's relevant values
    currentVave = FilesArray(i).AvgV;
    currentVave_rms = FilesArray(i).AvgV_rms;

    CL_loc1 = find(currentVave.x == 0);

    CL_loc2 = find(currentVave.y == 0);

    CL_Vel = currentVave.vz(CL_loc1,CL_loc2);

    minx = min(currentVave.x);
    maxx = max(currentVave.x);
    maxy = max(currentVave.y);
    miny = min(currentVave.y);
    xlims = [minx,maxx];
    ylims = [maxy,miny];

    % Preparing the save pathway
    experiment_string1 = ['W:\Carlos Mendez\Matlab Results Folder
\Vector Field Colour Maps\Emf Version\' ...
FilesArray(i).AR FilesArray(i).Plane FilesArray(i).FlowRate];
    experiment_string2 = ['W:\Carlos Mendez\Matlab Results Folder
\Vector Field Colour Maps\Mat Version\' ...
FilesArray(i).AR FilesArray(i).Plane FilesArray(i).FlowRate];

% Plots

% U Vector Field
figure(1)
imagesc(xlims,ylims,rot90(currentVave.vz/CL_Vel))
axis xy

```

```

colormap jet
c1 = colorbar;
ylabel(c1, '{U/U_{c1}}', 'FontSize',16, 'FontName', 'Times New Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_UUcl'];
savename2 = [experiment_string2 '_UUcl'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

% V Vector Field
imagesc(xlims,ylim,rot90(currentVave.vy/CL_Vel))
axis xy
colormap jet
c1 = colorbar;
ylabel(c1, '{V/U_{c1}}', 'FontSize',16, 'FontName', 'Times New Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_VUcl'];
savename2 = [experiment_string2 '_VUcl'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

% W Vector Field
figure(3)
imagesc(xlims,ylim,rot90(currentVave.vx/CL_Vel))
axis xy
colormap jet
c1 = colorbar;
ylabel(c1, '{W/U_{c1}}', 'FontSize',16, 'FontName', 'Times New Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_WUcl'];
savename2 = [experiment_string2 '_WUcl'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

% U rms Vector Field
figure(4)
imagesc(xlims,ylim,rot90(currentVave_rms.vz/CL_Vel));
axis xy
colormap jet
c1 = colorbar;
ylabel(c1, '$\mathrm{\sqrt{{u}^2}}/U_{c1}$', 'Interpreter', 'latex', 'FontSize',16, 'FontName', 'Times New Roman');
xlabel('Z/D_e');

```

```

ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_uUcl_2'];
savename2 = [experiment_string2 '_uUcl_2'];
saveas(gcf,savename1,'emf');
saveas(gcf,savename2,'m');

% V rms Vector Field
figure(5)
imagesc(xlims,ylims,rot90(currentVave_rms.vy/CL_Vel));
axis xy
colormap jet
c1 = colorbar;
ylabel(c1,'$\mathrm{\sqrt{{v}^2}}/$
U_{c1}$','Interpreter','latex','FontSize',16,'FontName','Times New
Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_vUcl_2'];
savename2 = [experiment_string2 '_vUcl_2'];
saveas(gcf,savename1,'emf');
saveas(gcf,savename2,'m');

% W rms Vector Field
figure(6)
imagesc(xlims,ylims,rot90(currentVave_rms.vx/CL_Vel))
axis xy
colormap jet
c1 = colorbar;
ylabel(c1,'$\mathrm{\sqrt{{w}^2}}/$
U_{c1}$','Interpreter','latex','FontSize',16,'FontName','Times New
Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_wUcl_2'];
savename2 = [experiment_string2 '_wUcl_2'];
saveas(gcf,savename1,'emf');
saveas(gcf,savename2,'m');

% Rey_xx Shear Stress Vector Field
figure(7)
Rey_xx = currentVave.Rey_zz;
Rey_xx = (Rey_xx/(CL_Vel.^2));
Rey_xx = Rey_xx*100;
imagesc(xlims,ylims,rot90(Rey_xx))
axis xy
colormap jet
c1 = colorbar;

```

```

ylabel(c1,'$\mathrm{\overline{u'u'}}/U_{cl}^2 \ \times$
  100$', 'Interpreter', 'latex', 'FontSize',16, 'FontName', 'Times New
  Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_uuUcl2'];
savename2 = [experiment_string2 '_uuUcl2'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

% Rey_xy Shear Stress Vector Field
figure(8)
Rey_xy = currentVave.Rey_yz;
Rey_xy = (Rey_xy/(CL_Vel.^2));
Rey_xy = Rey_xy*100;
imagesc(xlims,ylims,rot90(Rey_xy))
axis xy
colormap jet
c1 = colorbar;
ylabel(c1,'$\mathrm{\overline{u'v'}}/U_{cl}^2 \ \times$
  100$', 'Interpreter', 'latex', 'FontSize',16, 'FontName', 'Times New
  Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_uvUcl2'];
savename2 = [experiment_string2 '_uvUcl2'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

% Rey_xz Shear Stress Vector Field
figure(9)
Rey_xz = currentVave.Rey_xz;
Rey_xz = (Rey_xz/(CL_Vel.^2));
Rey_xz = Rey_xz*100;
imagesc(xlims,ylims,rot90(Rey_xz))
axis xy
colormap jet
c1 = colorbar;
ylabel(c1,'$\mathrm{\overline{u'w'}}/U_{cl}^2 \ \times$
  100$', 'Interpreter', 'latex', 'FontSize',16, 'FontName', 'Times New
  Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_uwUcl2'];
savename2 = [experiment_string2 '_uwUcl2'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

```

```

% Turbulent Kinetic Energy Vector Field
figure(14)
TKE = vec2scal(currentVave_rms, 'en');
TKE = (TKE.w/(CL_Vel.^2));
TKE = TKE*100;
imagesc(xlims,ylims,rot90(TKE))
axis xy
colormap jet
c1 = colorbar;
ylabel(c1, 'TKE/U_{c1}^2 \times 100', 'FontSize',16, 'FontName', 'Times
  New Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_TKEUc12'];
savename2 = [experiment_string2 '_TKEUc12'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

% Vorticity Vector Field
De = 2*sqrt(12/pi);
curl = vec2scal(currentVave, 'curl');
curl = curl.w;
curl = curl*(De/CL_Vel);
figure(16)
imagesc(xlims,ylims,rot90(curl));
axis xy
colormap jet
c1 = colorbar;
ylabel(c1, '\omega \times D_e/U_{c1}', 'FontSize',16, 'FontName', 'Times
  New Roman');
xlabel('Z/D_e');
ylabel('Y/D_e');
FigureFormatScript

savename1 = [experiment_string1 '_Vorticity'];
savename2 = [experiment_string2 '_Vorticity'];
saveas(gcf,savename1, 'emf');
saveas(gcf,savename2, 'm');

close all
end

end

```

Published with MATLAB® R2019a

D Two-Dimensional PIV Figures

This appendix contains all of the figures that were generated from the 2D-PIV experiments.

The extra figures pertaining to Chapters 5 and 6 are provided in the following order:

- The Streamwise Velocity Decay (Aspect Ratio Study; pp.259-261)
- The Streamwise Velocity Turbulence Intensity (Aspect Ratio Study; pp.262-264)
- The Lateral Velocity Half-Width (Aspect Ratio Study; pp.265-267)
- The Streamwise Velocity Decay (Reynolds Number Study; pp.268-269)
- The Lateral Velocity Half-Width (Reynolds Number Study; pp.270-271)
- The Streamwise Velocity Turbulence Intensity (Reynolds Number Study; pp.272-273)

Additionally, supplementary whole-field figures related to the stitching process described in Chapter 4 are presented in the following order:

- Vector Field Plots Showing Magnitude via Color Maps (pp.274-279)
- Vector Field Plots Showing Direction via Arrows (pp.280-285)
- Vector Field Plots Showing Magnitude and Direction (pp.286-291)

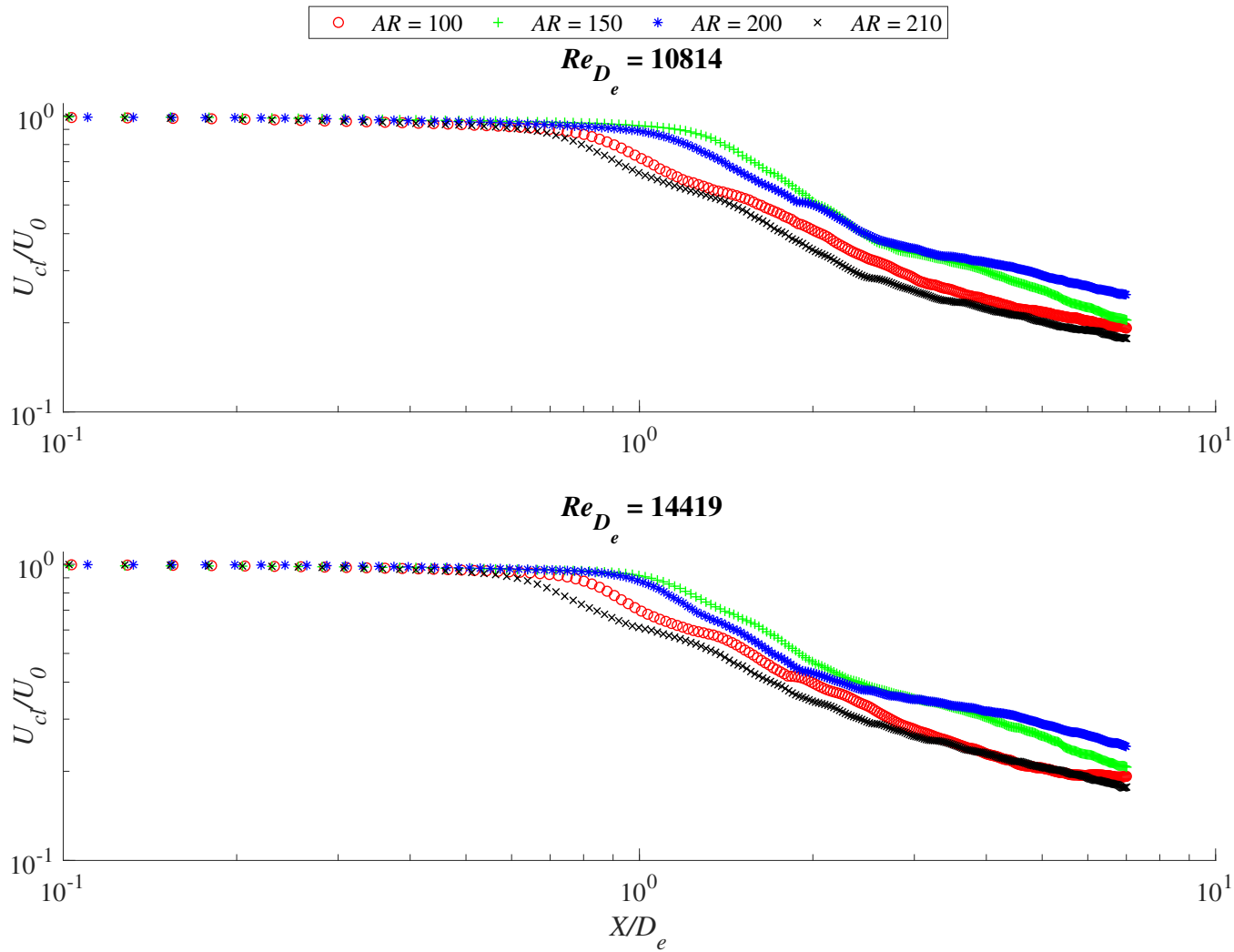


Figure D.1: Velocity decay at $Re = 10800$ and $Re = 14400$. Every 5th data point has been plotted for clarity

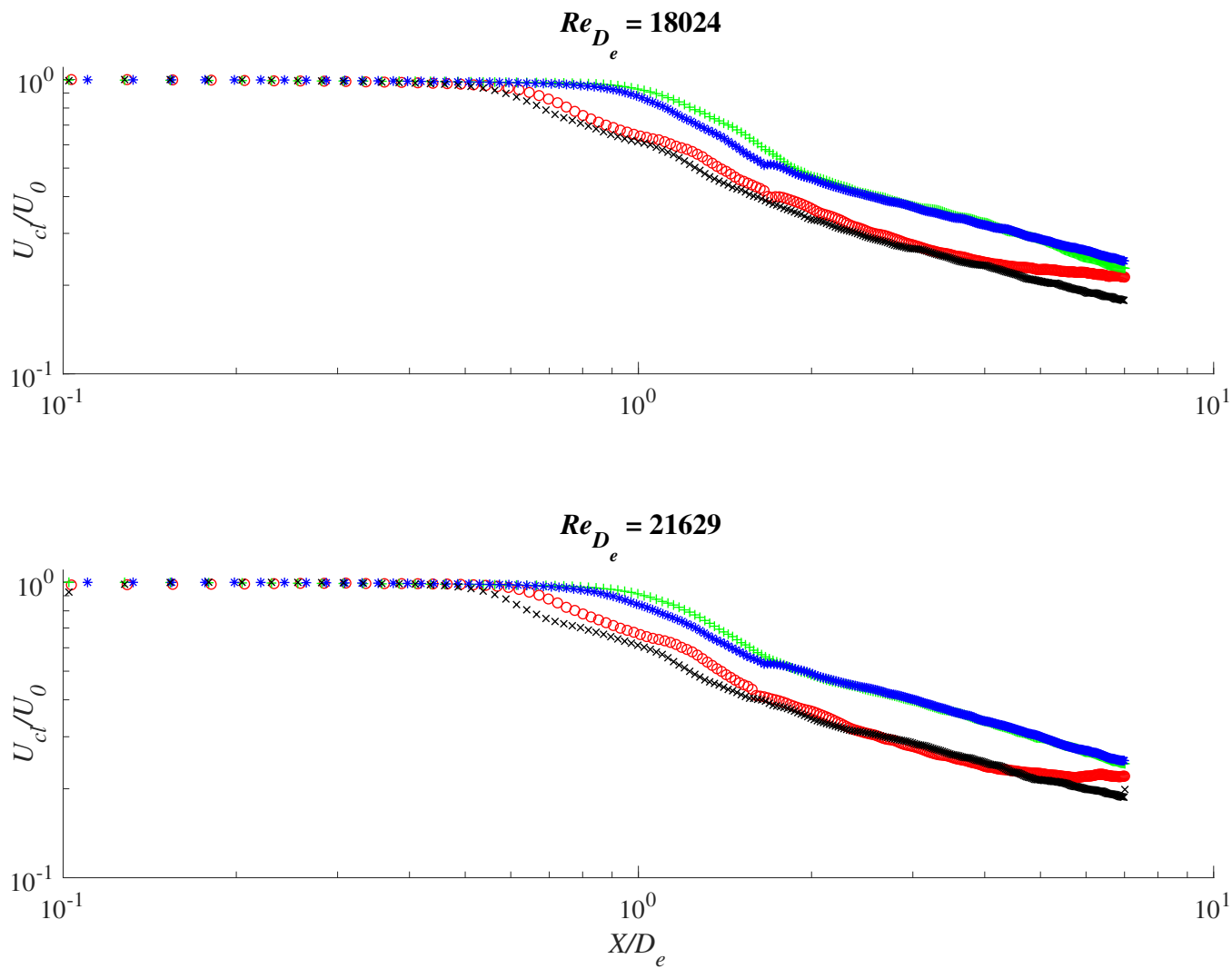


Figure D.2: Velocity decay at $Re = 18000$ and $Re = 21600$. Every 5th data point has been plotted for clarity

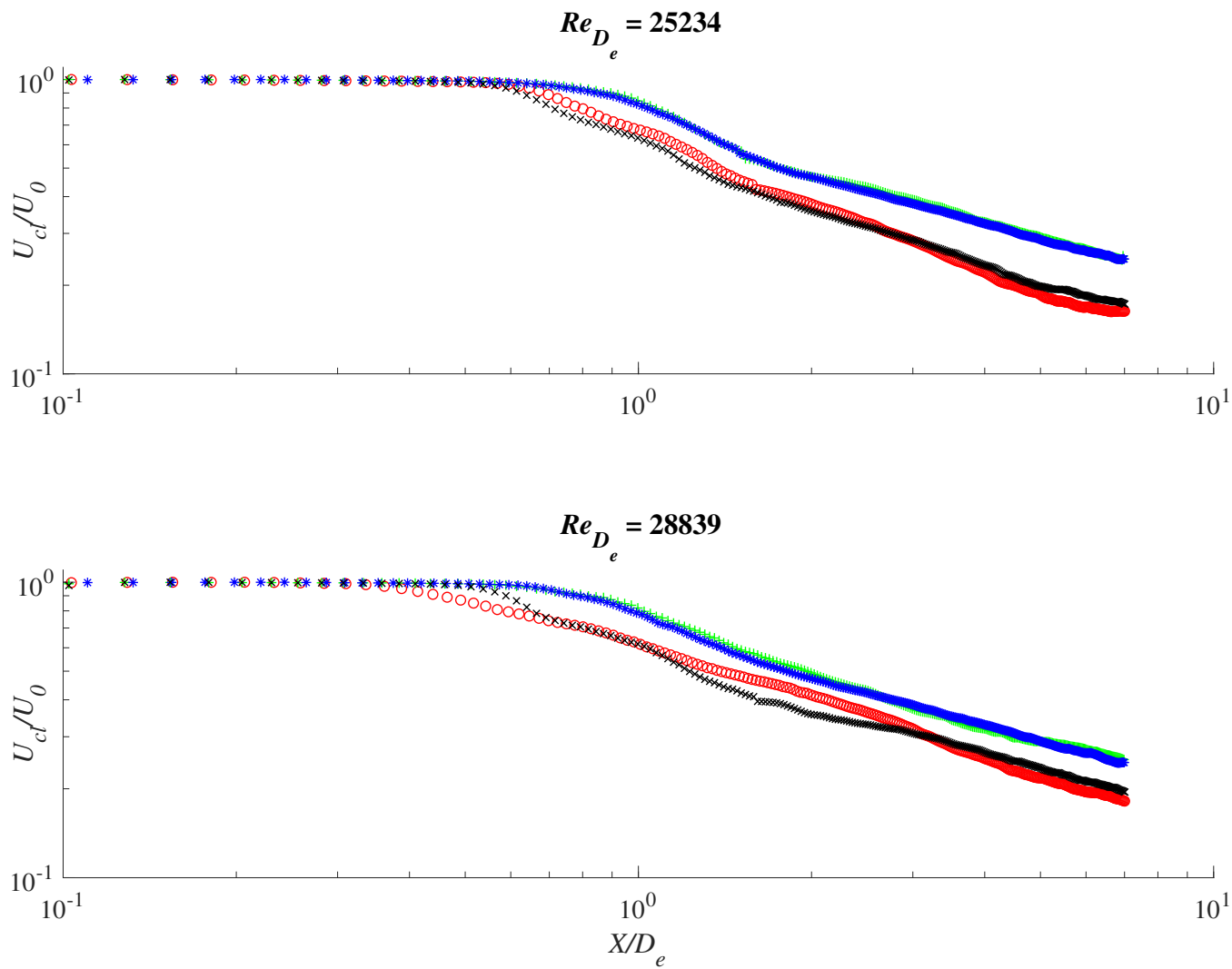


Figure D.3: Velocity decay at $Re = 25200$ and $Re = 28800$. Every 5th data point has been plotted for clarity

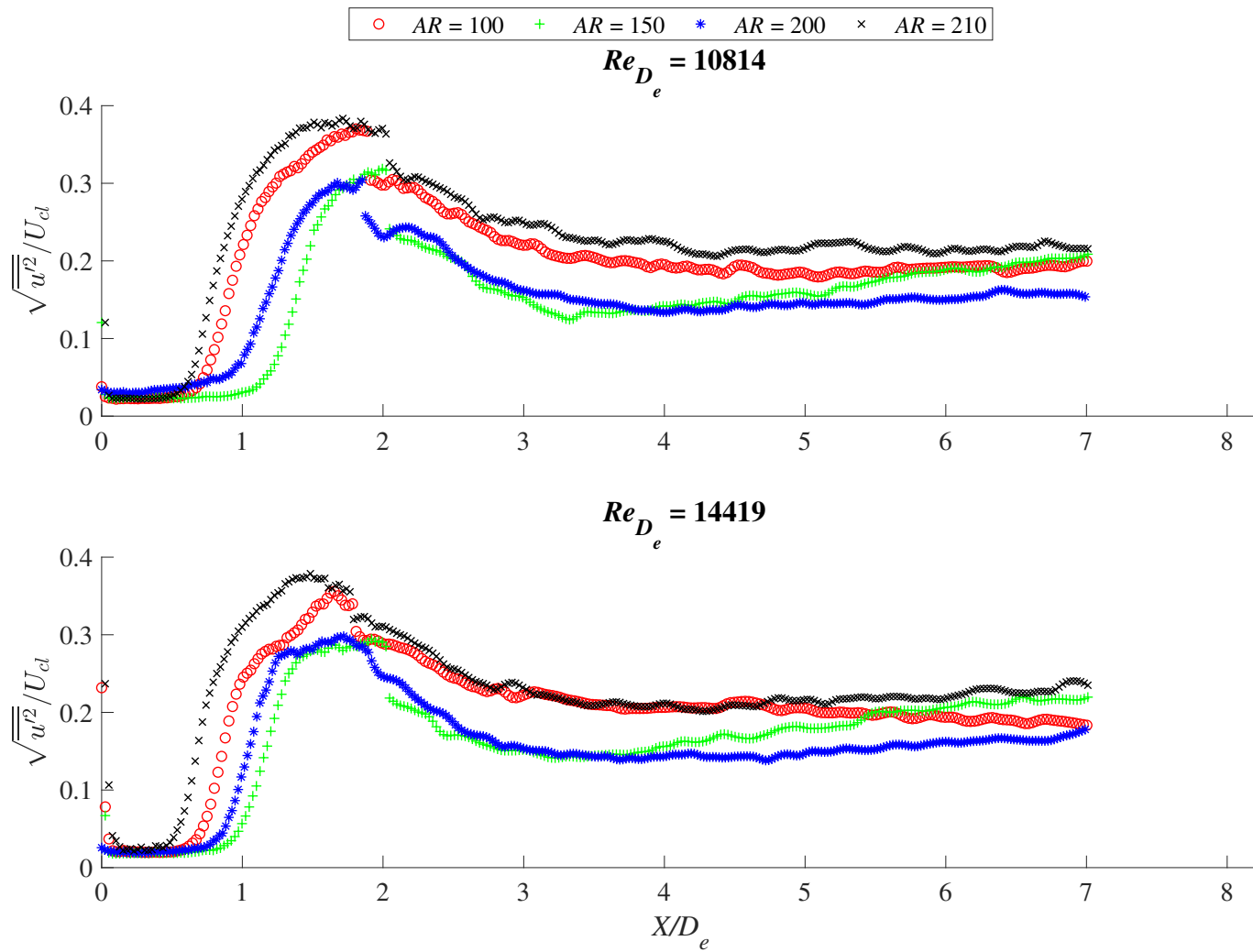


Figure D.4: Normalized turbulence intensity against X/D_e at $Re = 10800$ and $Re = 14400$. Every 5th data point has been plotted for clarity

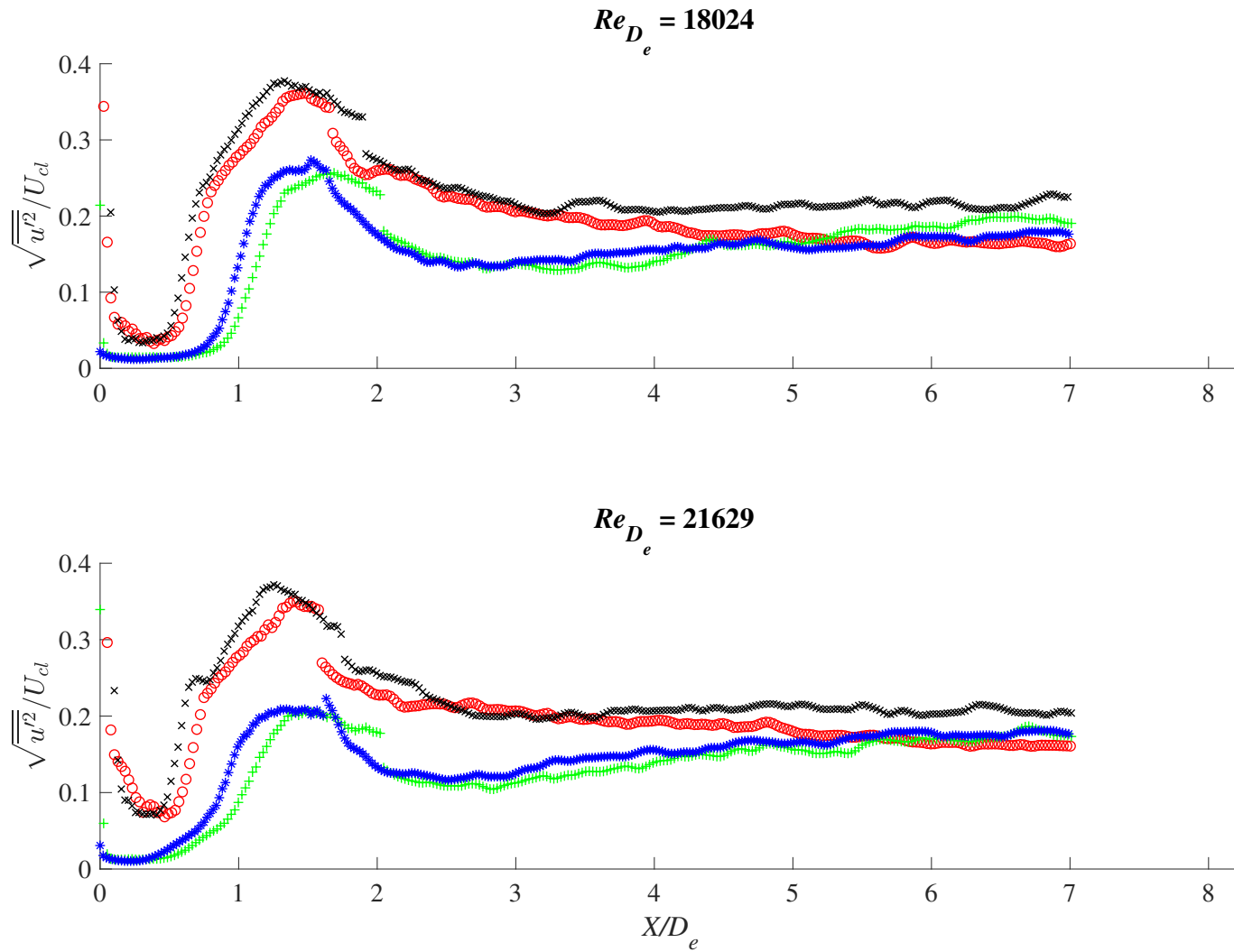


Figure D.5: Normalized turbulence intensity against X/D_e at $Re = 18000$ and $Re = 21600$. Every 5th data point has been plotted for clarity

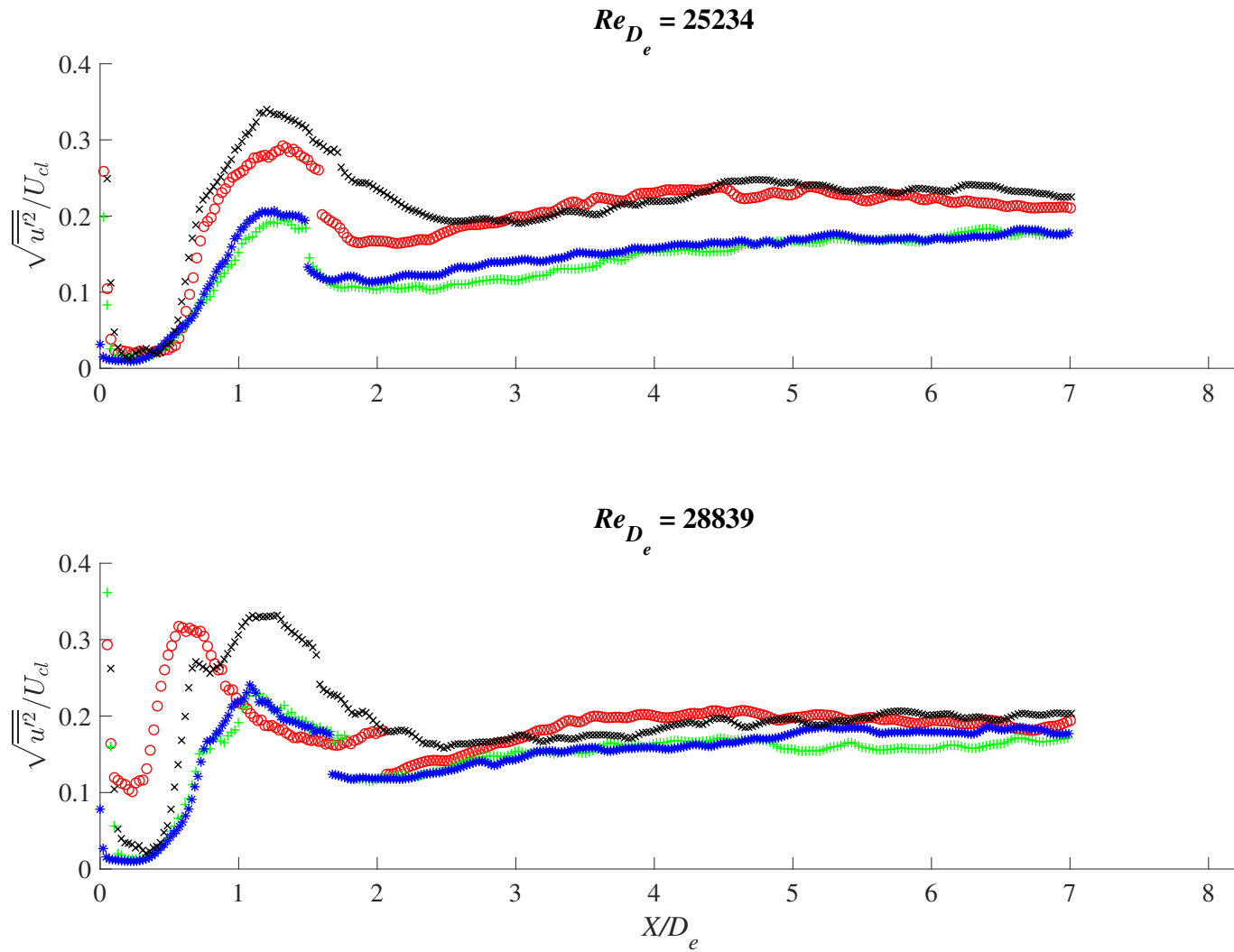


Figure D.6: Normalized turbulence intensity against X/D_e at $Re = 25200$ and $Re = 28800$. Every 5th data point has been plotted for clarity

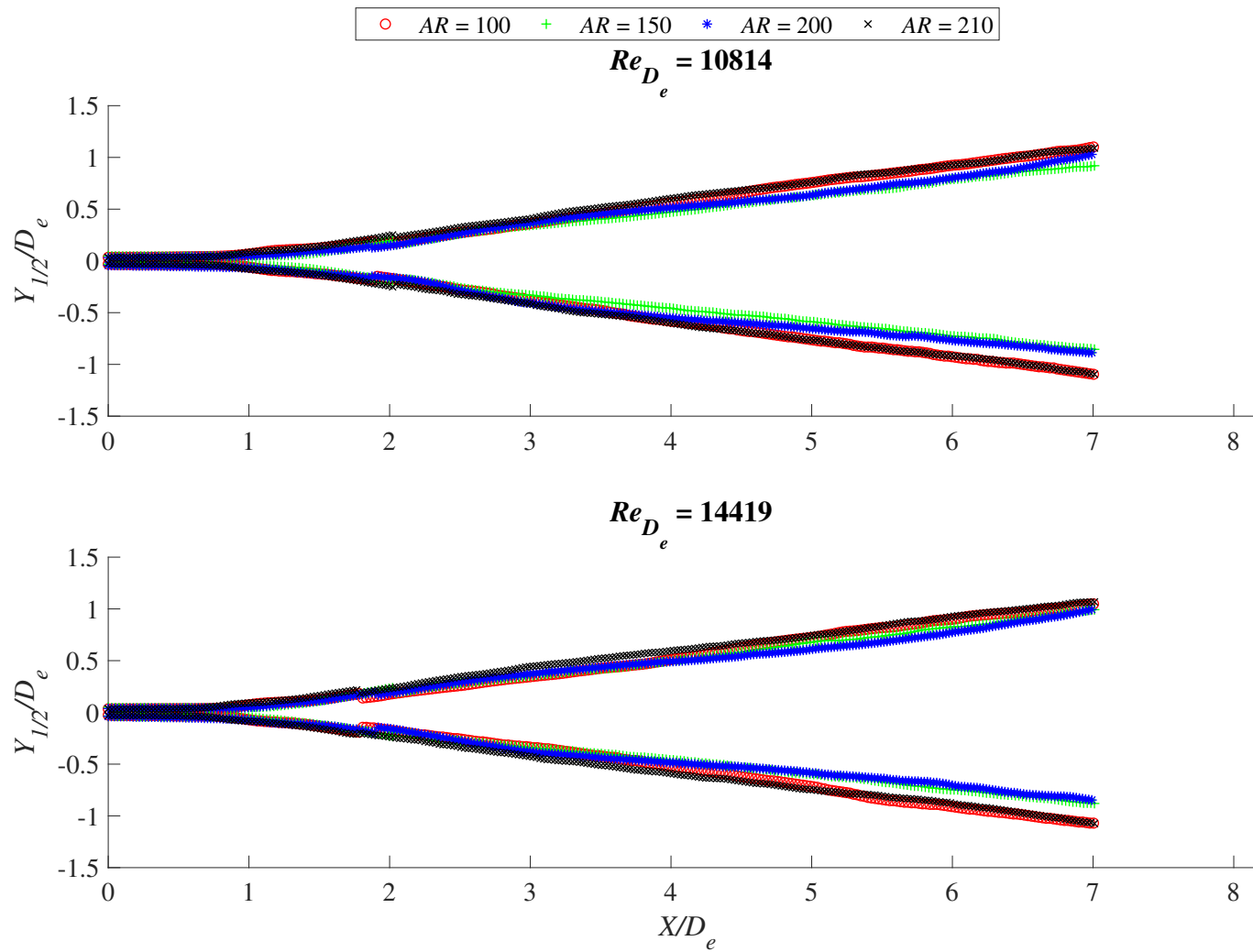


Figure D.7: Normalized lateral velocity half-width against X/D_e at $Re = 10800$ and $Re = 14400$. Every 5th data point has been plotted for clarity

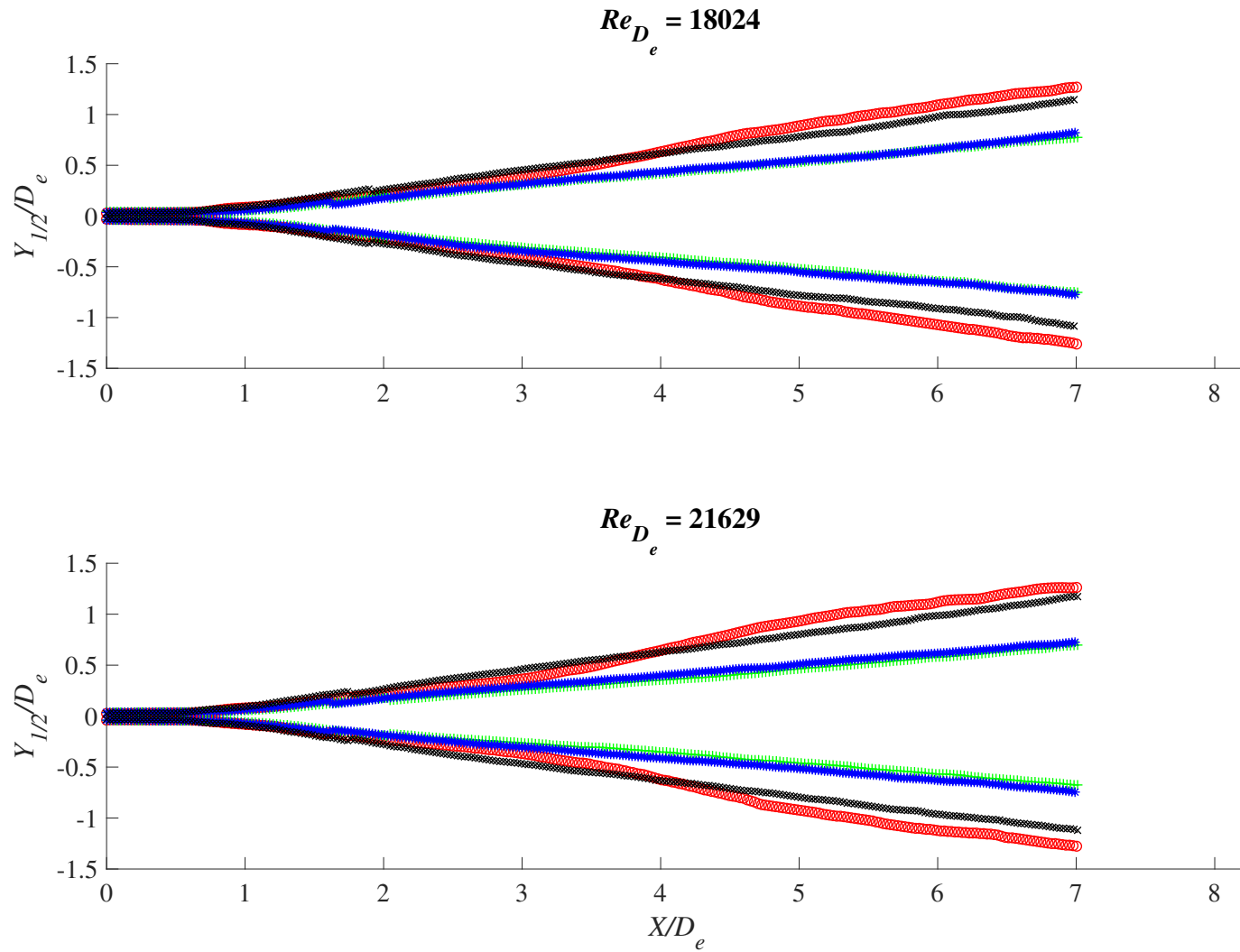


Figure D.8: Normalized lateral velocity half-width against X/D_e at $Re = 18000$ and $Re = 21600$. Every 5th data point has been plotted for clarity

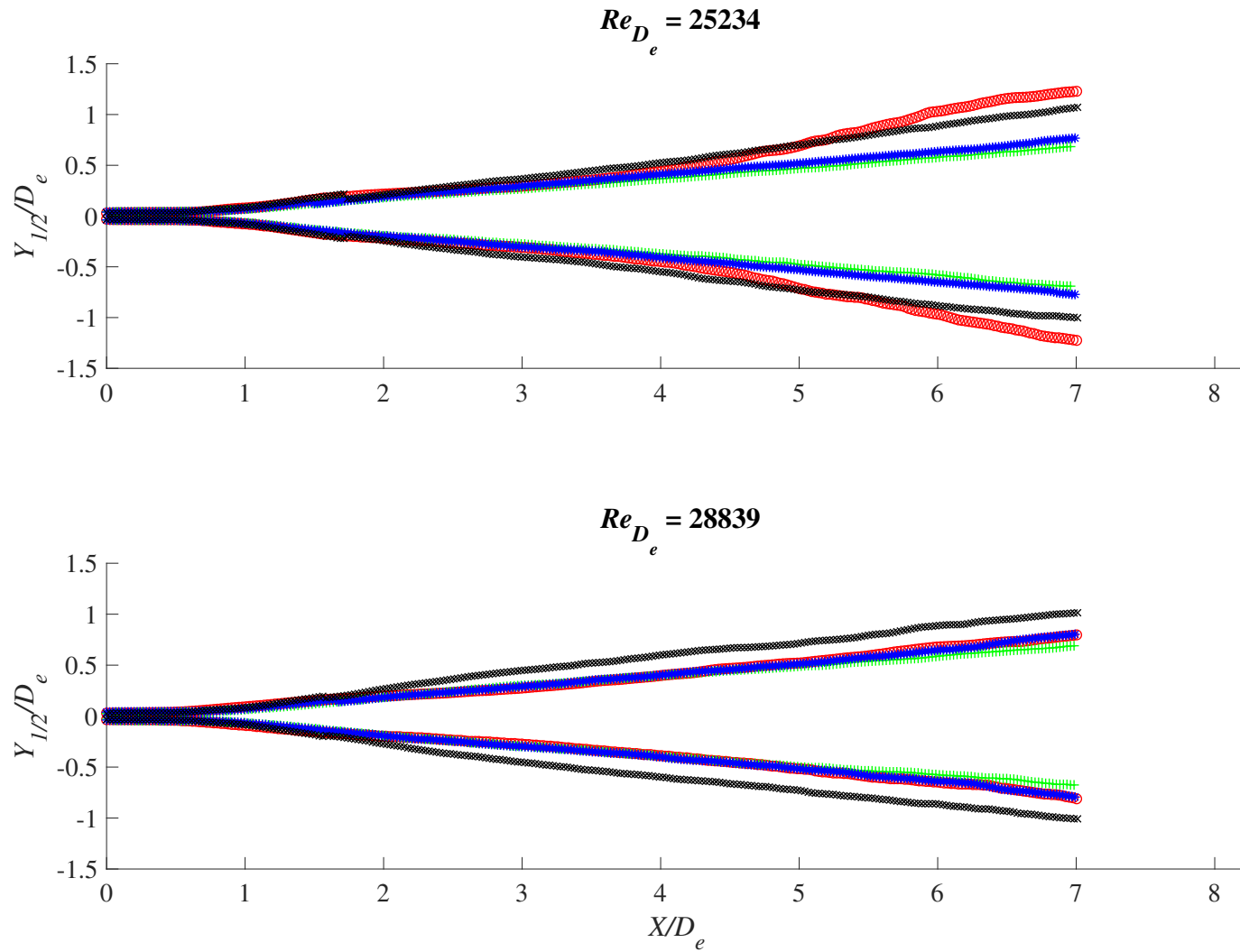


Figure D.9: Normalized lateral velocity half-width against X/D_e at $Re = 25200$ and $Re = 28800$. Every 5th data point has been plotted for clarity

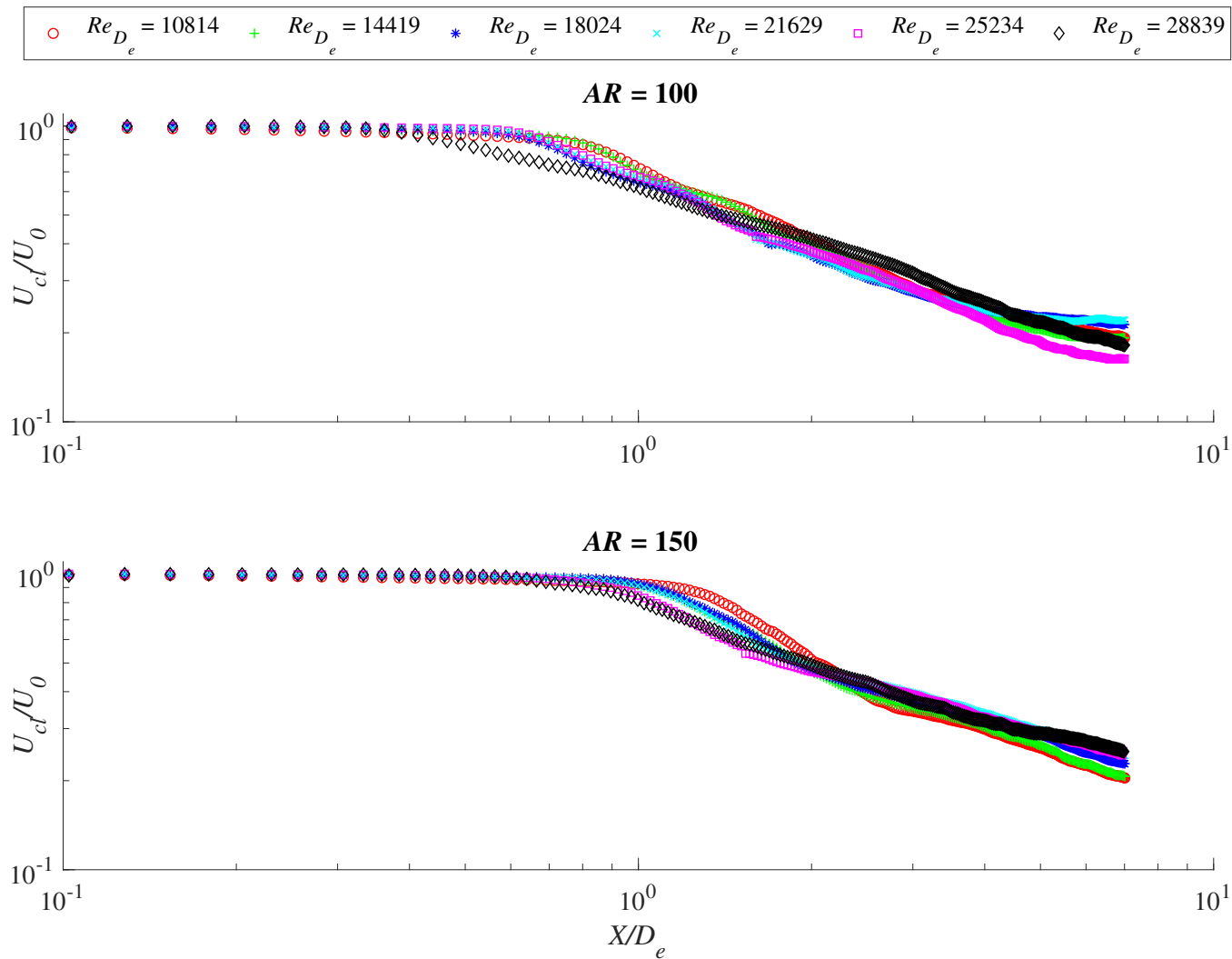


Figure D.10: Velocity decay at $AR = 100$ and $AR = 150$. Every 5th data point has been plotted for clarity

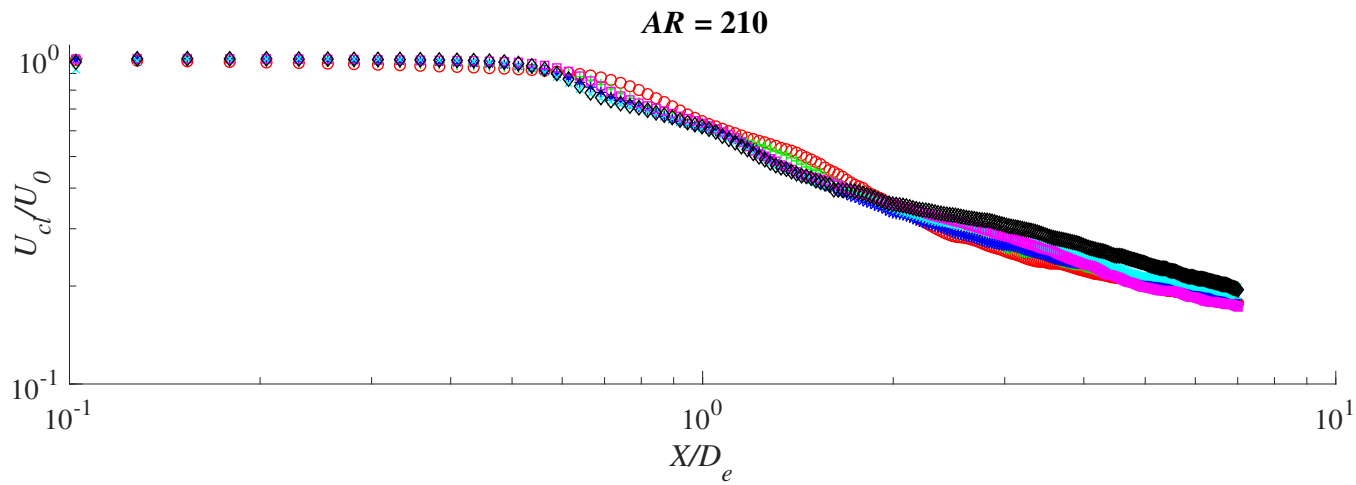
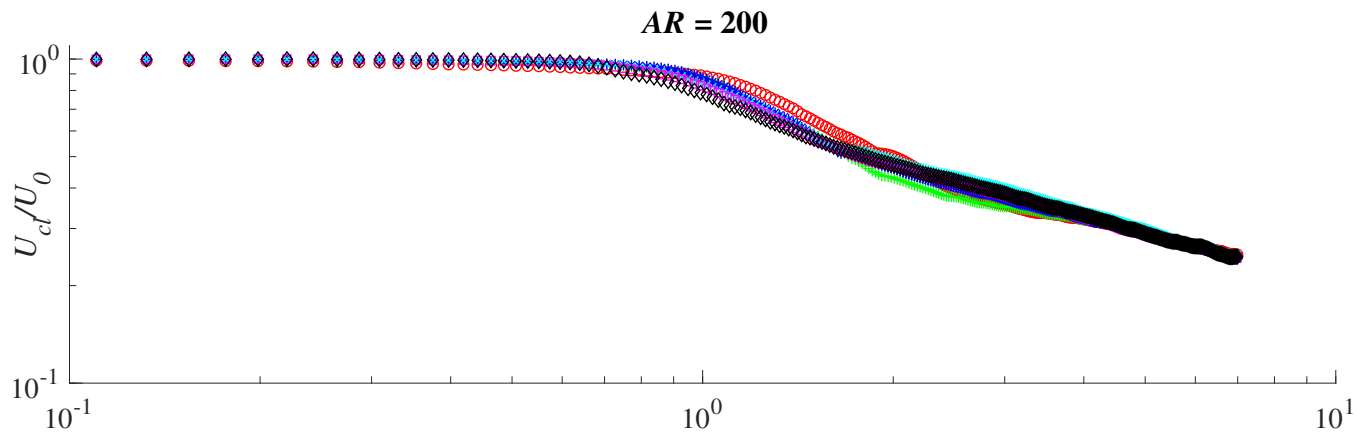


Figure D.11: Velocity decay at $AR = 200$ and $AR = 210$. Every 5th data point has been plotted for clarity

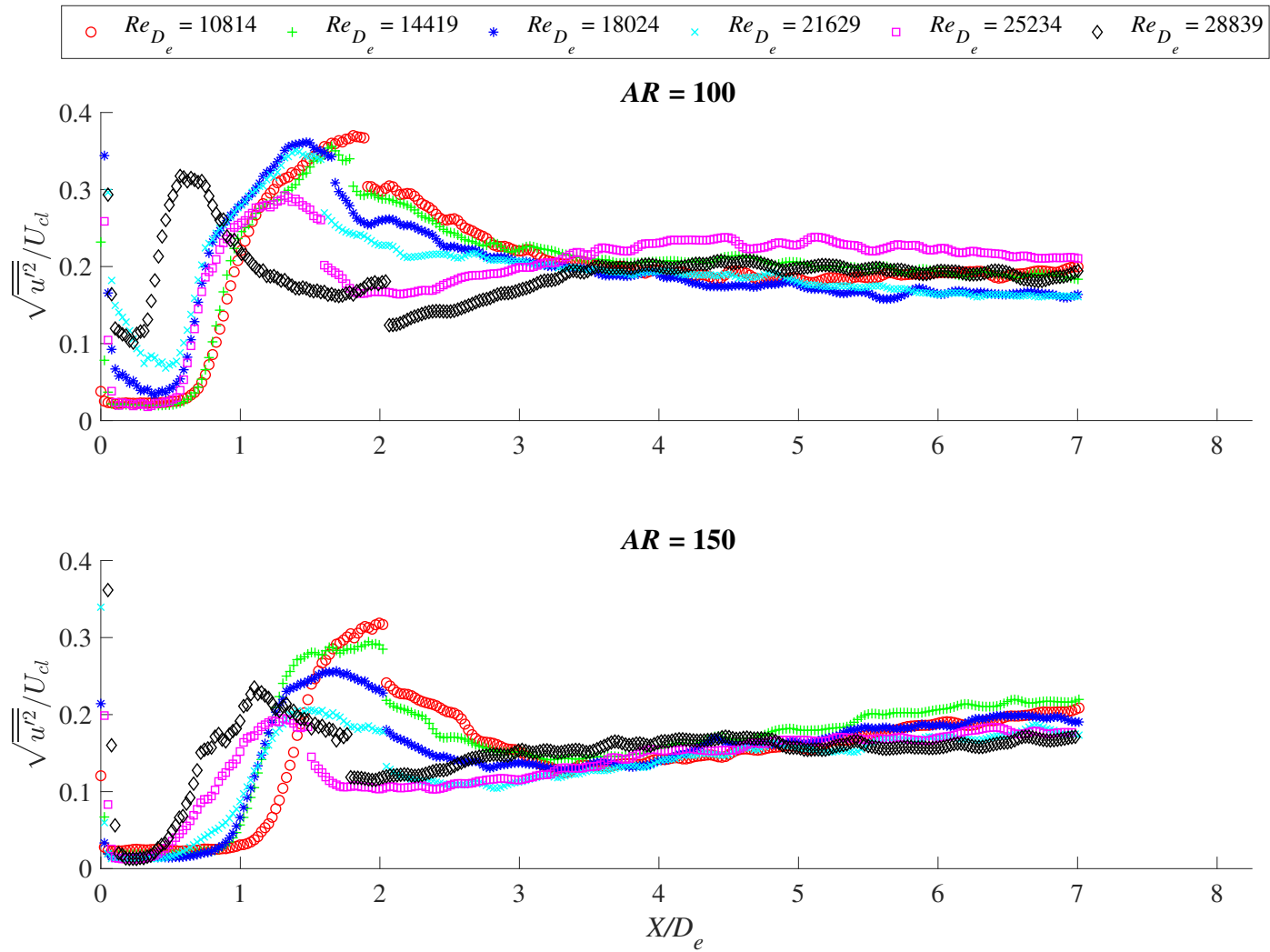


Figure D.12: Normalized turbulence intensity against X/D_e at $AR = 100$ and $AR = 150$. Every 5th data point has been plotted for clarity

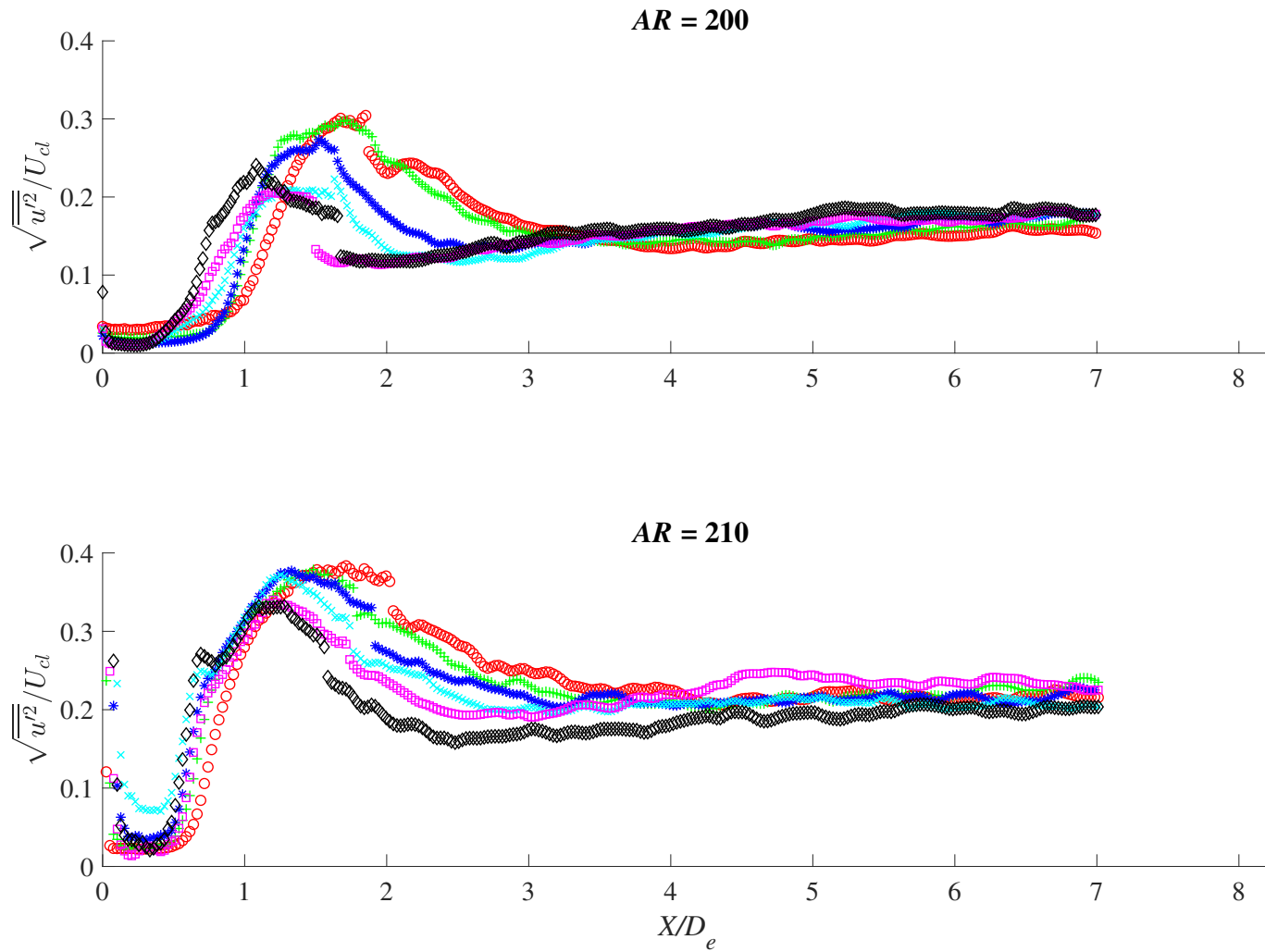


Figure D.13: Normalized turbulence intensity against X/D_e at $AR = 200$ and $AR = 210$. Every 5th data point has been plotted for clarity

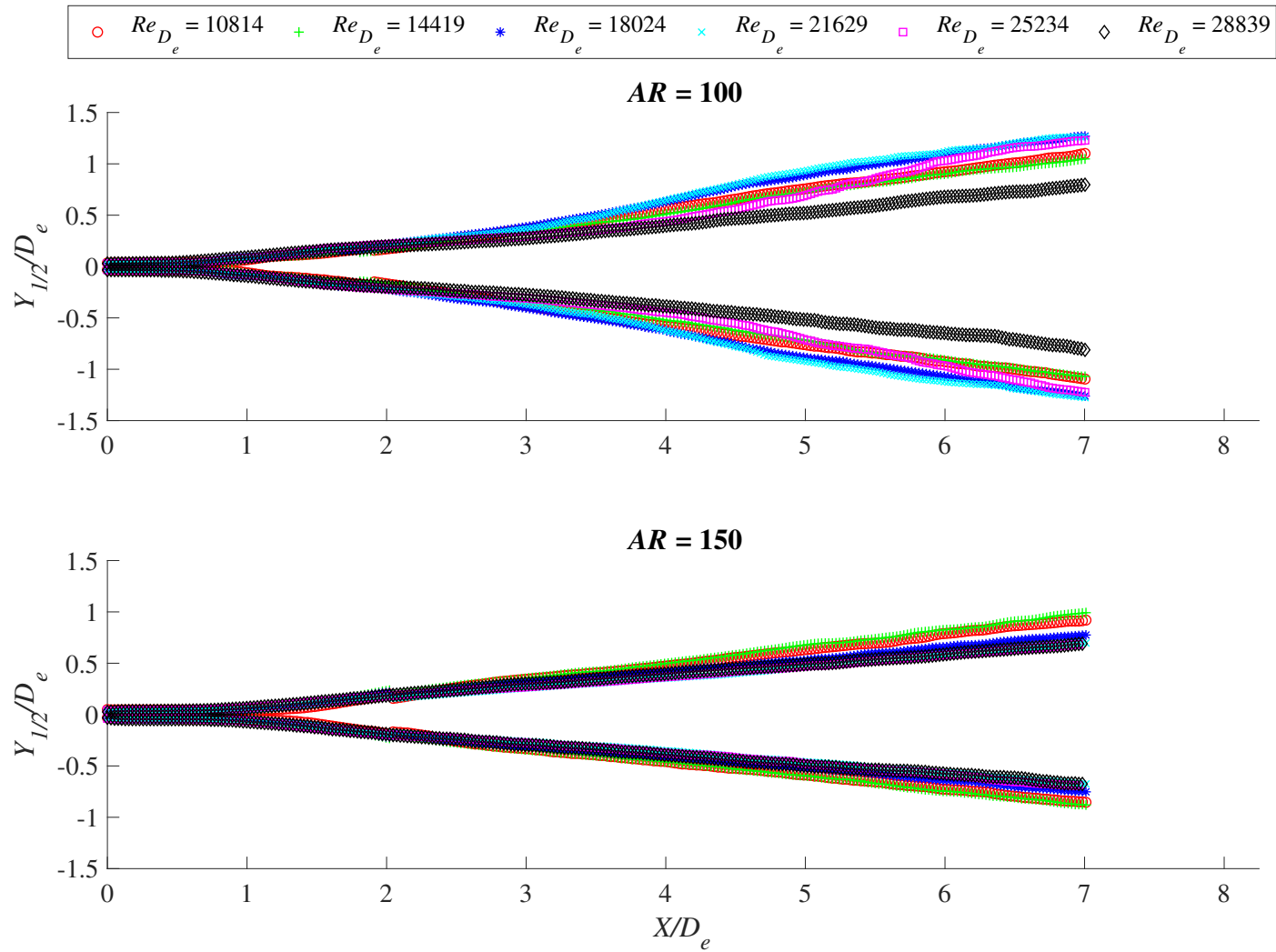


Figure D.14: Normalized lateral velocity half-width against X/D_e at $AR = 100$ and $AR = 150$. Every 5th data point has been plotted for clarity

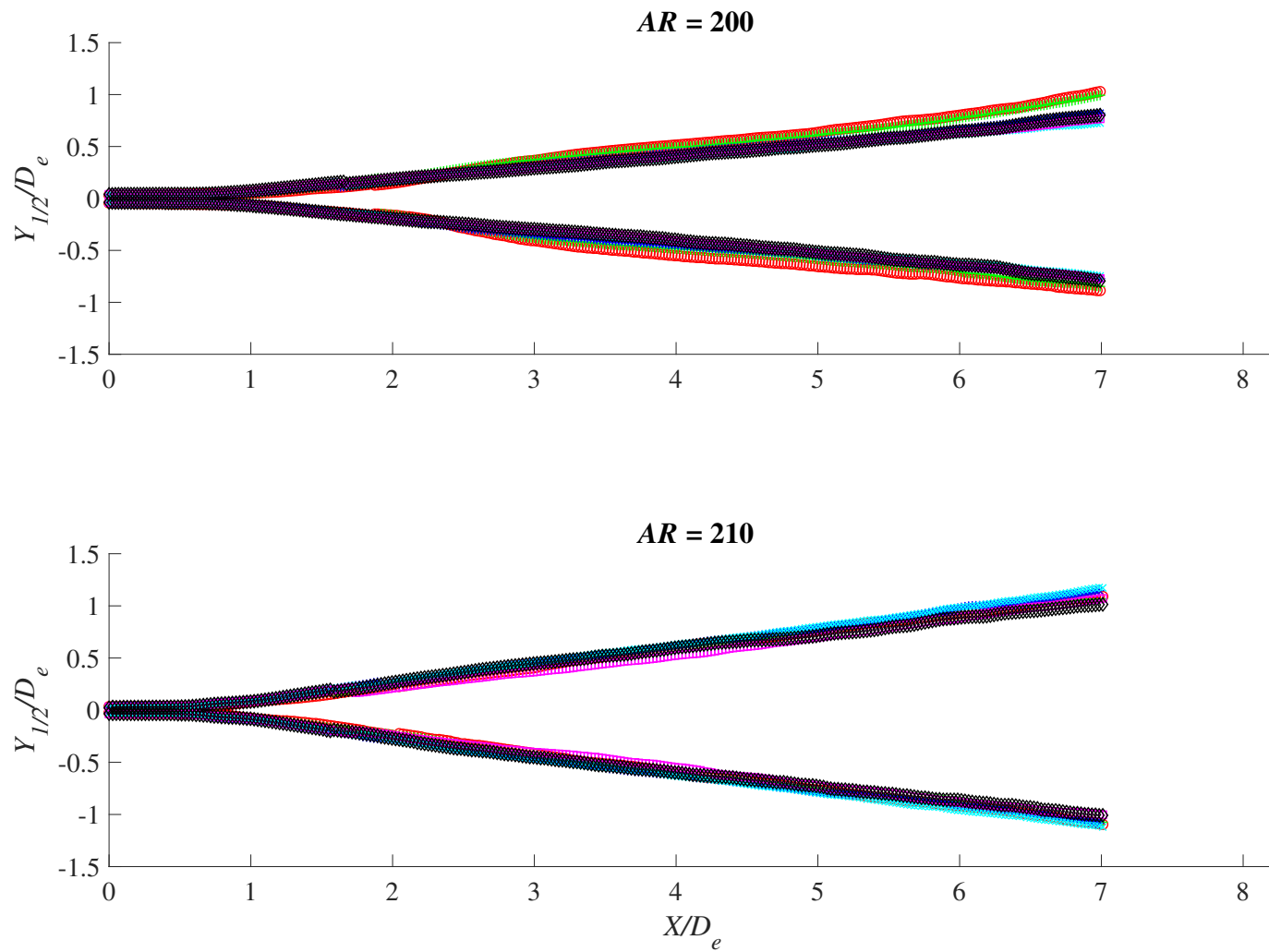
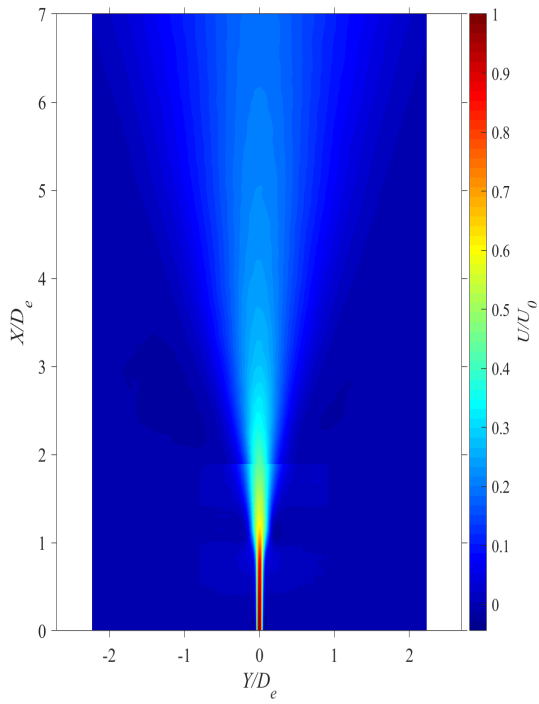
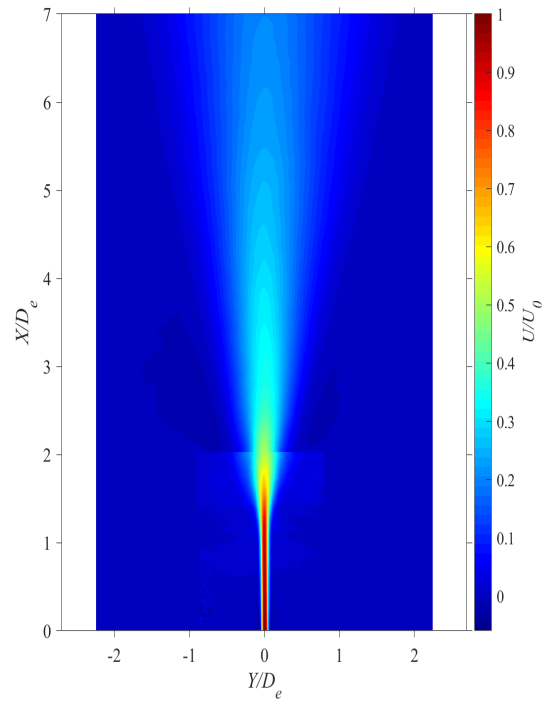


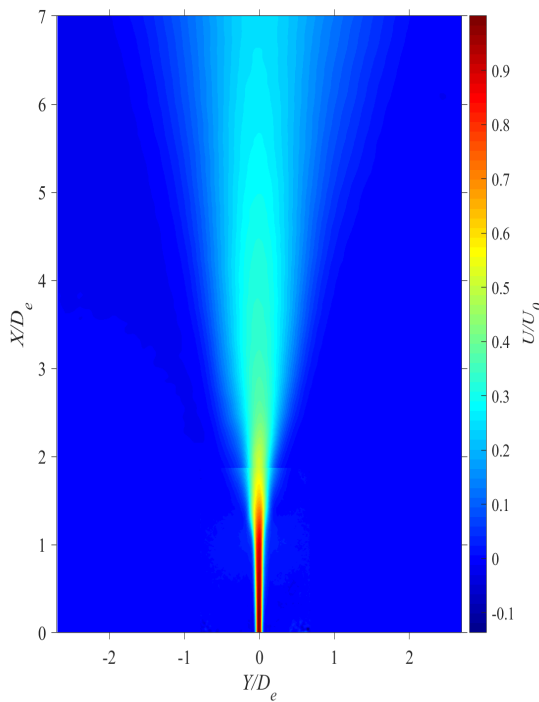
Figure D.15: Normalized lateral velocity half-width against X/D_e at $AR = 200$ and $AR = 210$. Every 5th data point has been plotted for clarity



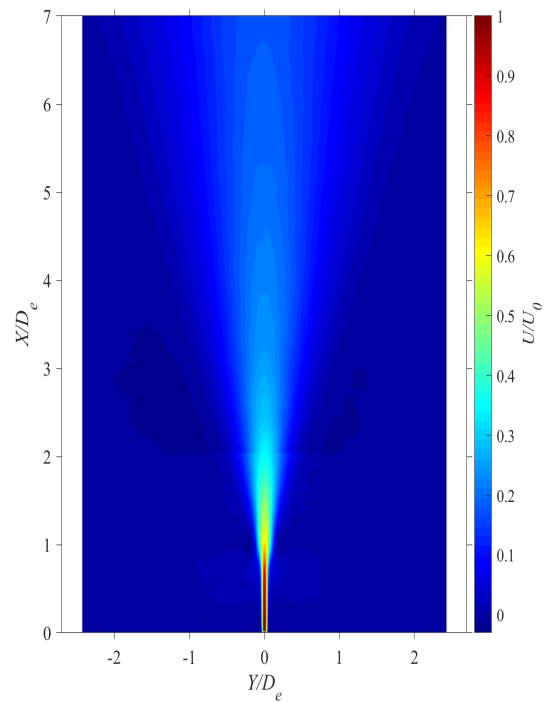
(a)



(b)

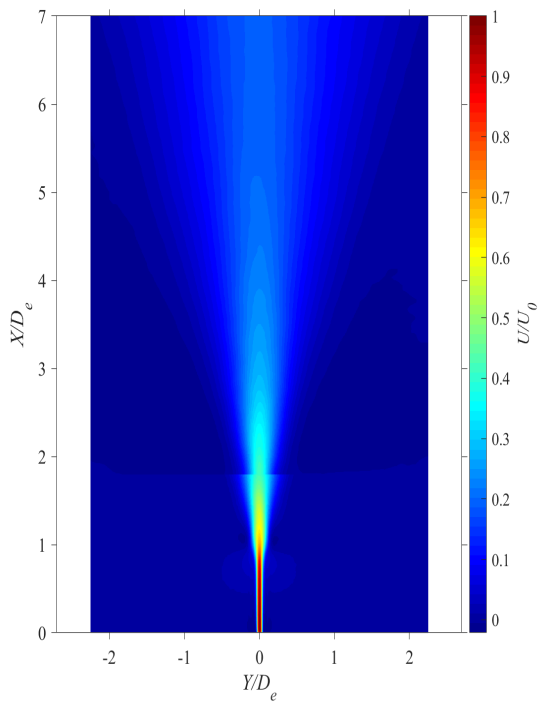


(c)

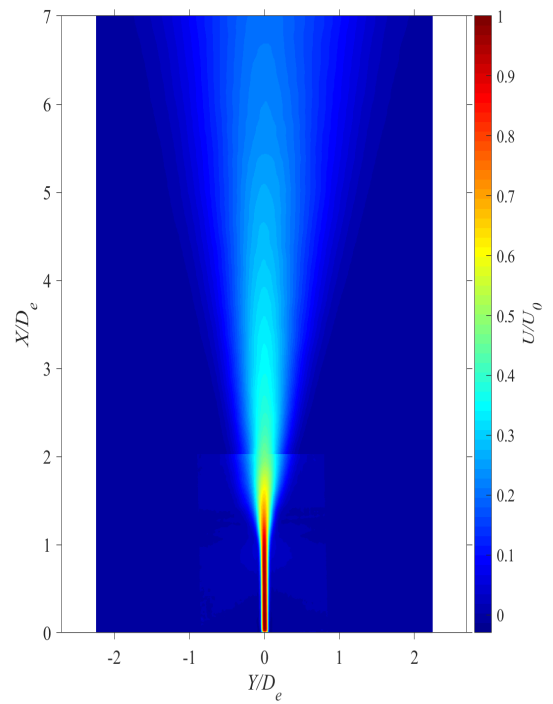


(d)

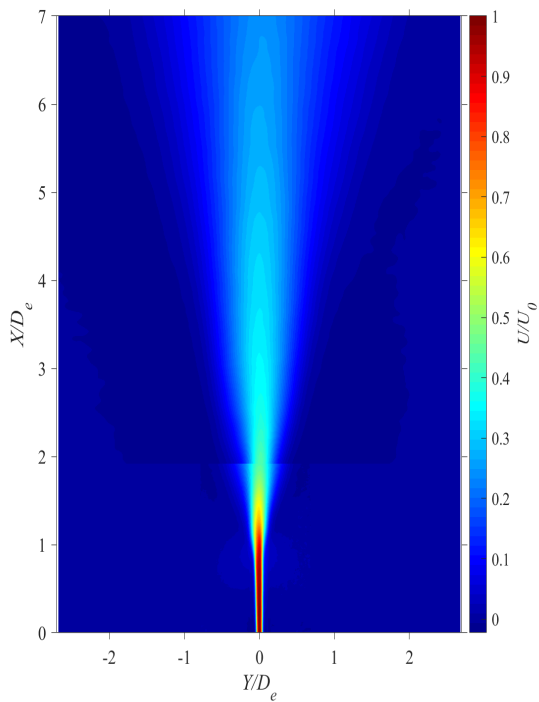
Figure D.16: Vector field plots showing magnitude at $Re = 10800$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



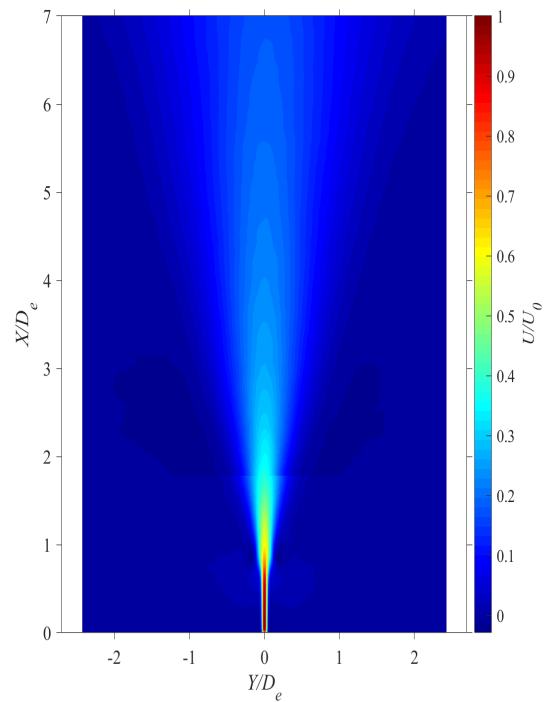
(a)



(b)



(c)



(d)

Figure D.17: Vector field plots showing magnitude at $Re = 14400$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

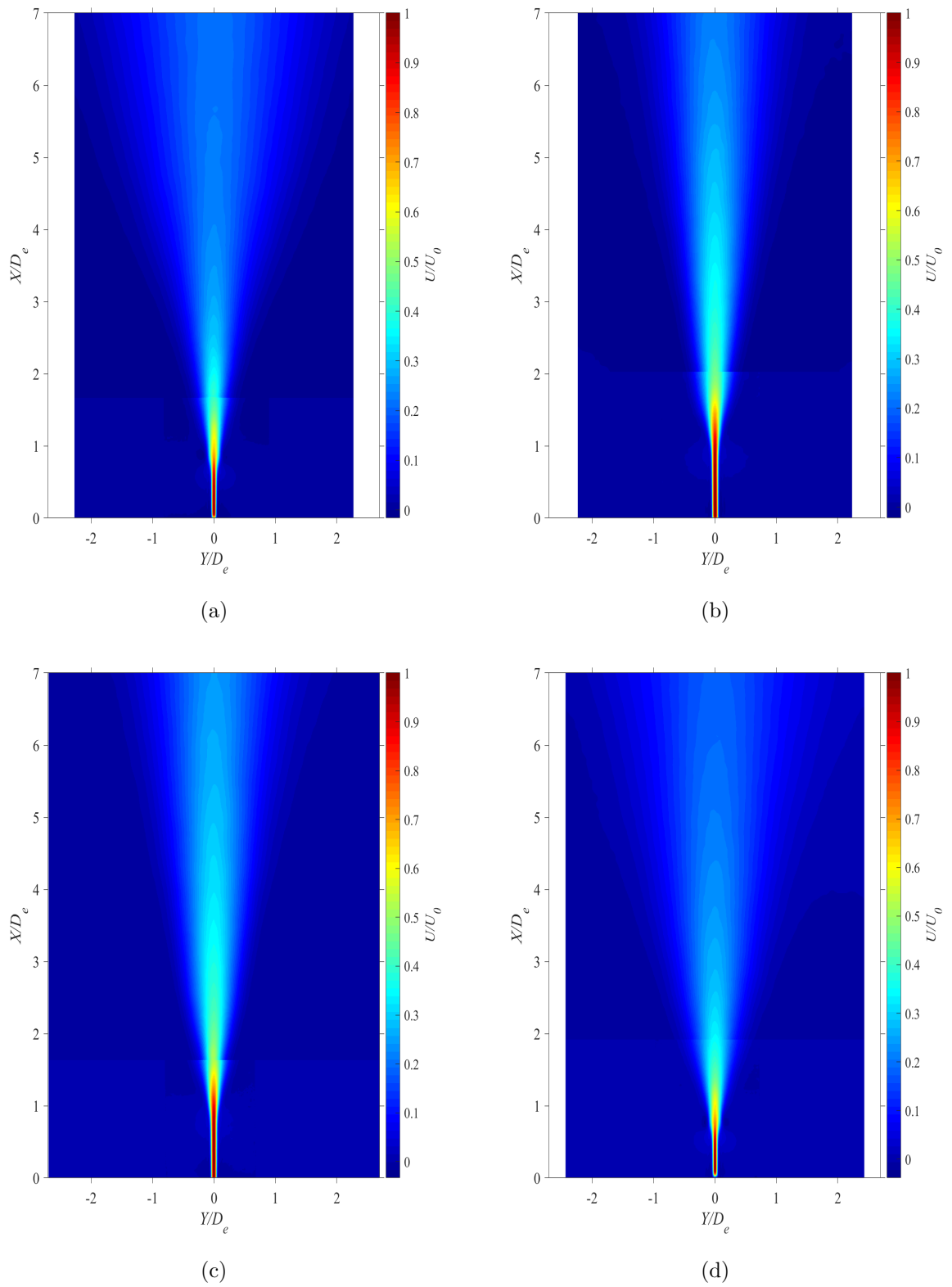
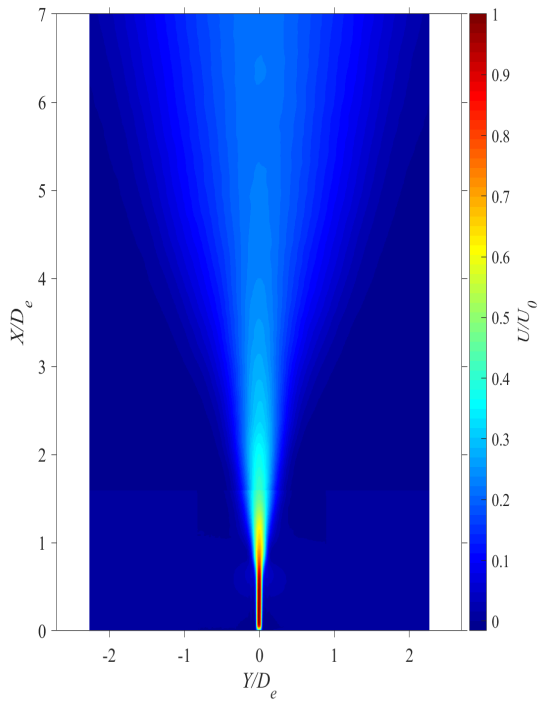
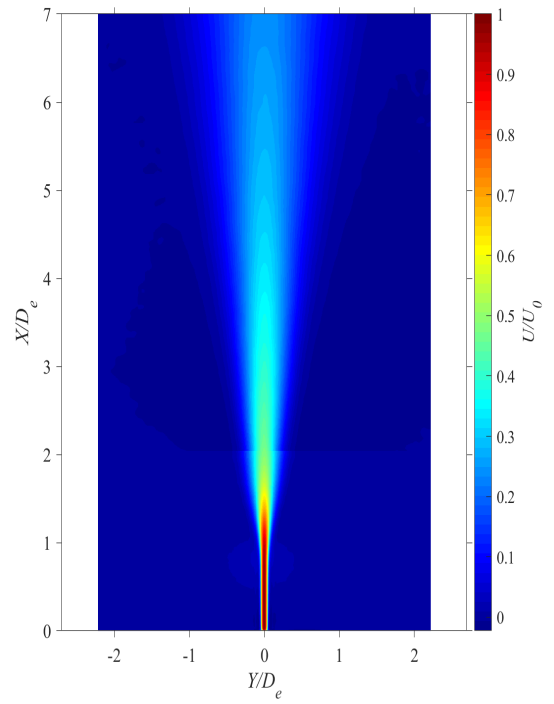


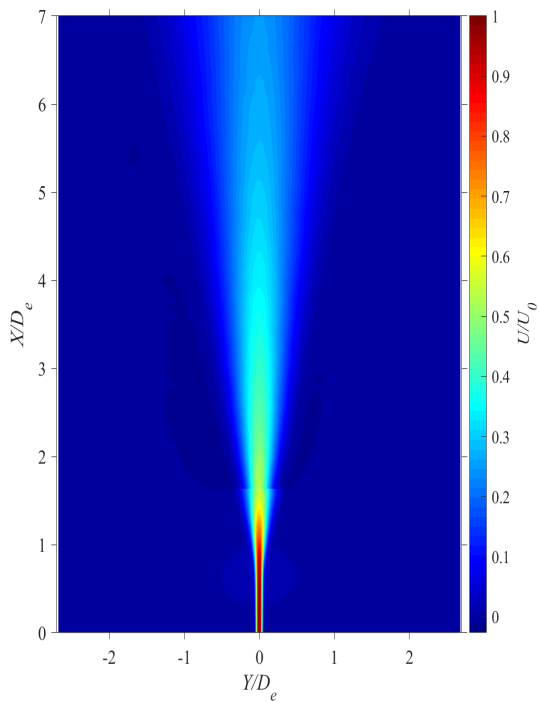
Figure D.18: Vector field plots showing magnitude at $Re = 18000$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



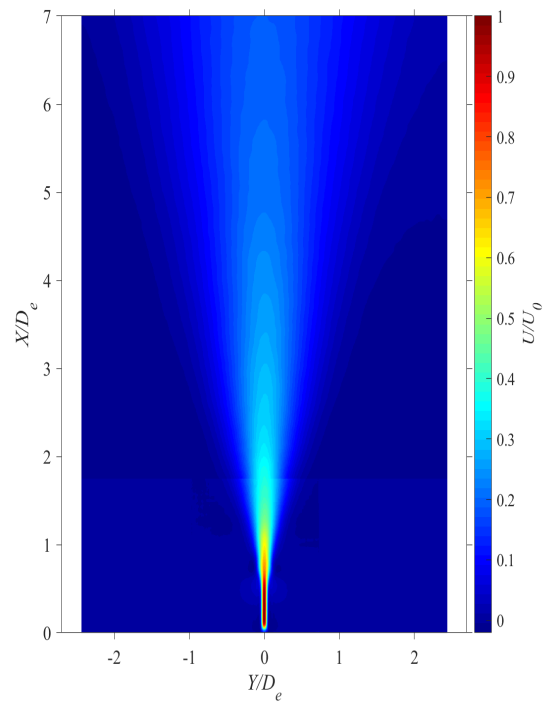
(a)



(b)



(c)



(d)

Figure D.19: Vector field plots showing magnitude at $Re = 21600$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

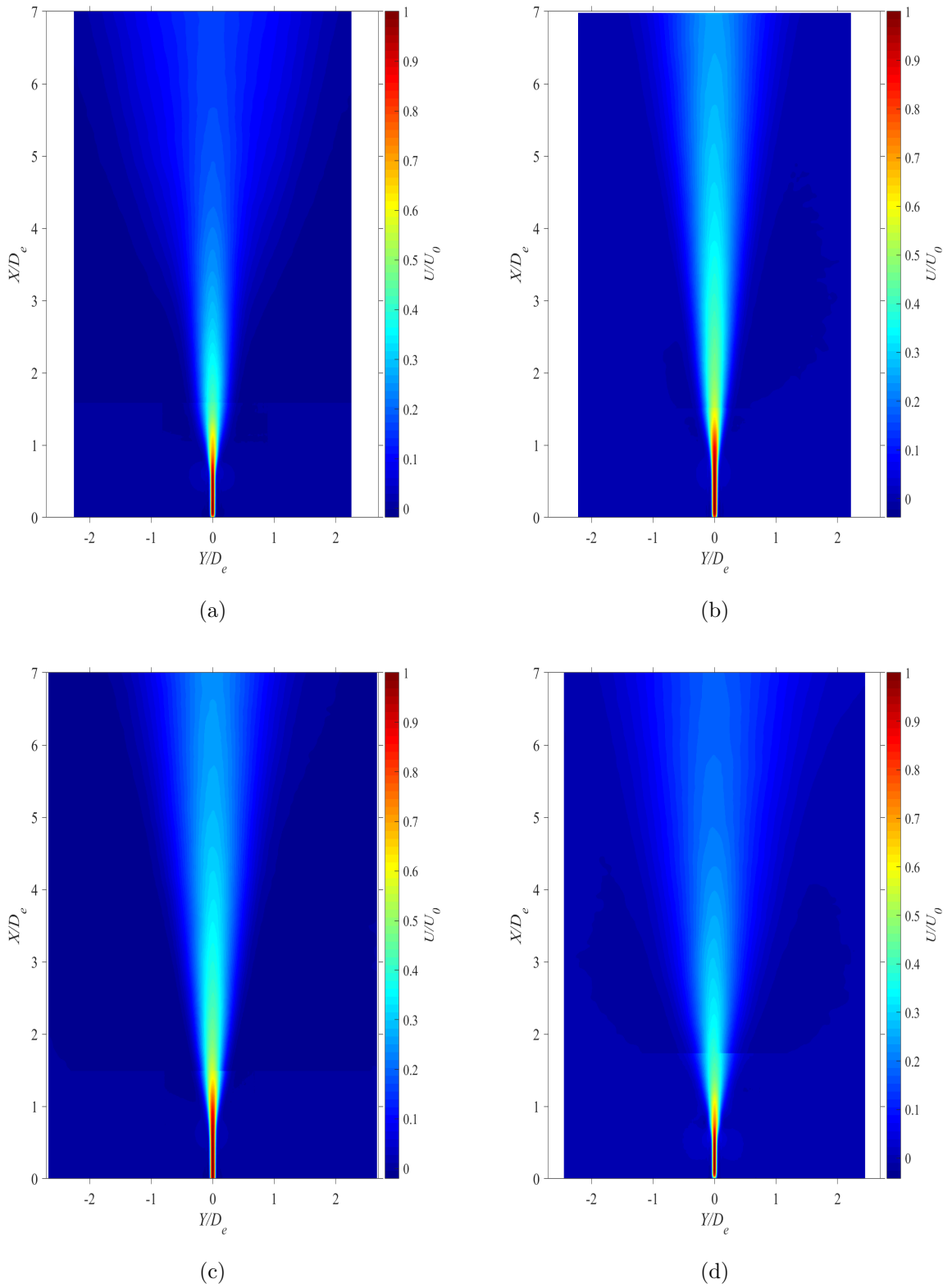


Figure D.20: Vector field plots showing magnitude at $Re = 25200$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

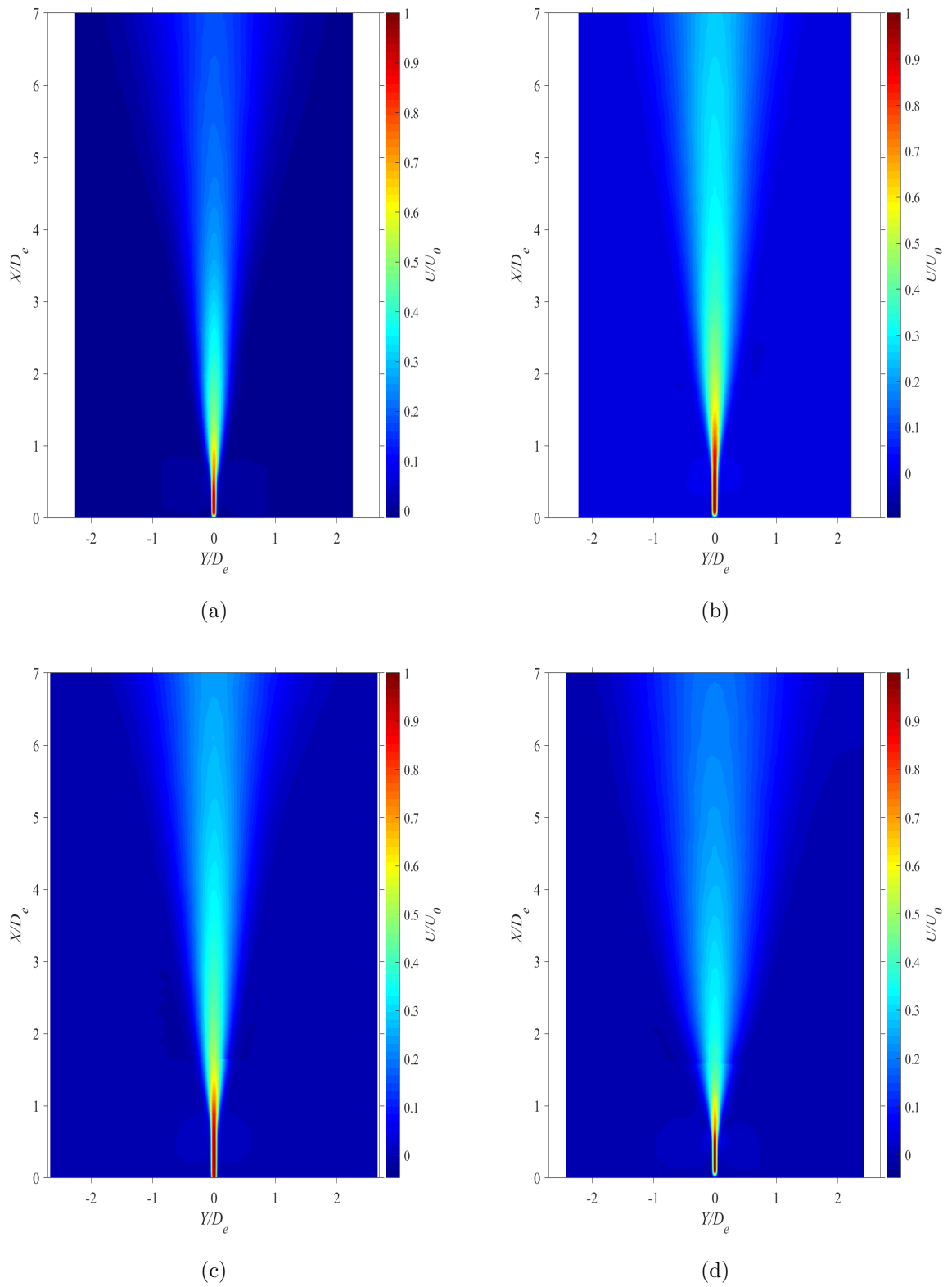
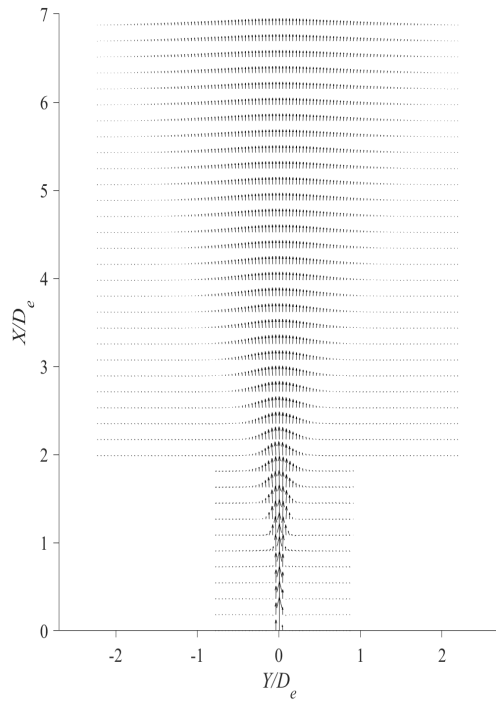
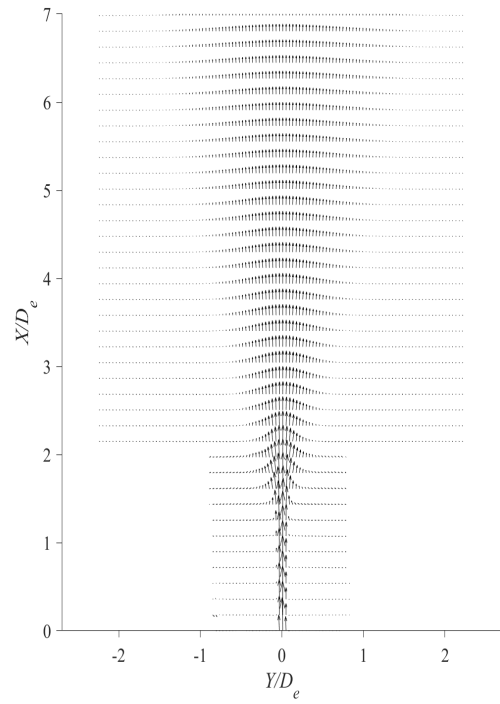


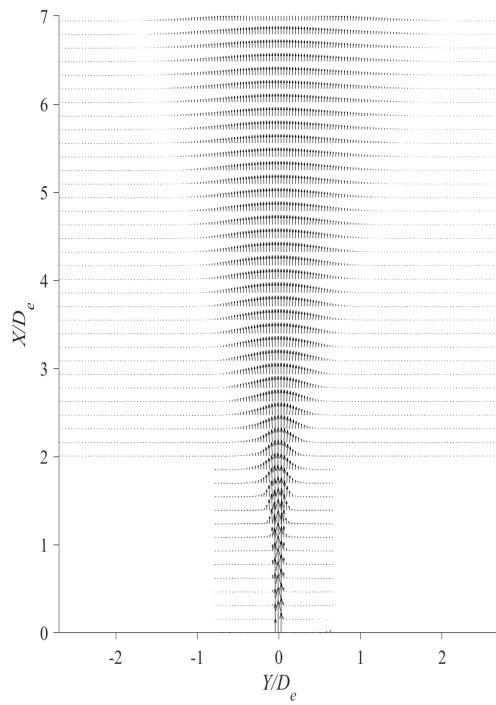
Figure D.21: Vector field plots showing magnitude at $Re = 28800$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



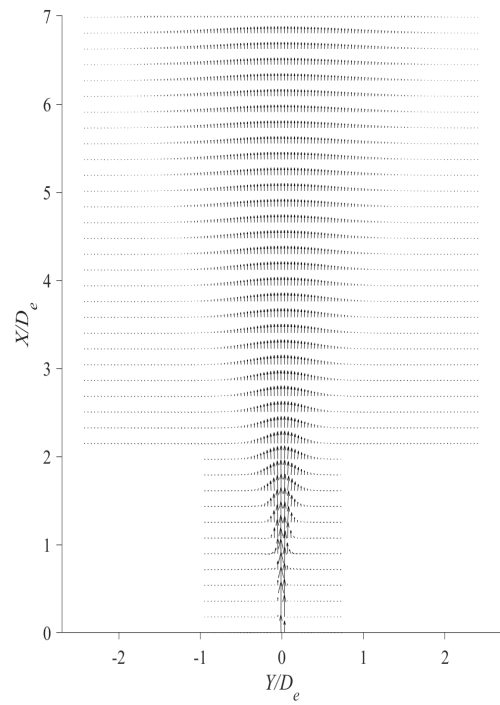
(a)



(b)

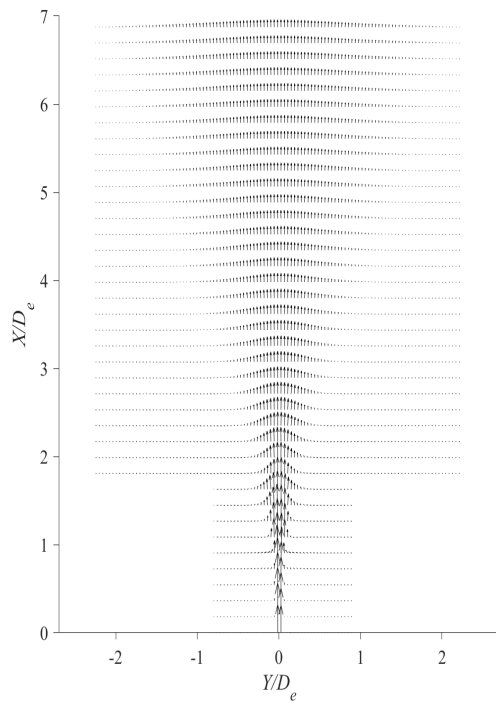


(c)

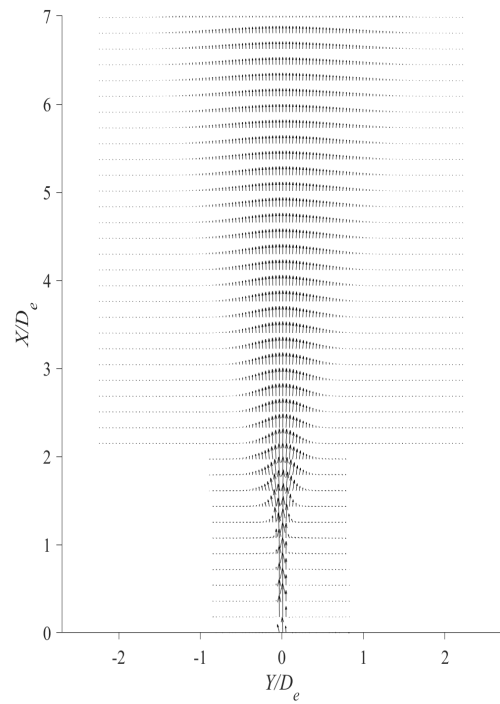


(d)

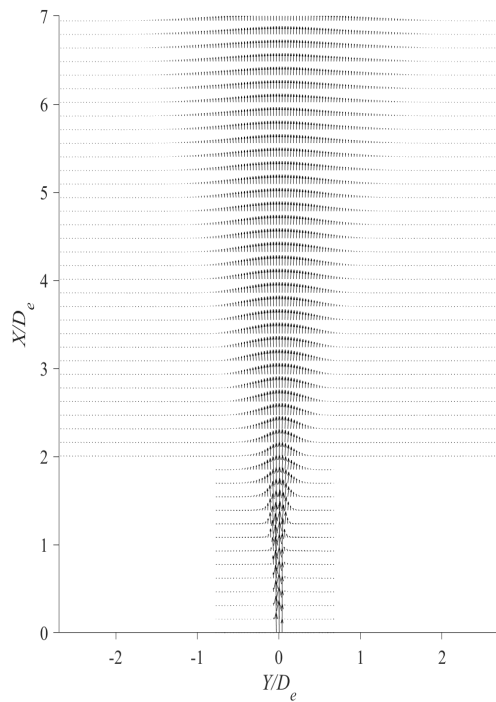
Figure D.22: Vector field plots showing direction at $Re = 10800$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



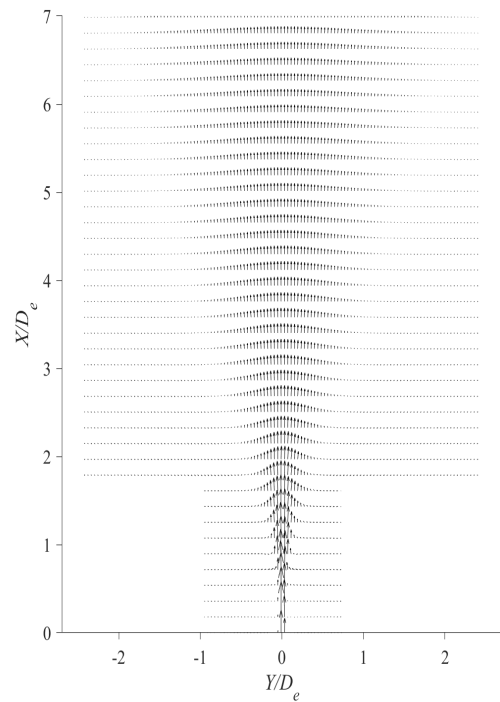
(a)



(b)

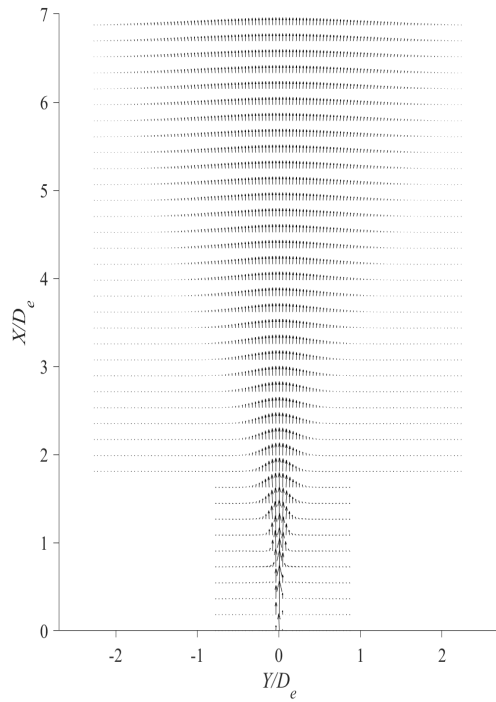


(c)

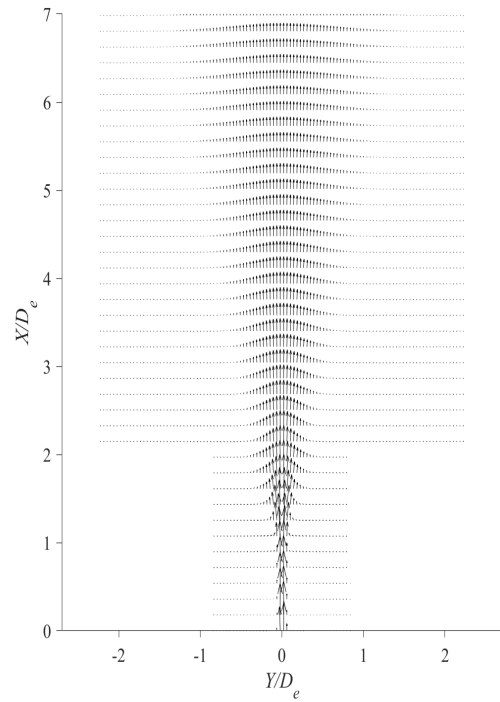


(d)

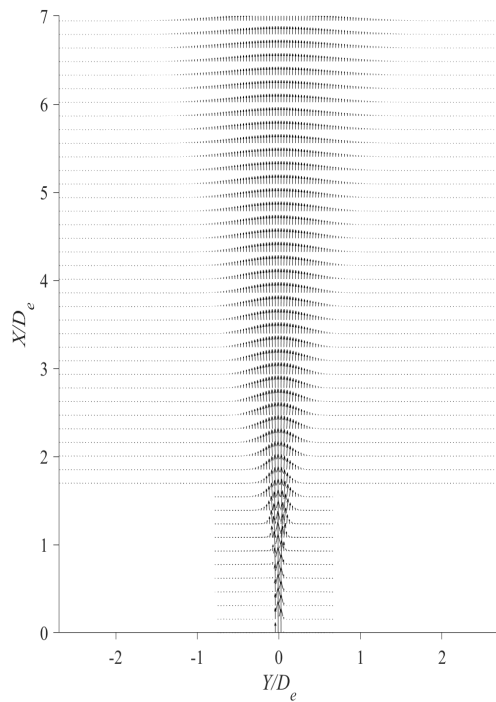
Figure D.23: Vector field plots showing direction at $Re = 14400$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



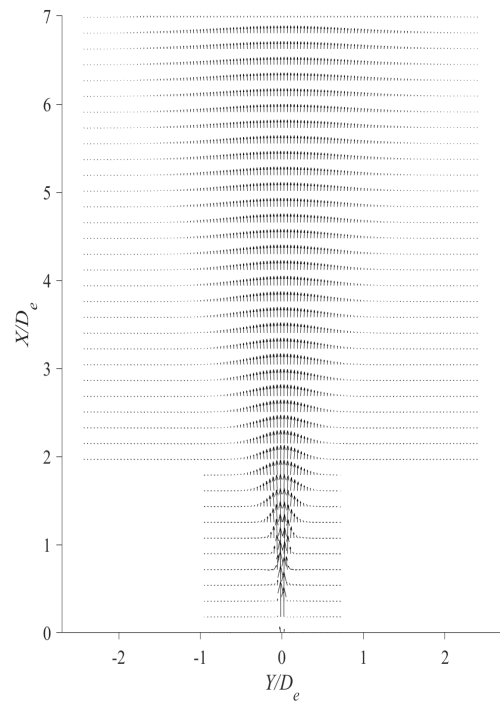
(a)



(b)

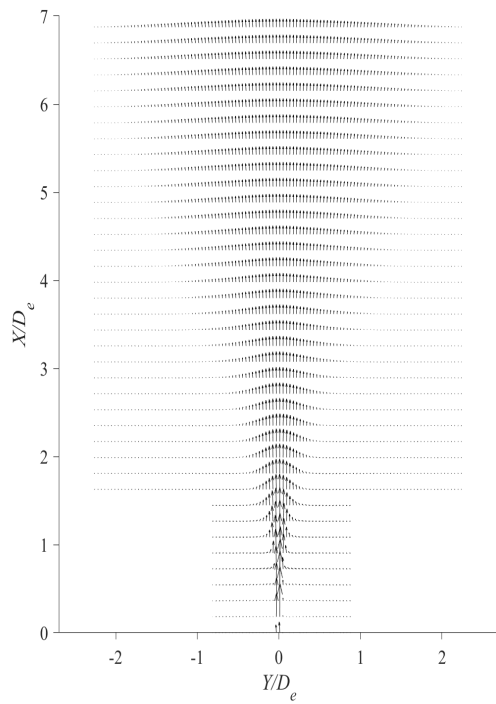


(c)

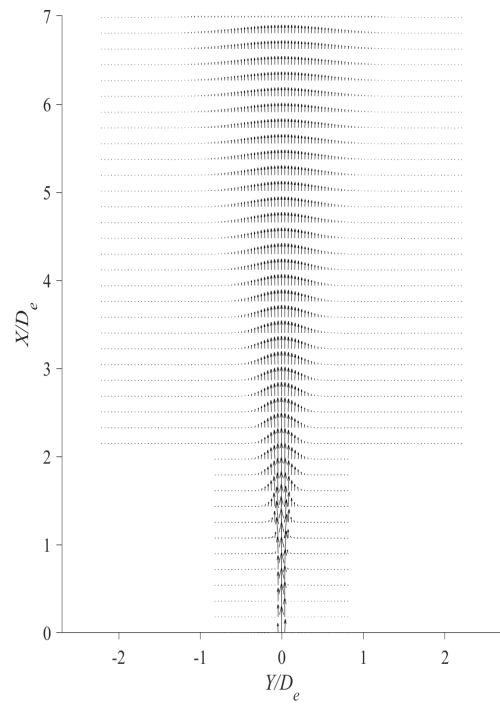


(d)

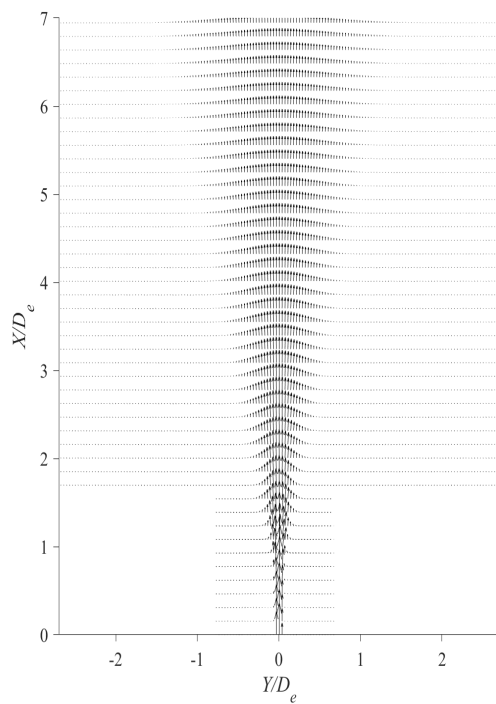
Figure D.24: Vector field plots showing direction at $Re = 18000$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



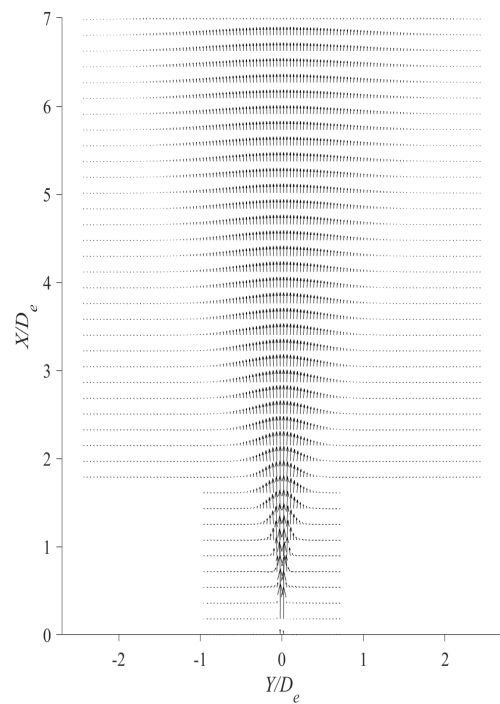
(a)



(b)

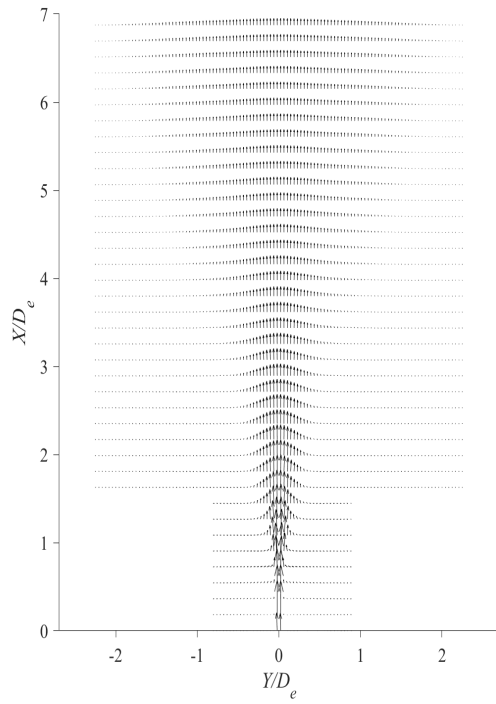


(c)

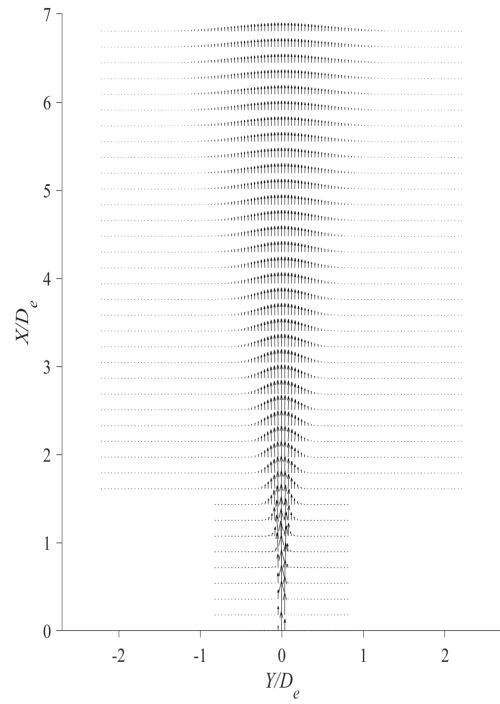


(d)

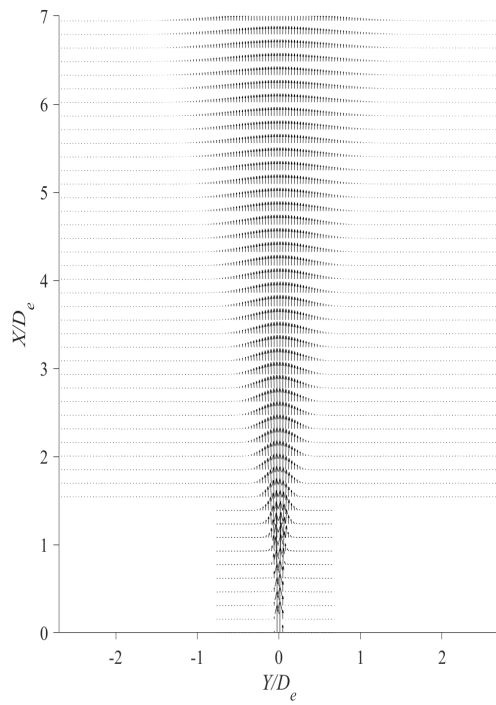
Figure D.25: Vector field plots showing direction at $Re = 21600$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



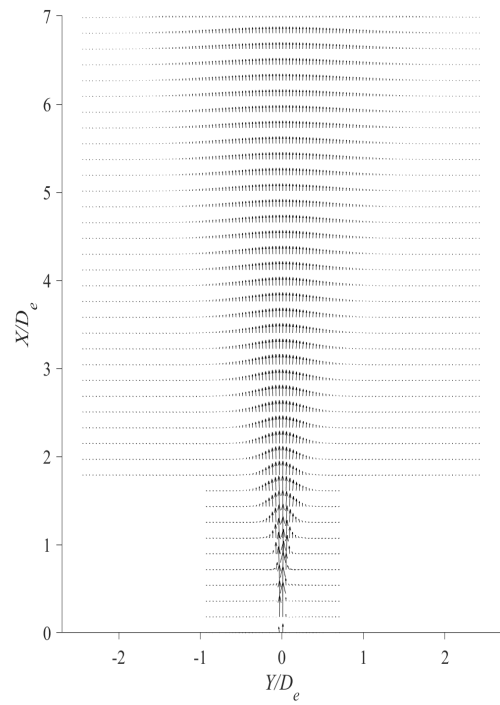
(a)



(b)

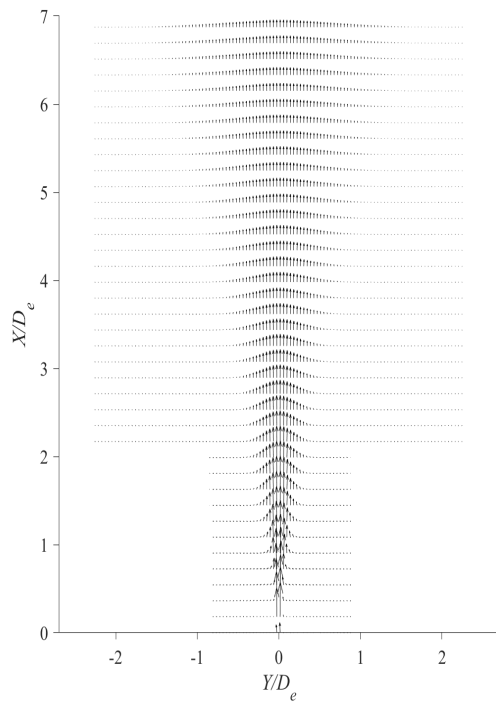


(c)

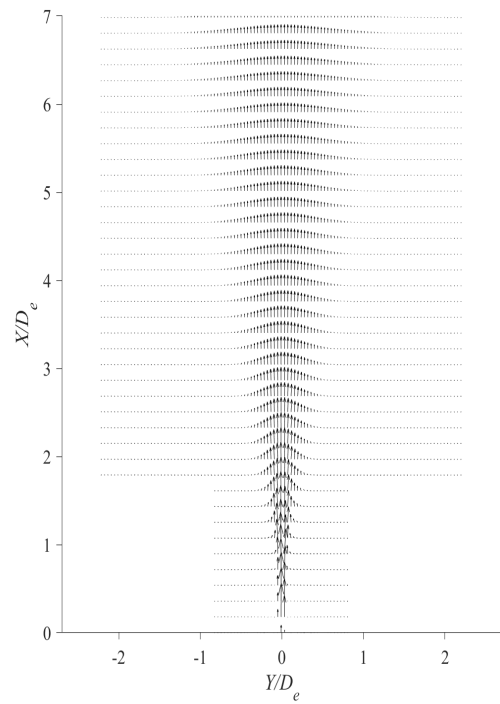


(d)

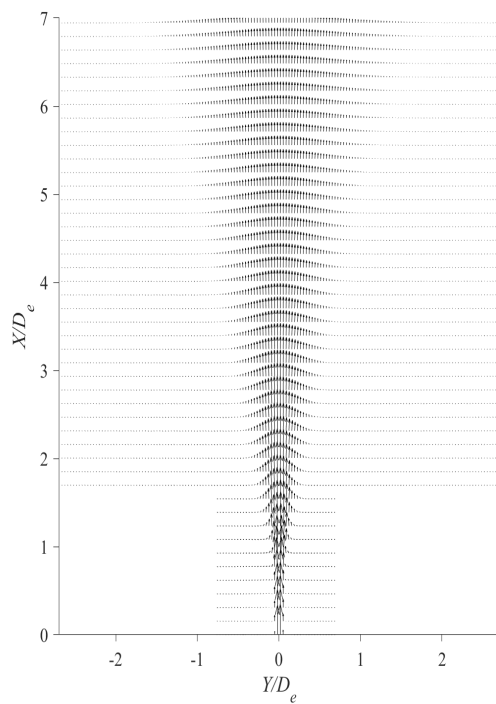
Figure D.26: Vector field plots showing direction at $Re = 25200$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



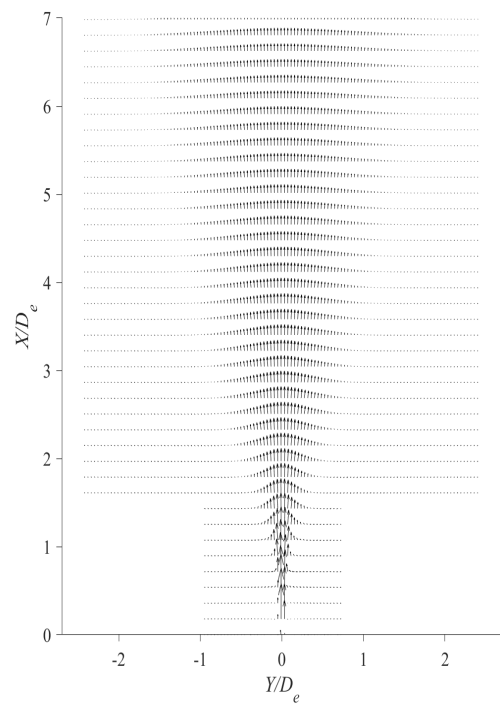
(a)



(b)



(c)



(d)

Figure D.27: Vector field plots showing direction at $Re = 28800$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

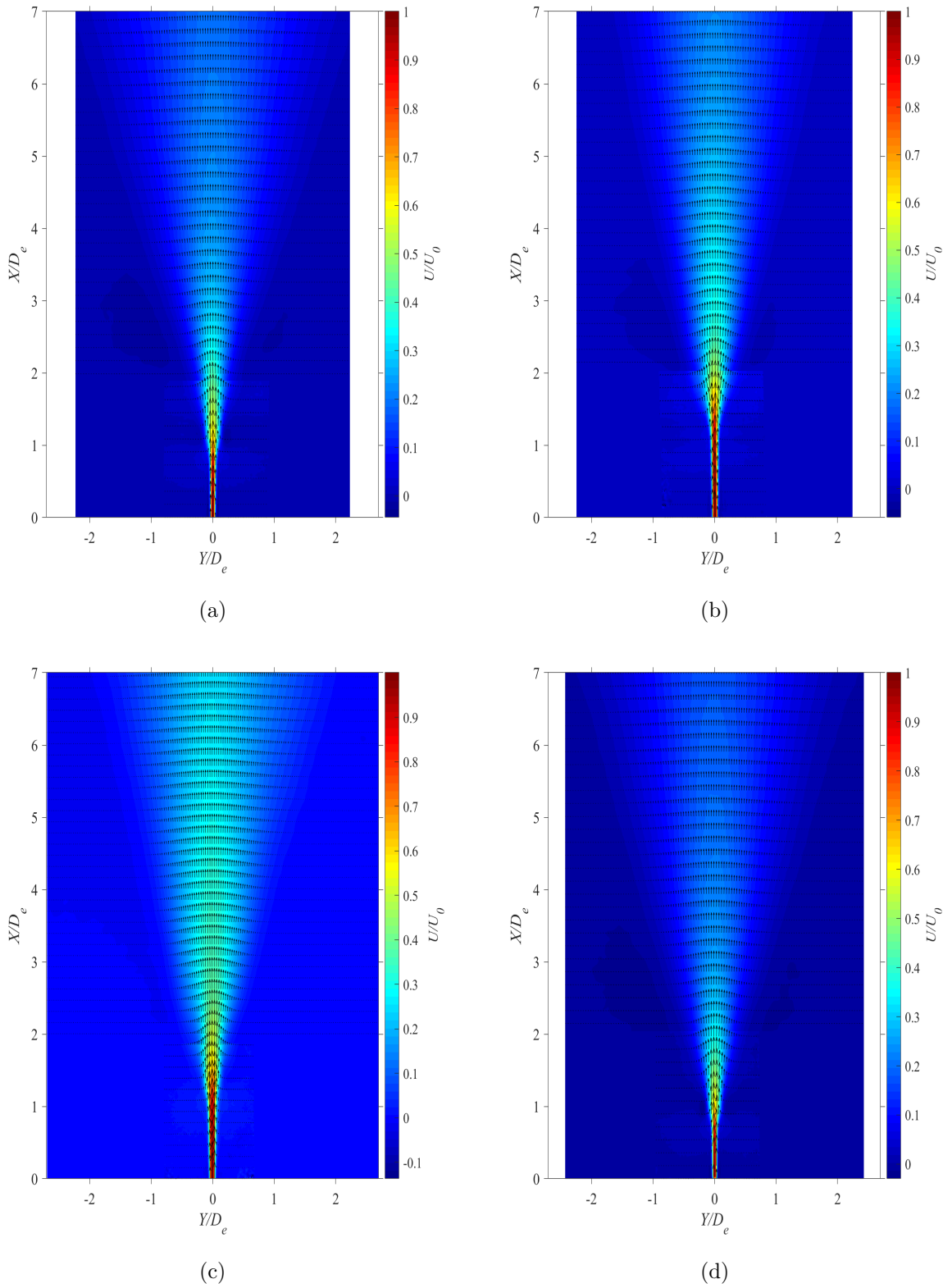
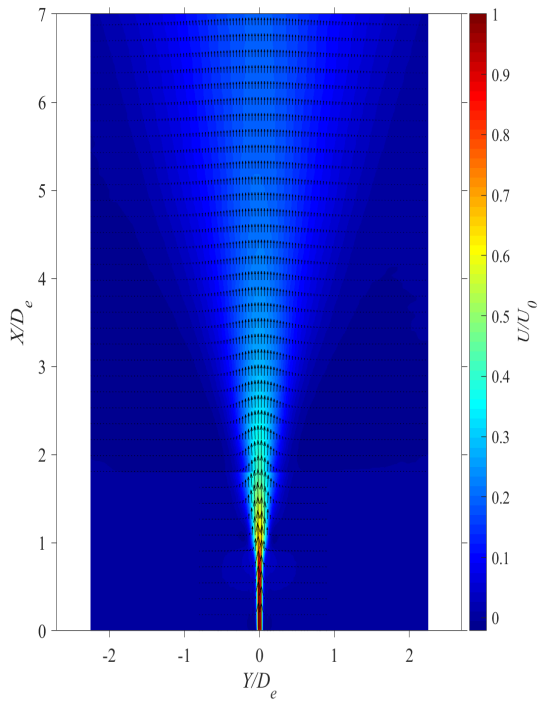
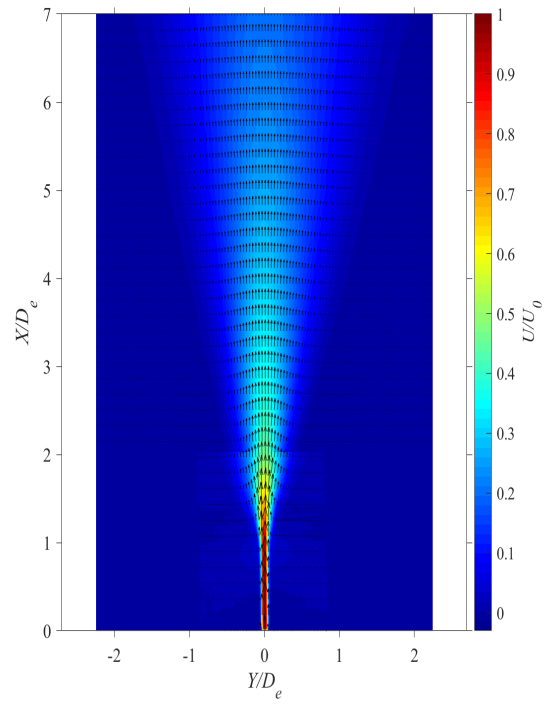


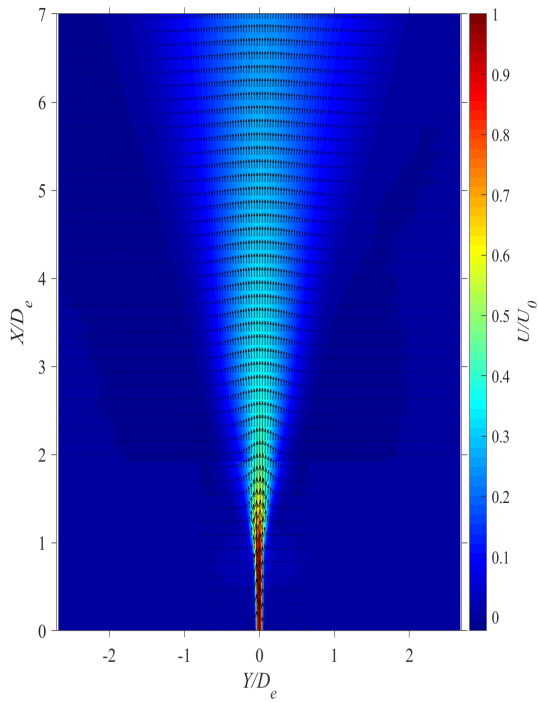
Figure D.28: Vector field plots showing magnitude and direction at $Re = 10800$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



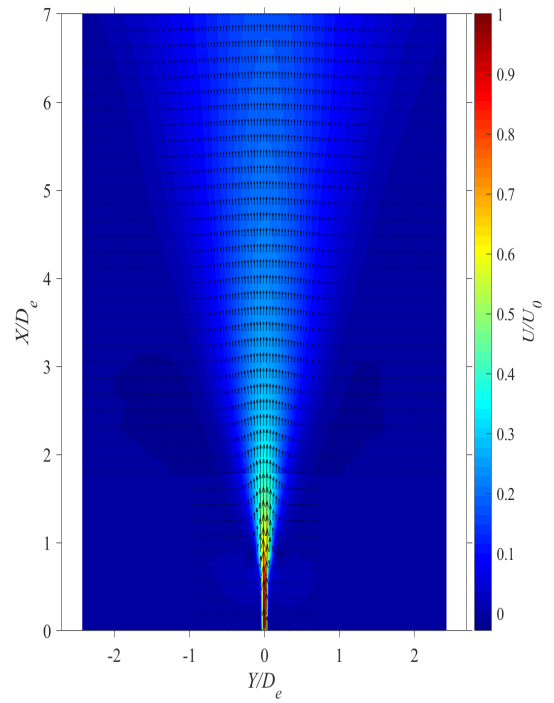
(a)



(b)



(c)



(d)

Figure D.29: Vector field plots showing magnitude and direction at $Re = 14400$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

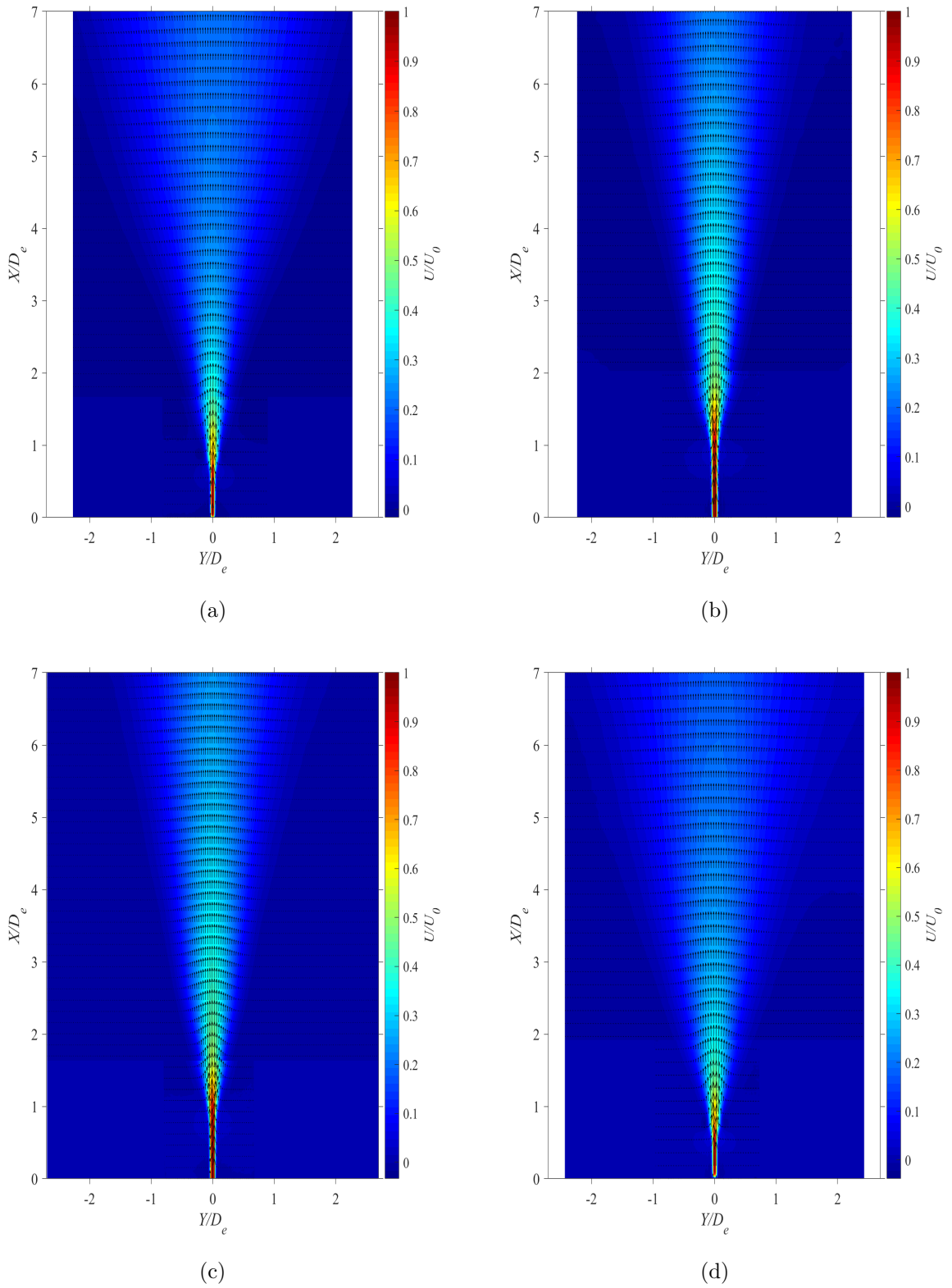
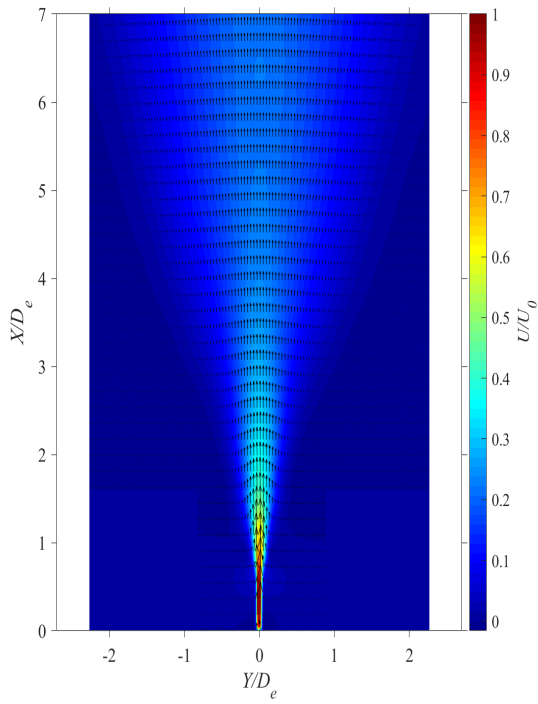
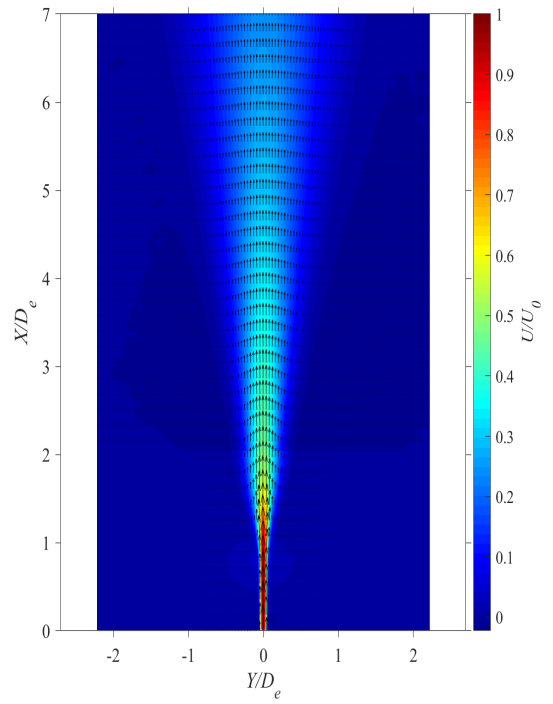


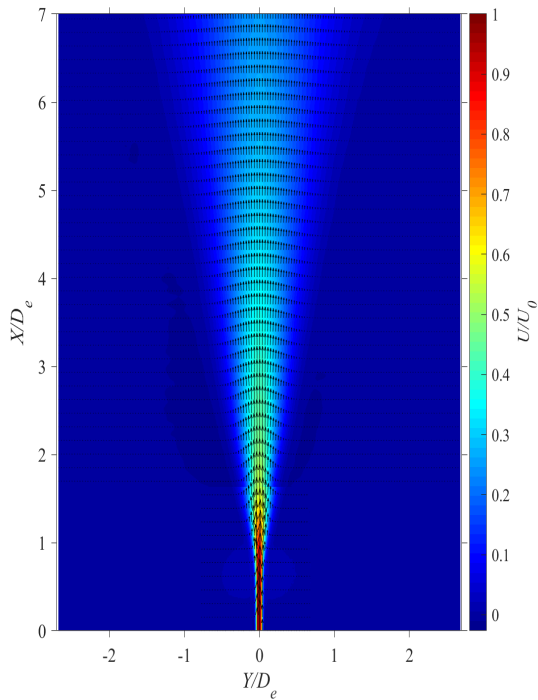
Figure D.30: Vector field plots showing magnitude and direction at $Re = 18000$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



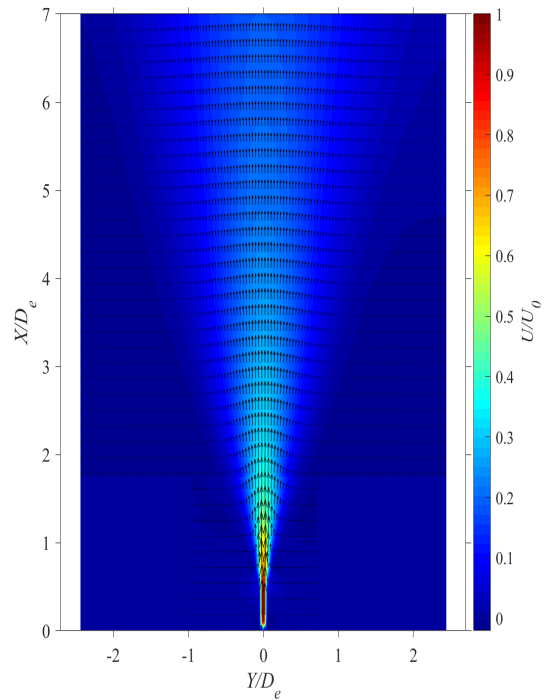
(a)



(b)



(c)



(d)

Figure D.31: Vector field plots showing magnitude and direction at $Re = 21600$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

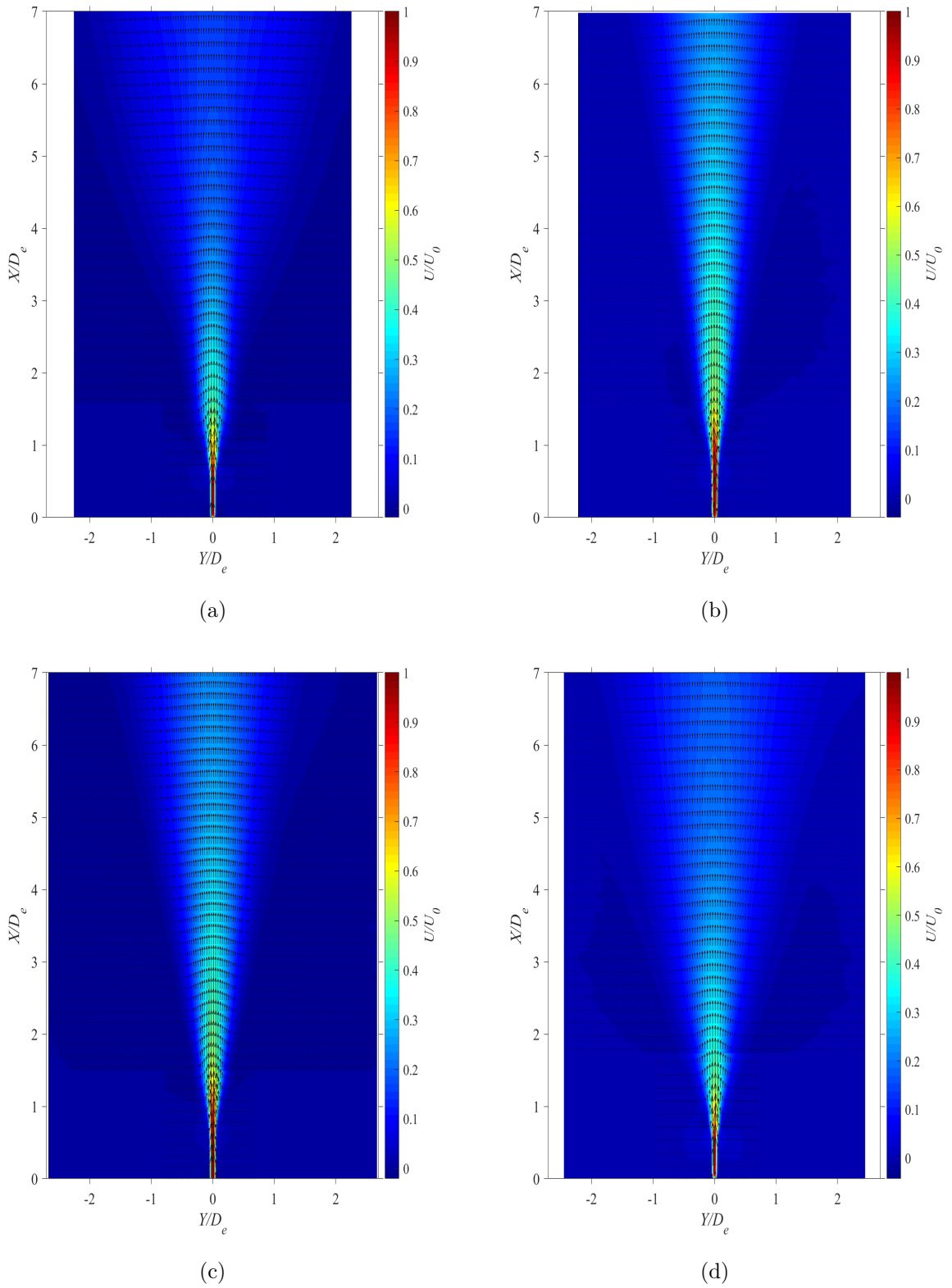
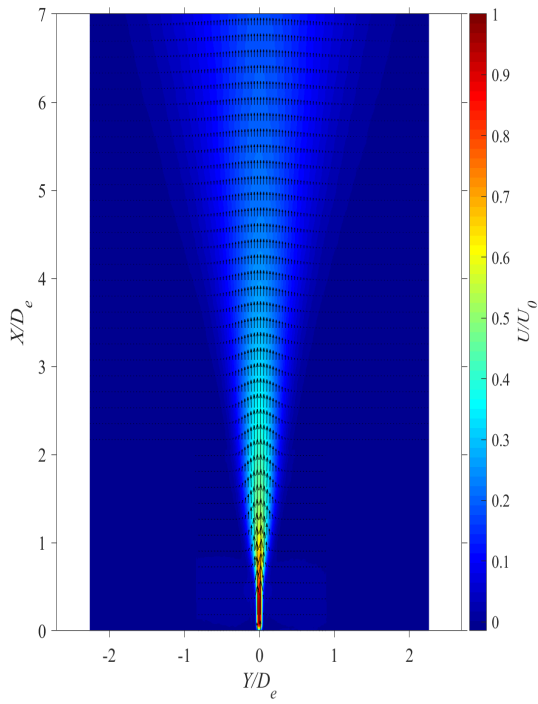
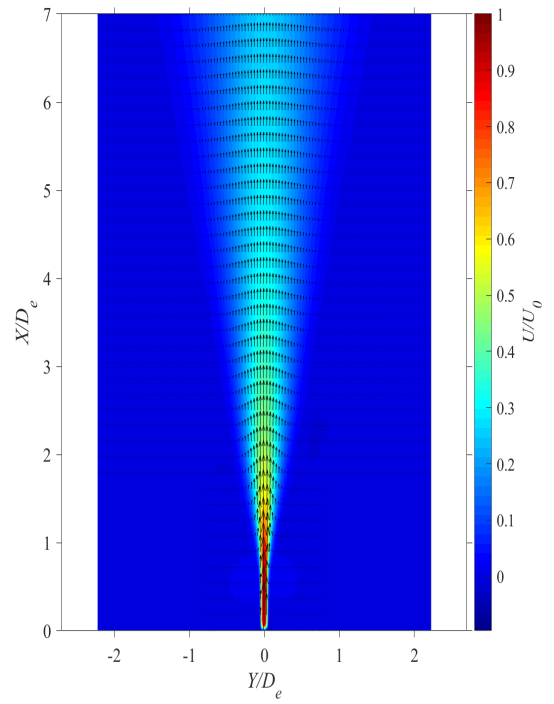


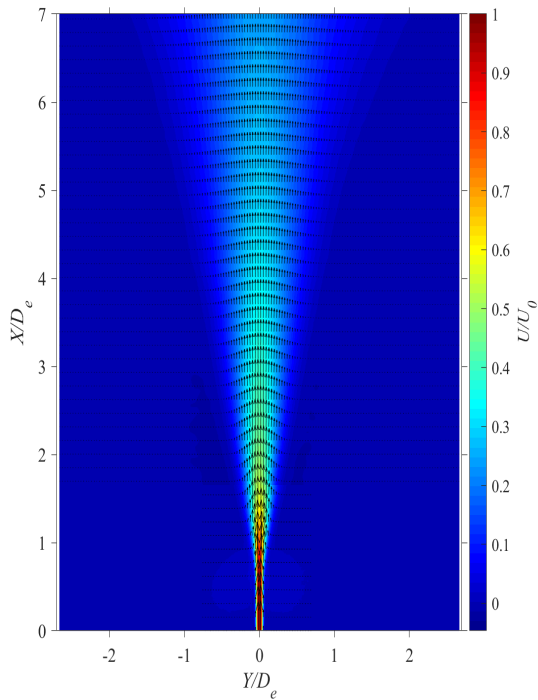
Figure D.32: Vector field plots showing magnitude and direction at $Re = 25200$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$



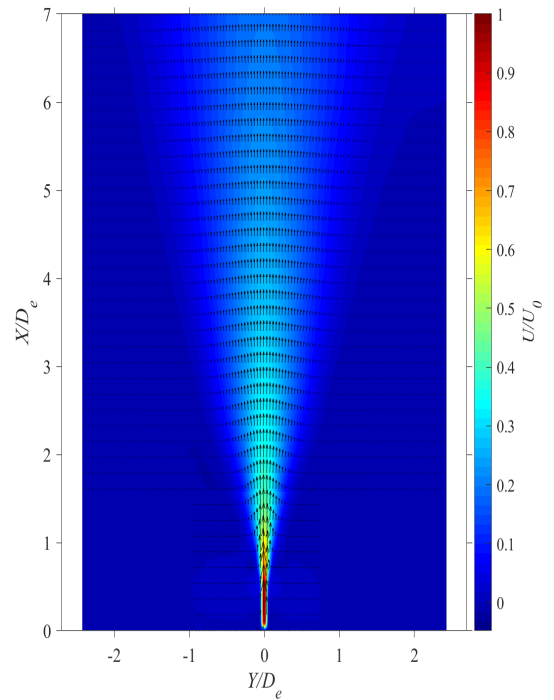
(a)



(b)



(c)



(d)

Figure D.33: Vector field plots showing magnitude and direction at $Re = 28800$ for (a) $AR = 100$, (b) $AR = 150$, (c) $AR = 200$, and (d) $AR = 210$

E Stereo-PIV Figures

This appendix contains all of the figures that were generated from the stereo-PIV experiments. The figures are presented in the following order:

- The whole-field streamwise velocity plots (pp.293-304)
- The whole-field lateral velocity plots (pp.305-316)
- The whole-field spanwise velocity plots (pp.317-328)
- The whole-field streamwise turbulence intensity plots (pp.329-340)
- The whole-field lateral turbulence intensity plots (pp.341-352)
- The whole-field spanwise turbulence intensity plots (pp.353-364)
- The whole-field lateral Reynolds shear stress plots (pp.365-376)
- The whole-field spanwise Reynolds shear stress plots (pp.377-388)
- The whole-field turbulent kinetic energy plots (pp.389-400)
- The whole-field streamwise vorticity plots (pp.401-412)
- The jet boundary plots (pp.413-416)

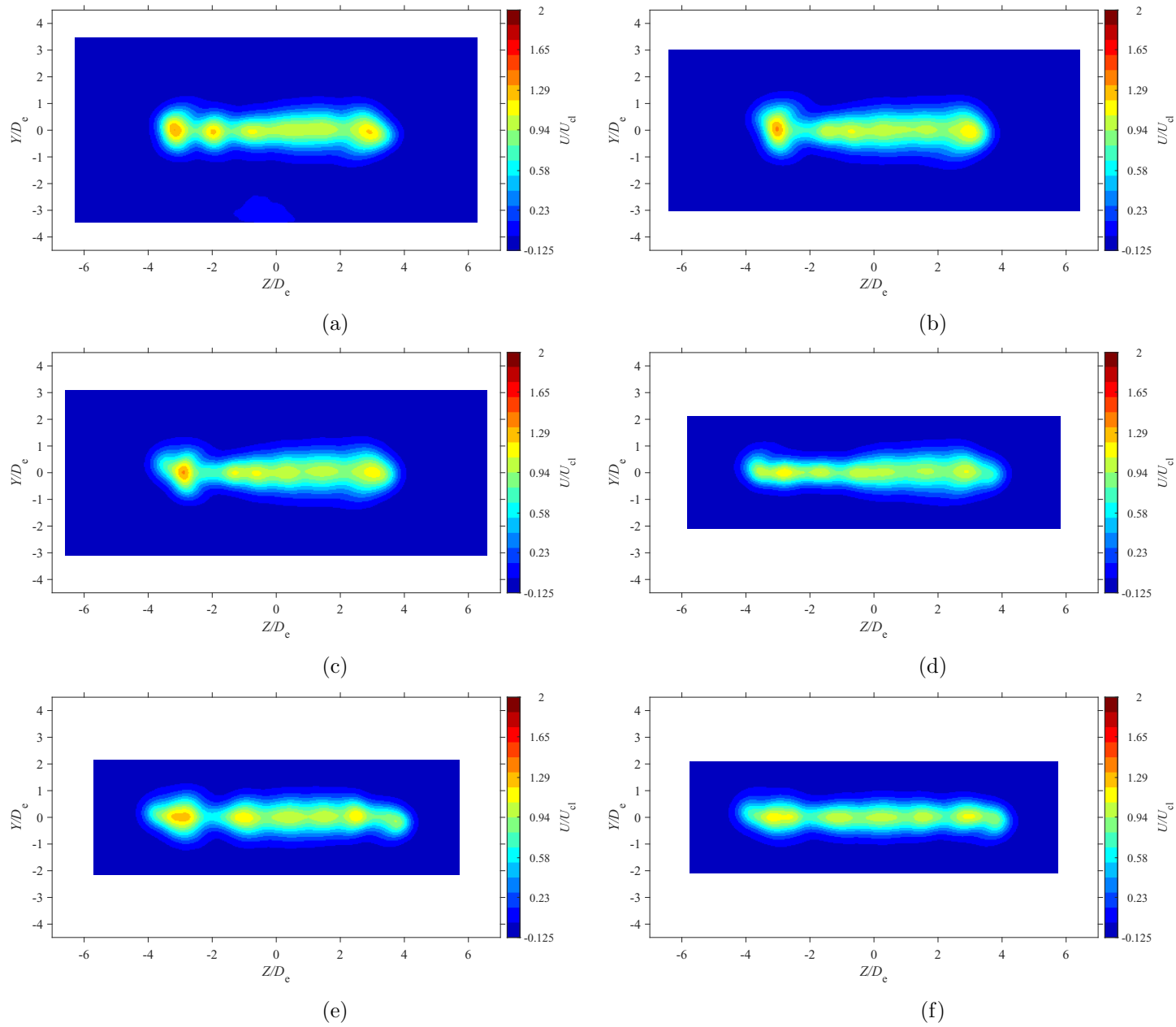


Figure E.1: Average streamwise velocity map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

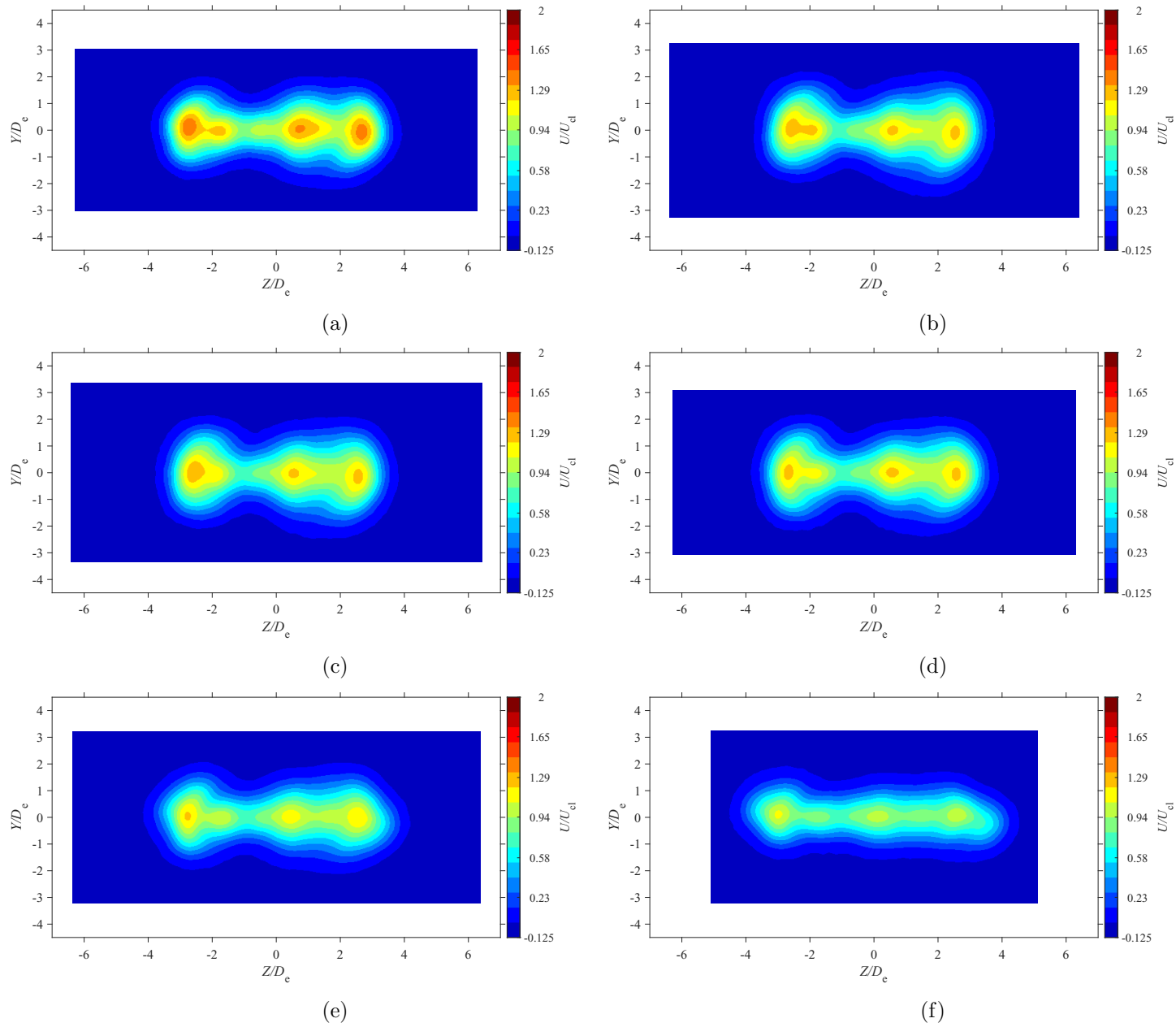


Figure E.2: Average streamwise velocity map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

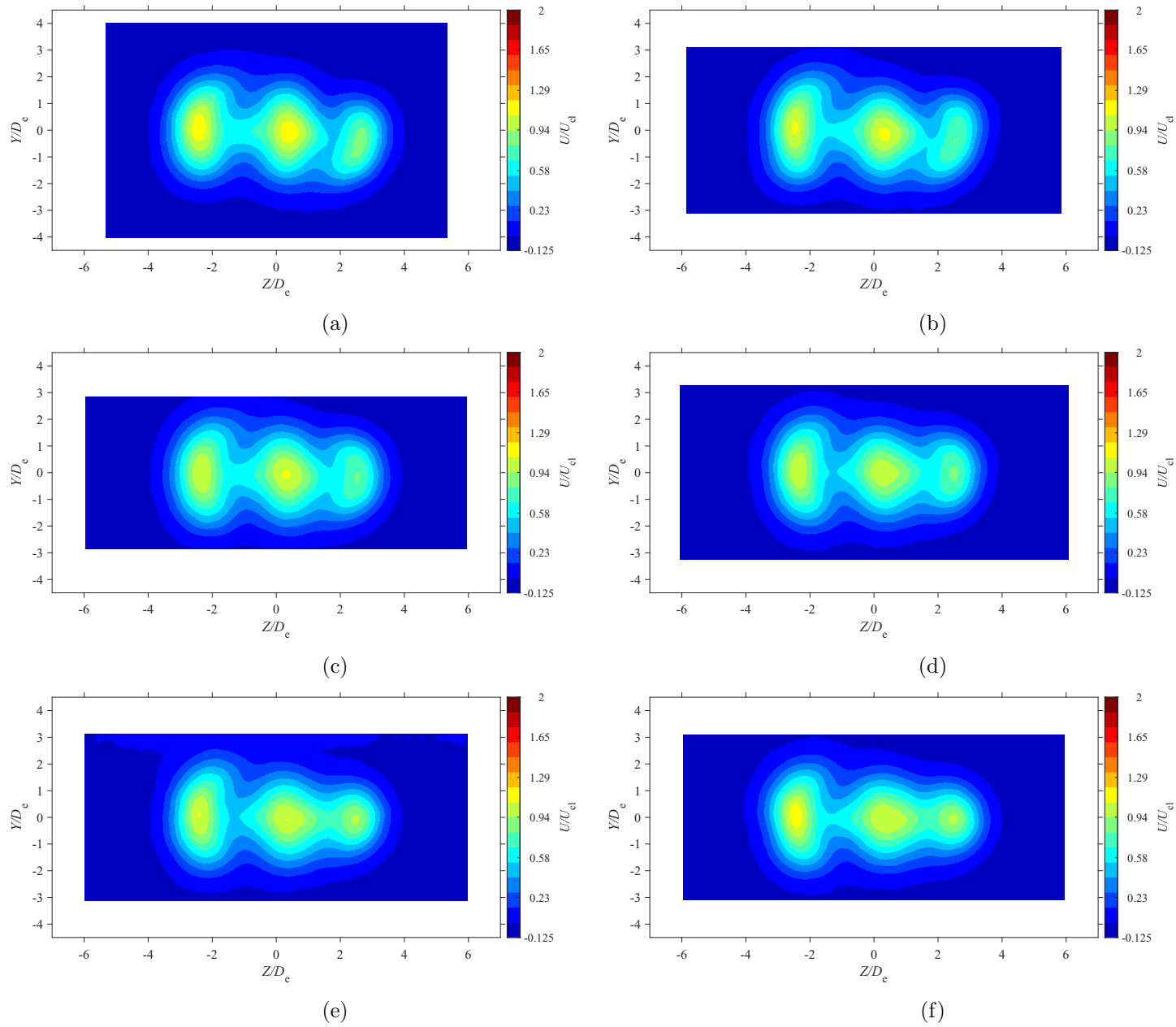


Figure E.3: Average streamwise velocity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

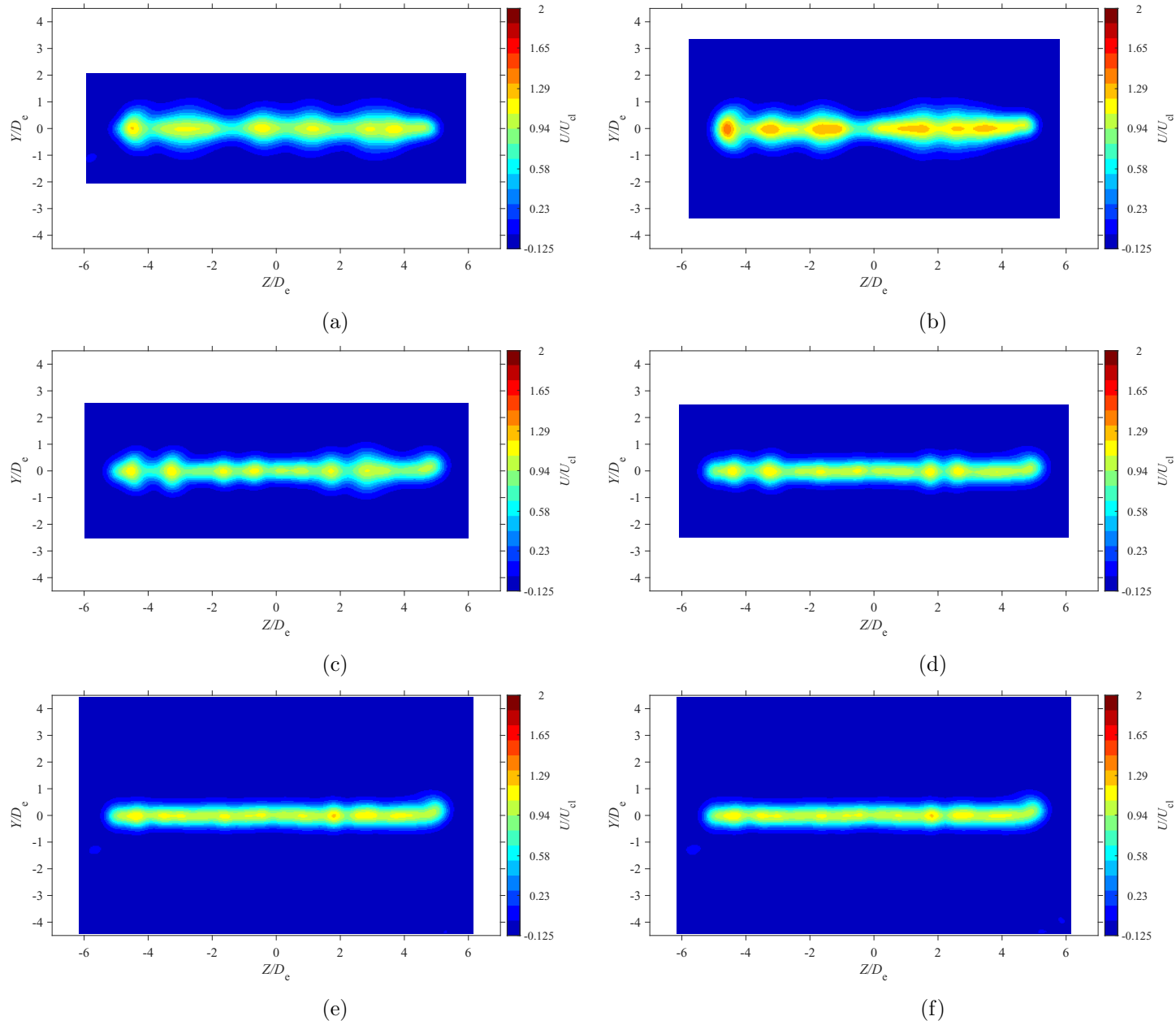


Figure E.4: Average streamwise velocity map at location $X = 3De$ for $AR = 15$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

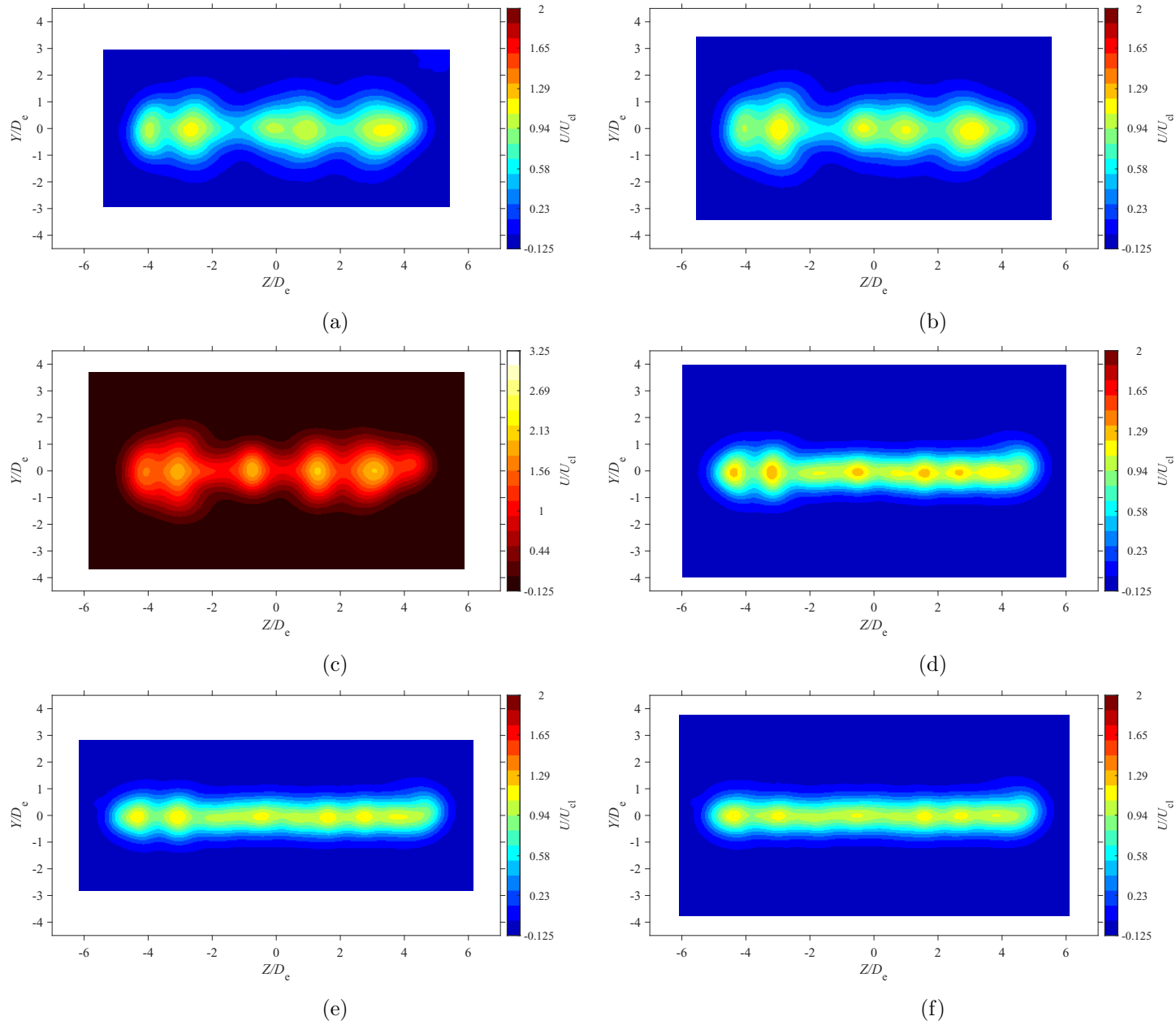


Figure E.5: Average streamwise velocity map at location $X = 5De$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

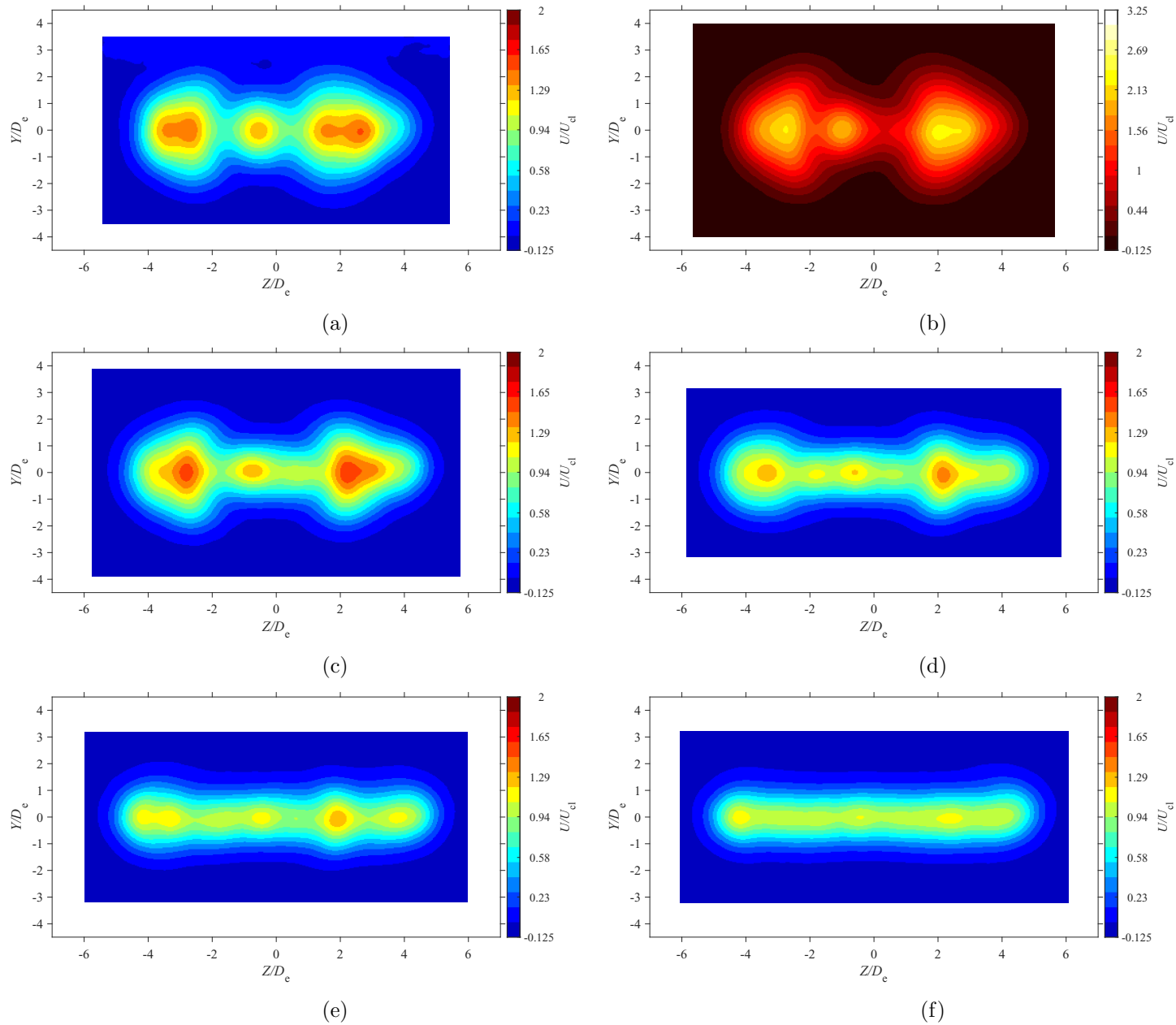


Figure E.6: Average streamwise velocity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

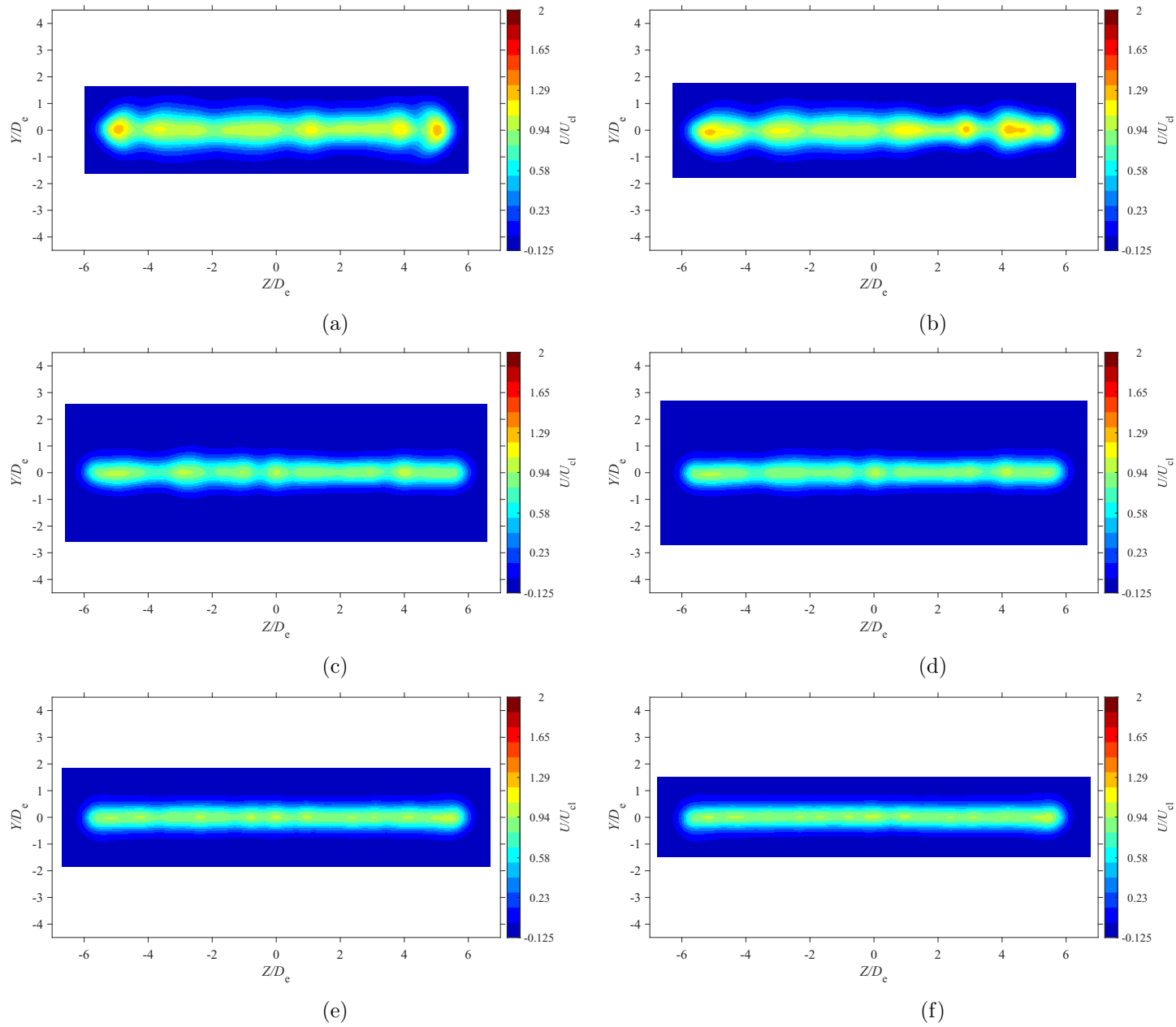


Figure E.7: Average streamwise velocity map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

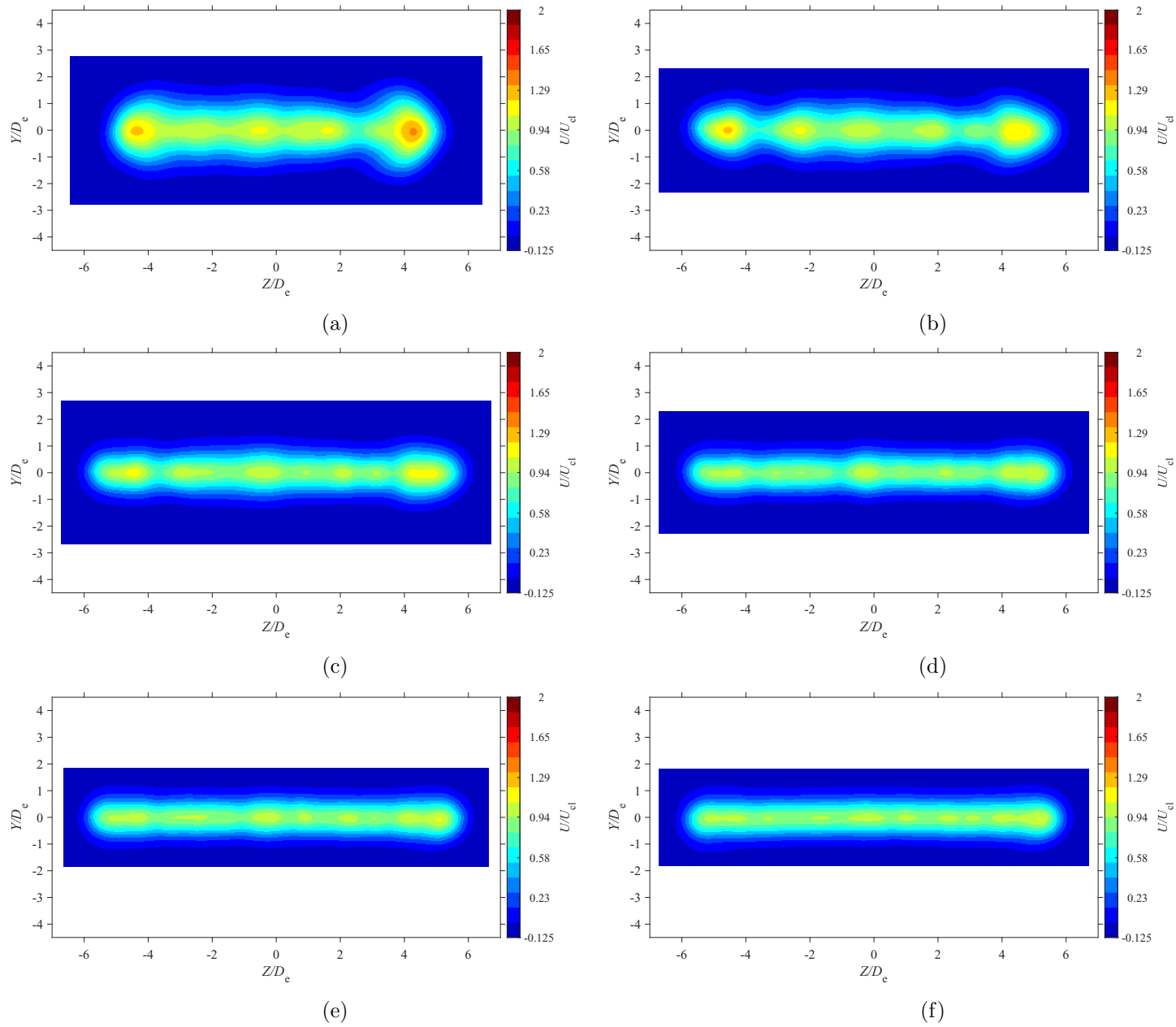


Figure E.8: Average streamwise velocity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

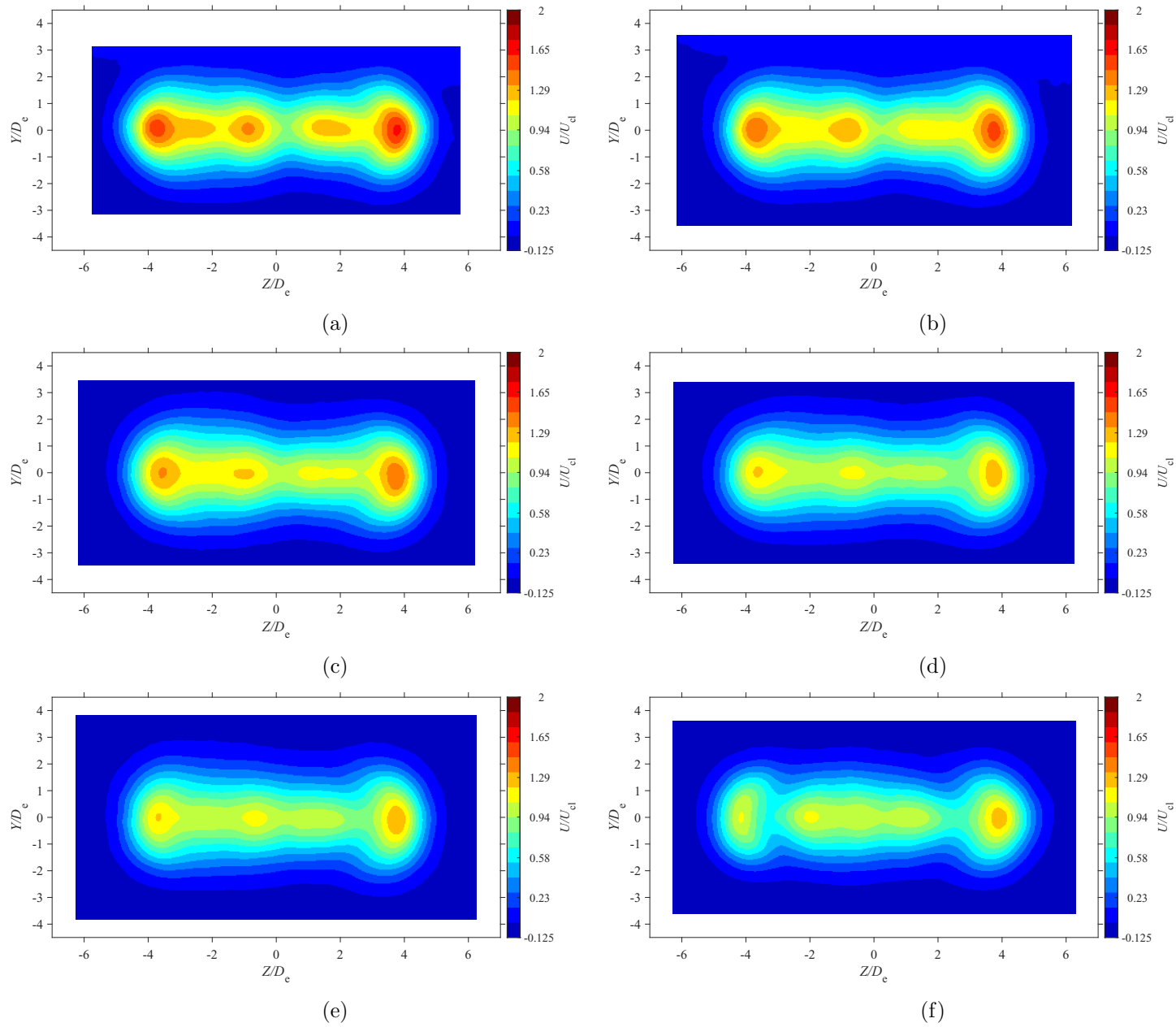


Figure E.9: Average streamwise velocity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

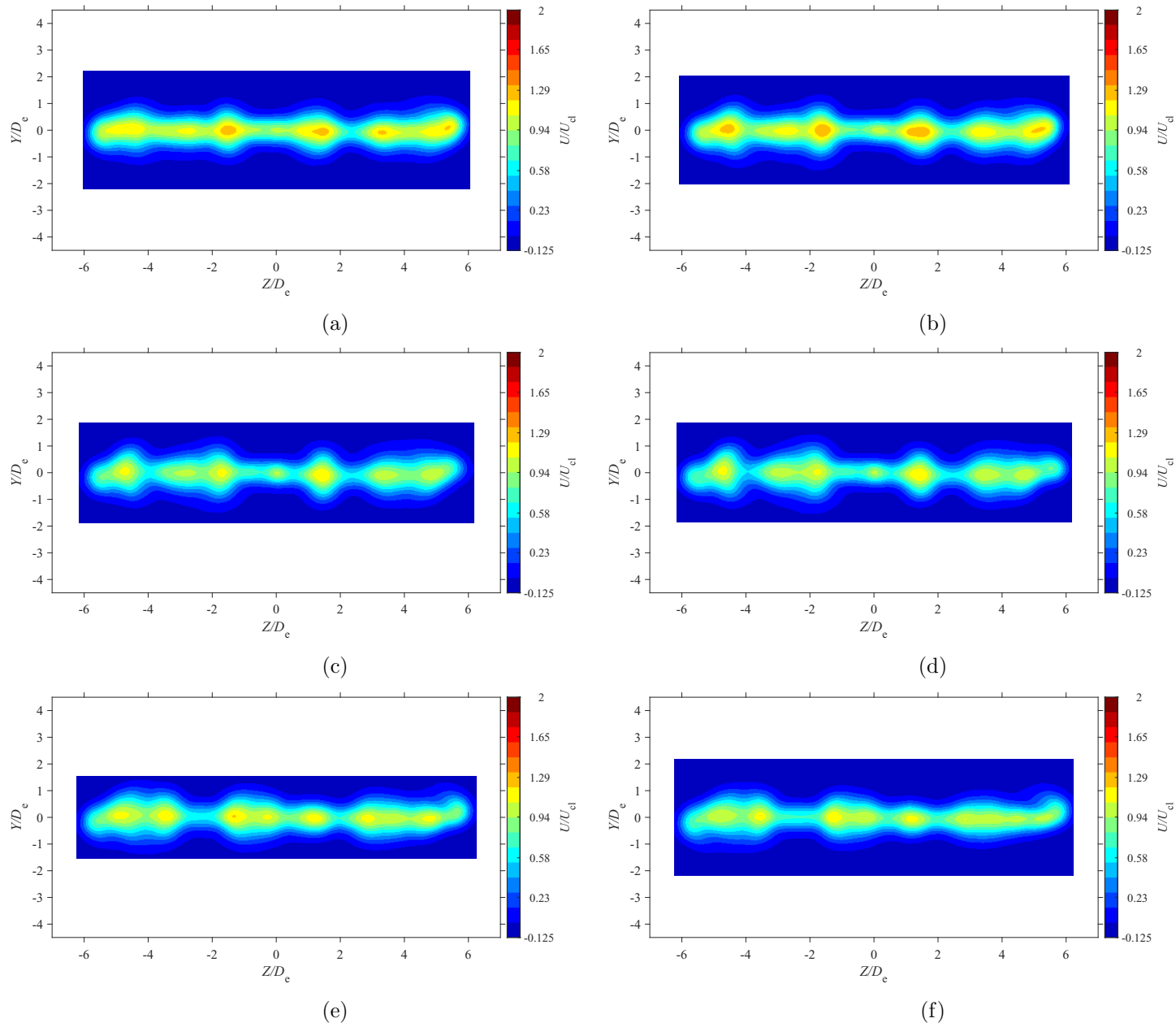


Figure E.10: Average streamwise velocity map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

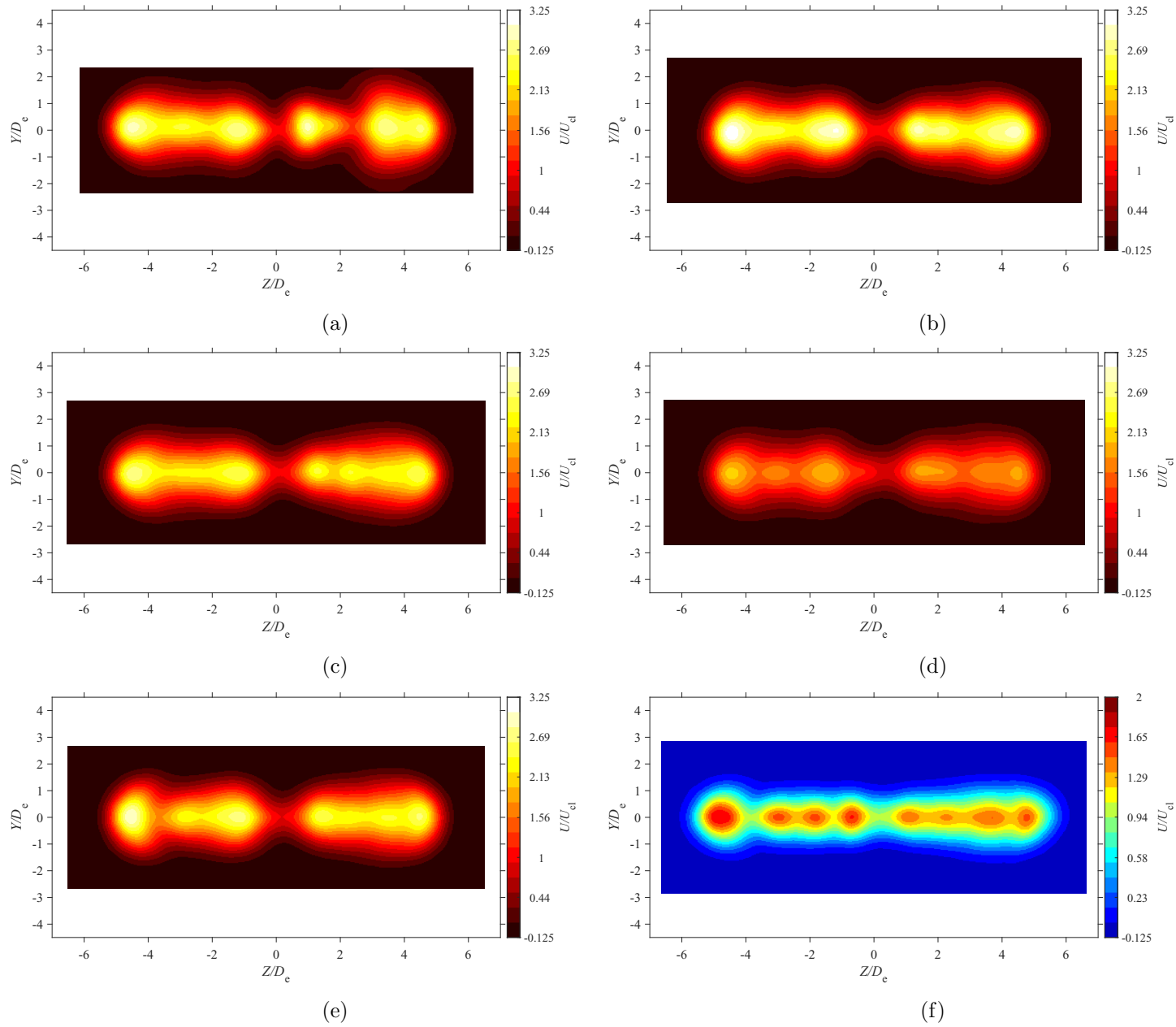


Figure E.11: Average streamwise velocity map at location $X = 5De$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

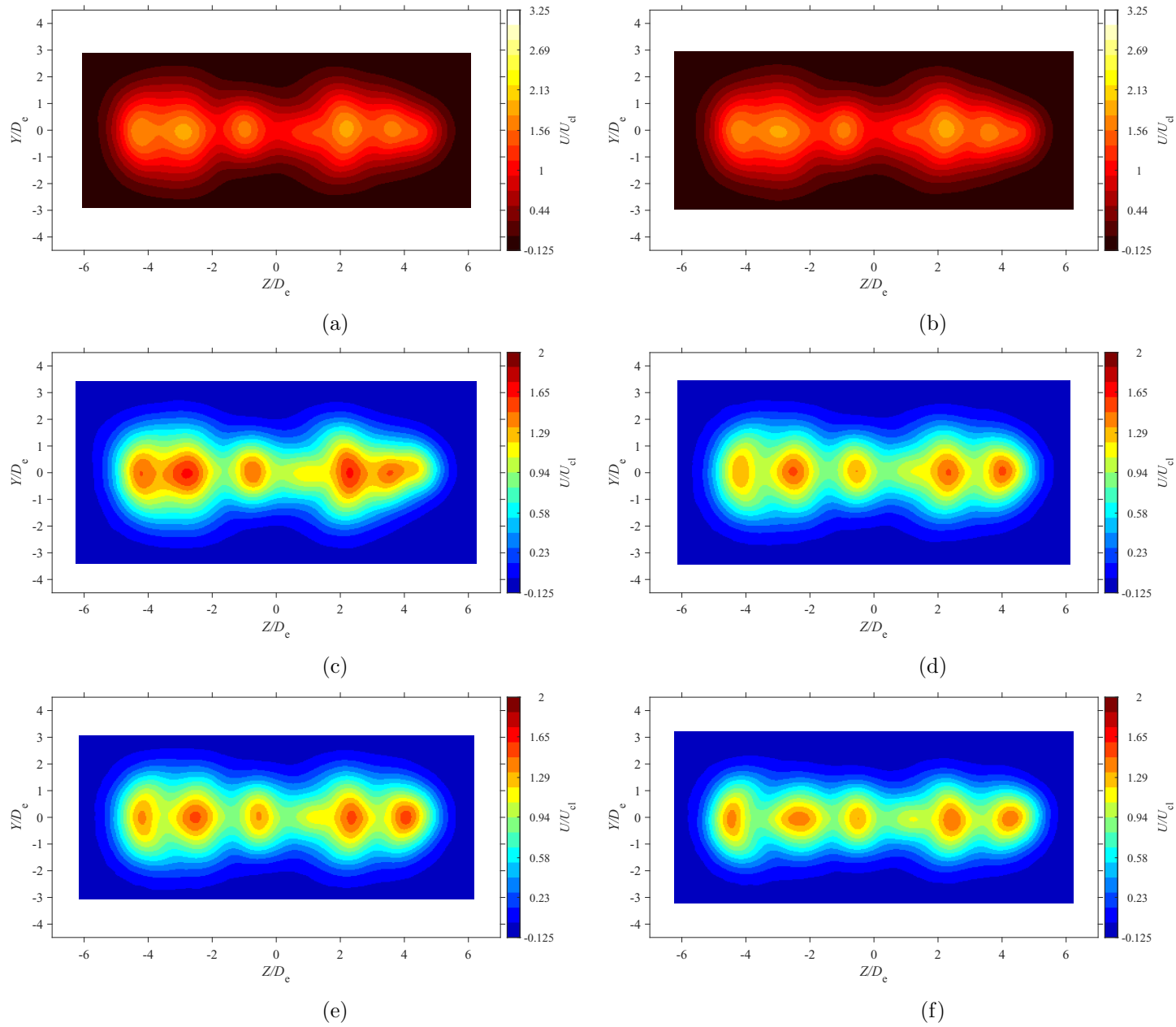


Figure E.12: Average streamwise velocity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

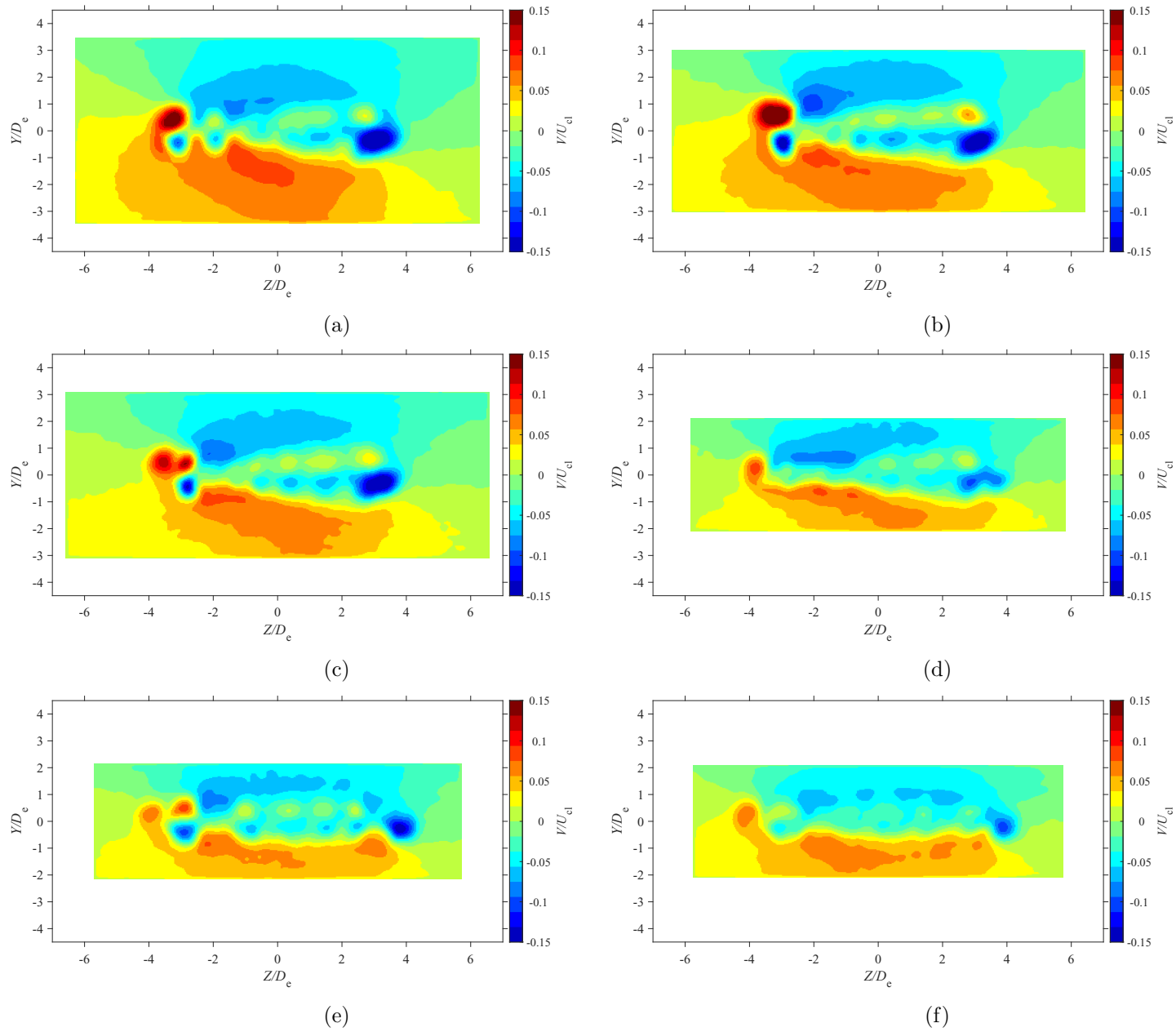


Figure E.13: Average lateral velocity map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

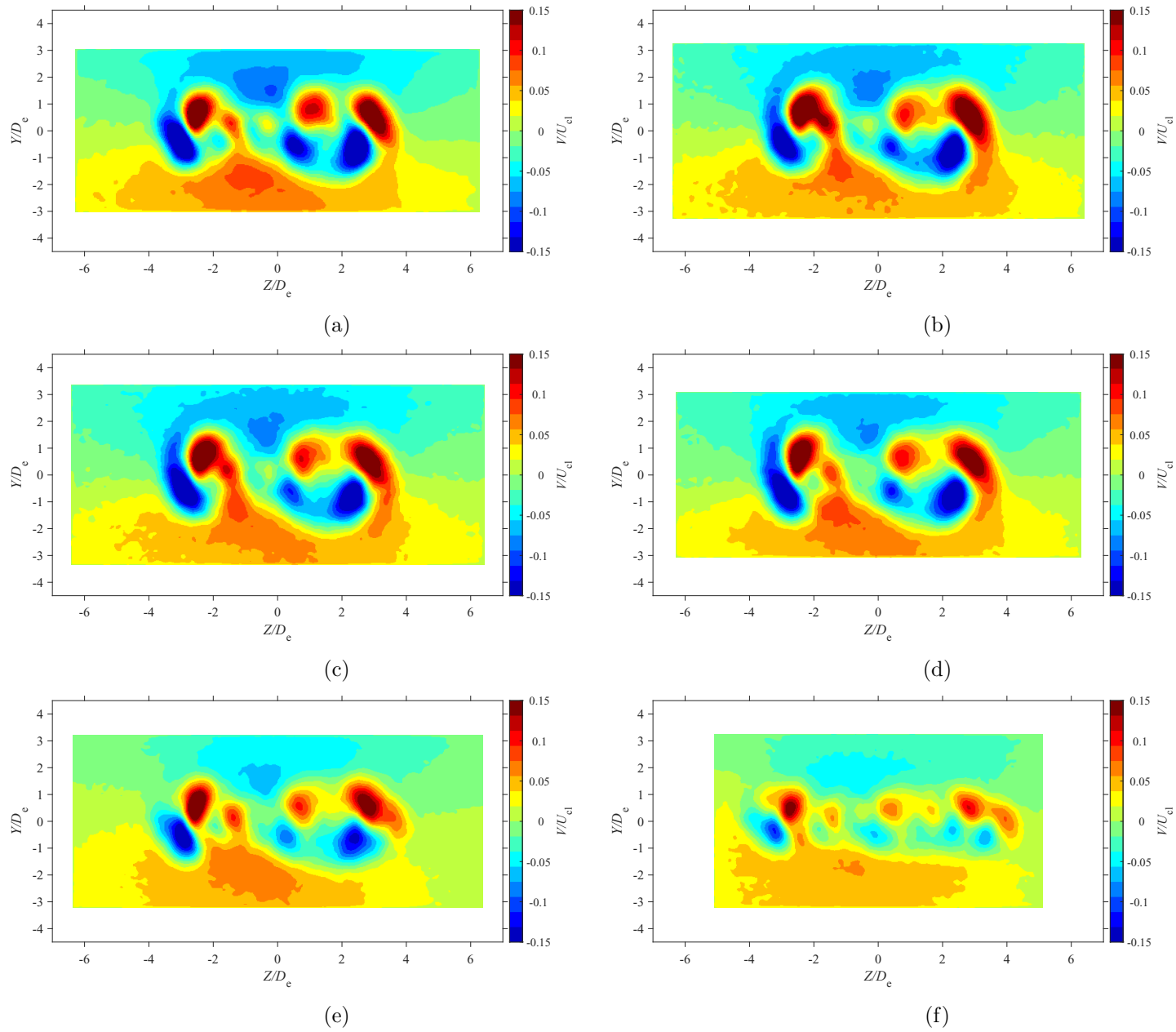


Figure E.14: Average lateral velocity map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

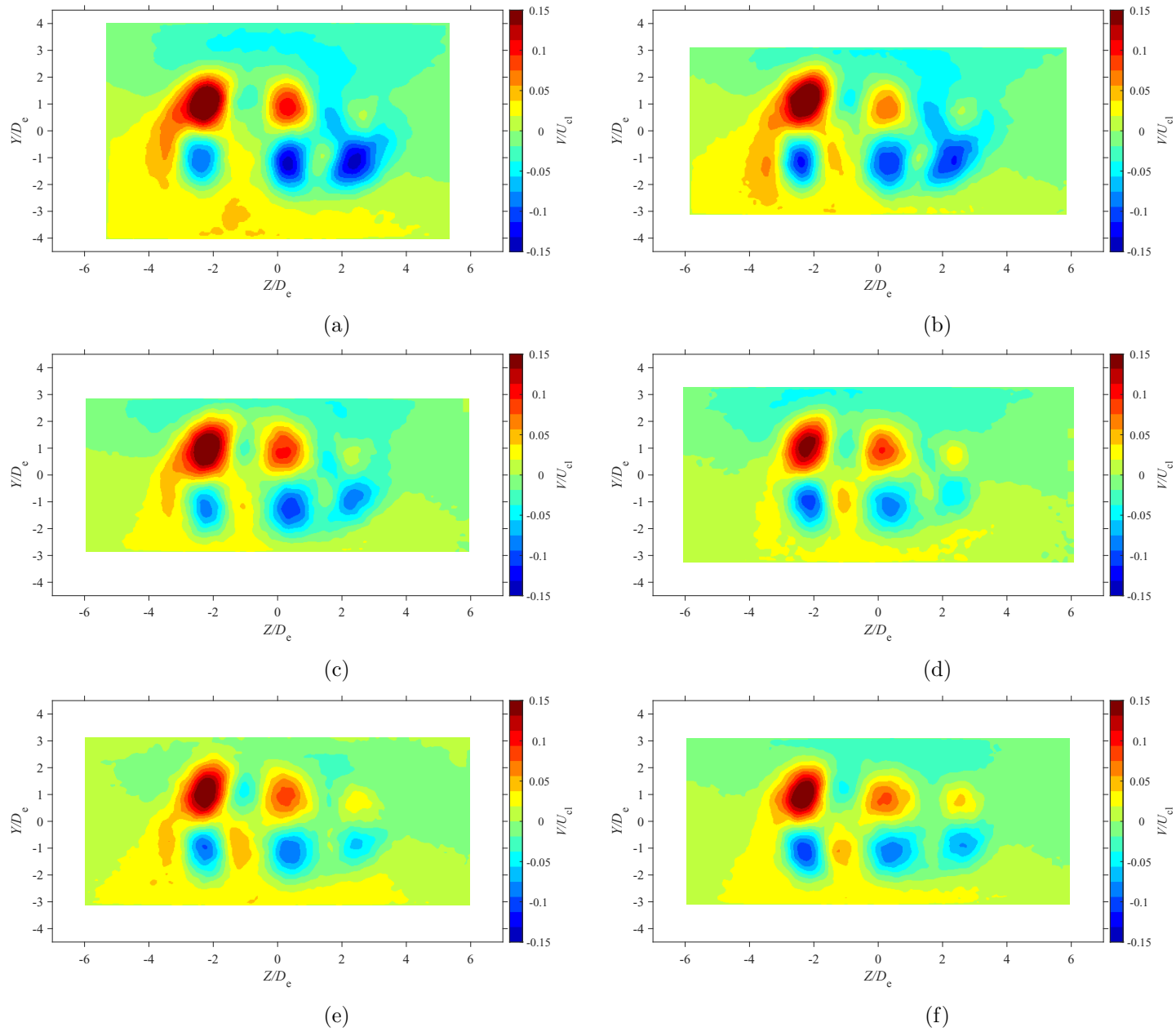


Figure E.15: Average lateral velocity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

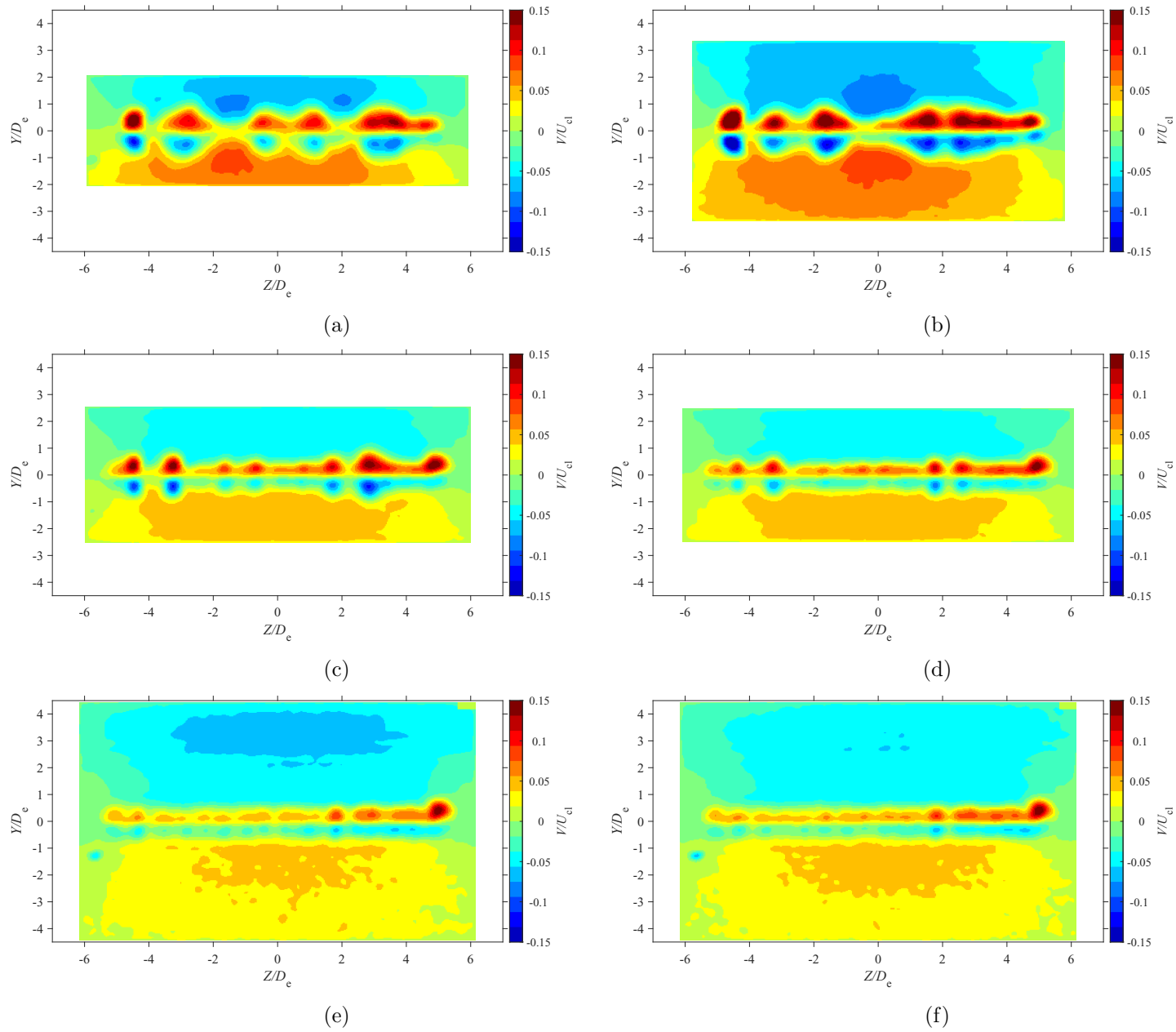


Figure E.16: Average lateral velocity map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

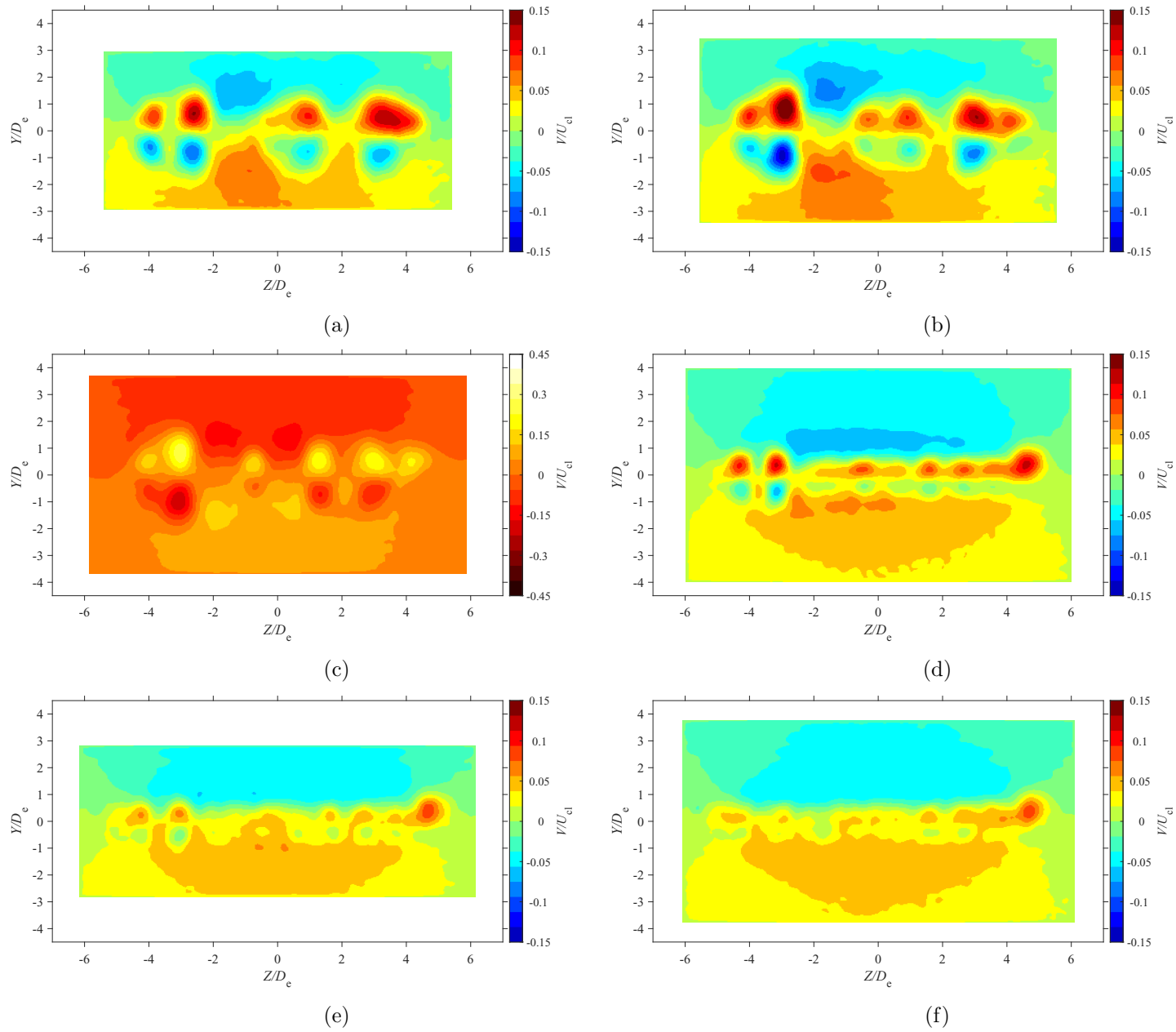


Figure E.17: Average lateral velocity map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

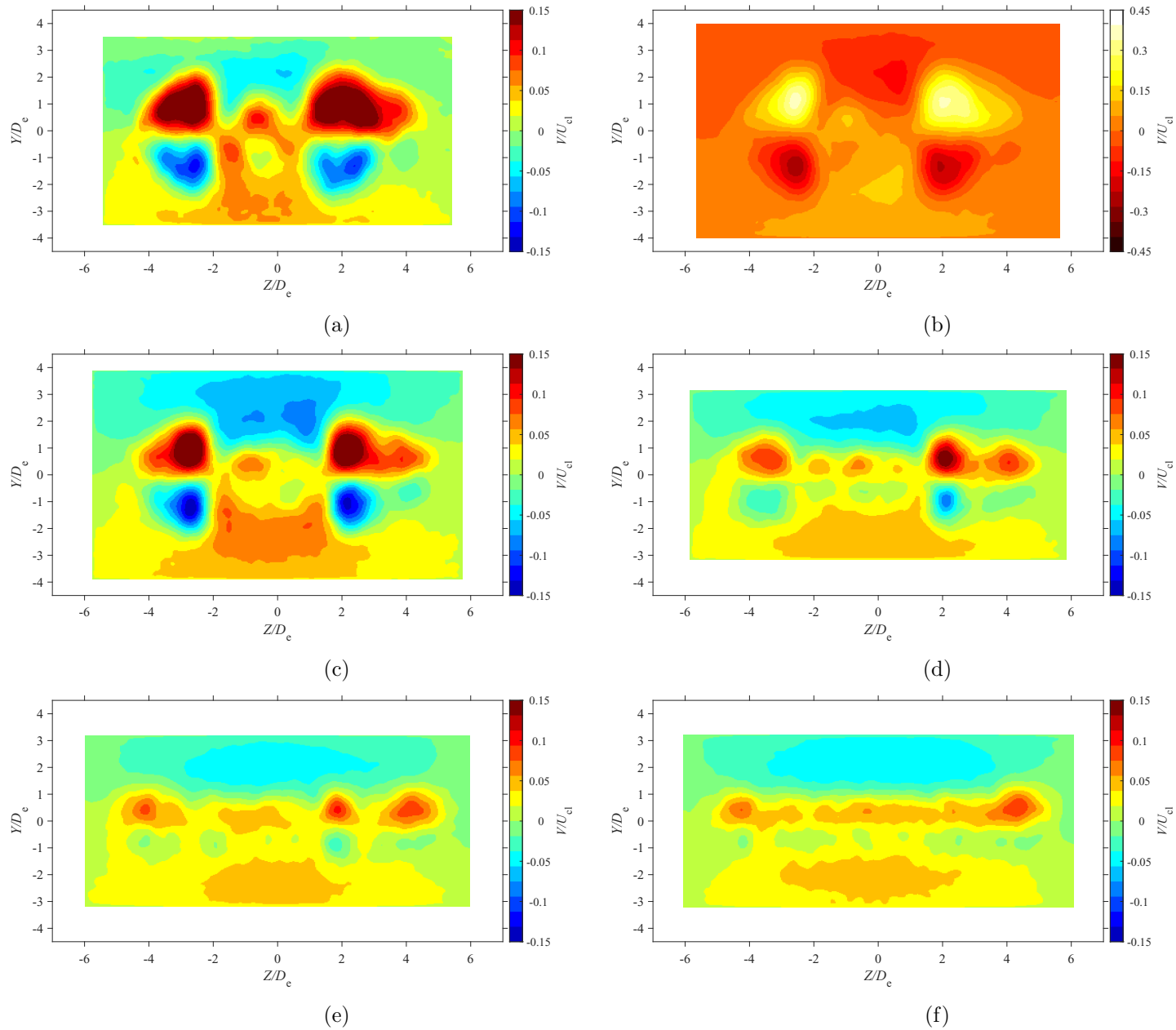


Figure E.18: Average lateral velocity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

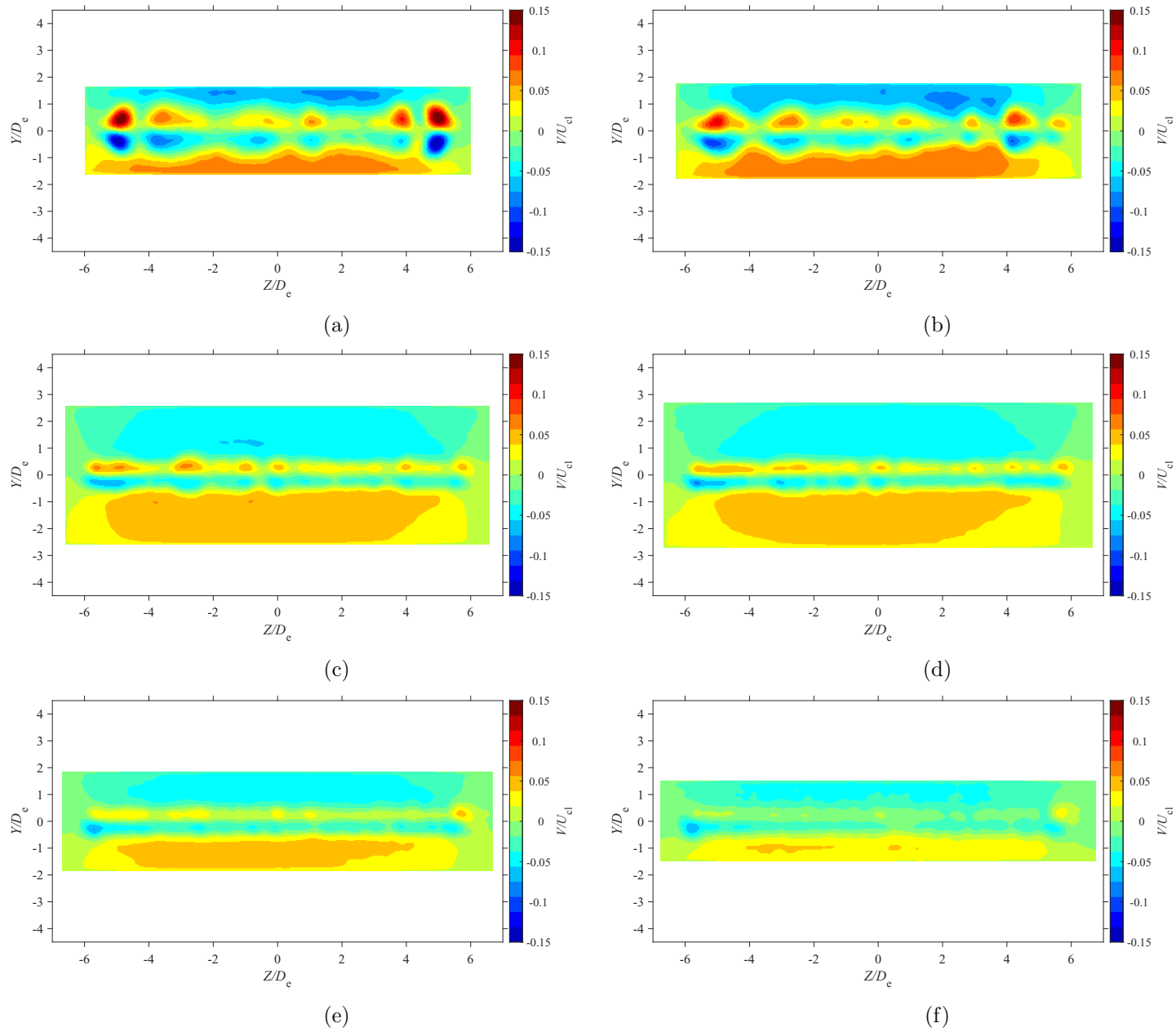


Figure E.19: Average lateral velocity map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

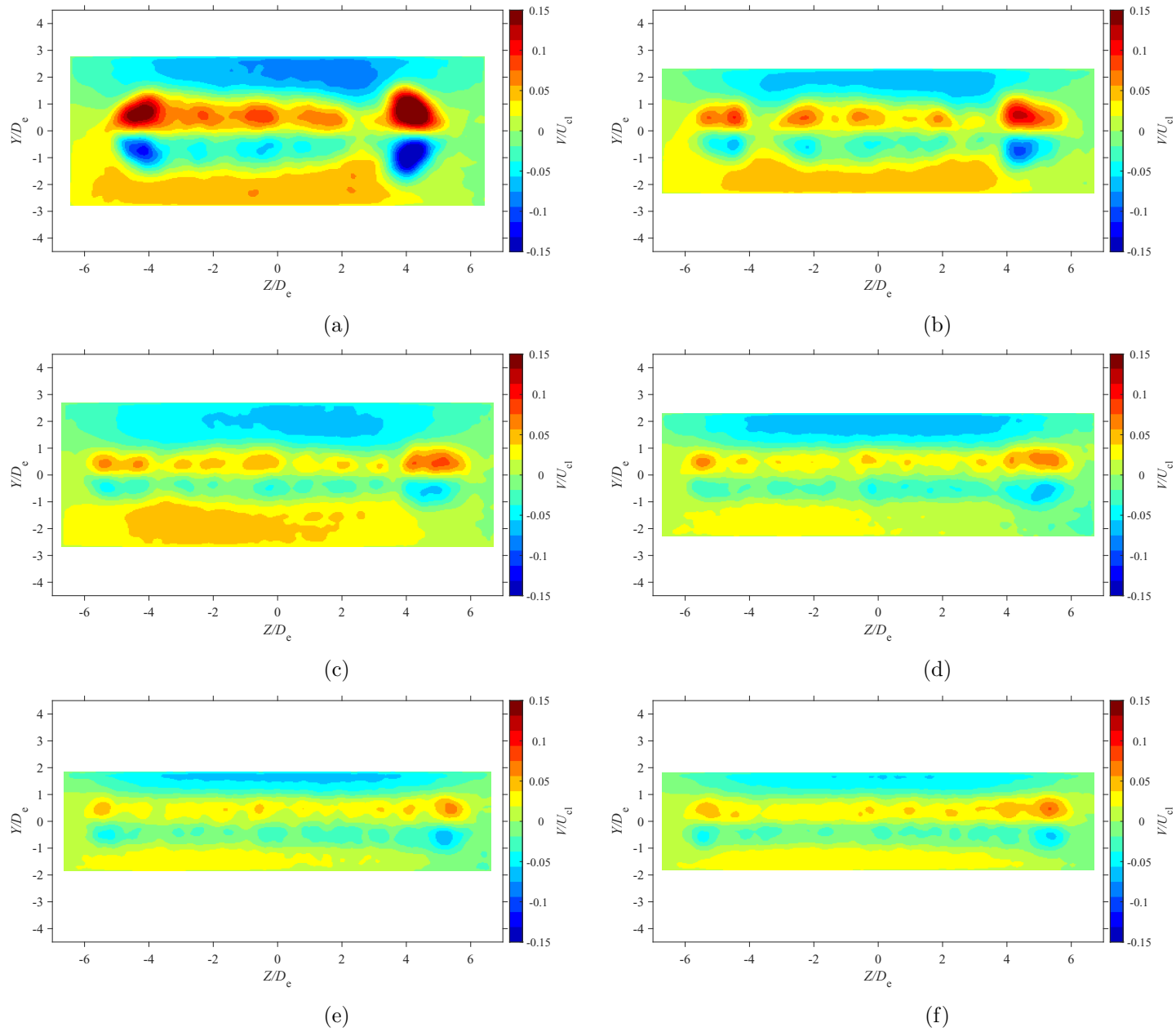


Figure E.20: Average lateral velocity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

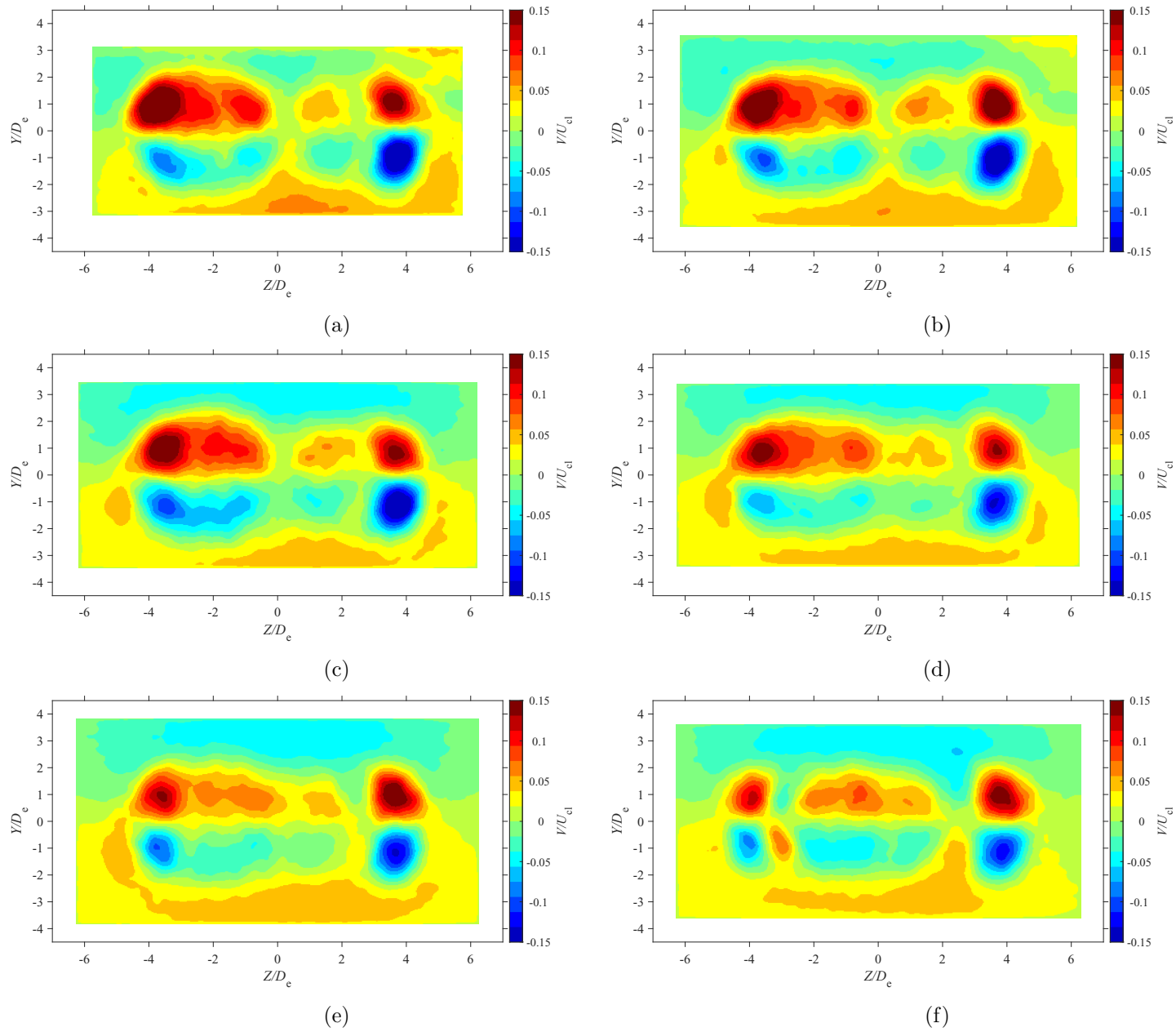


Figure E.21: Average lateral velocity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

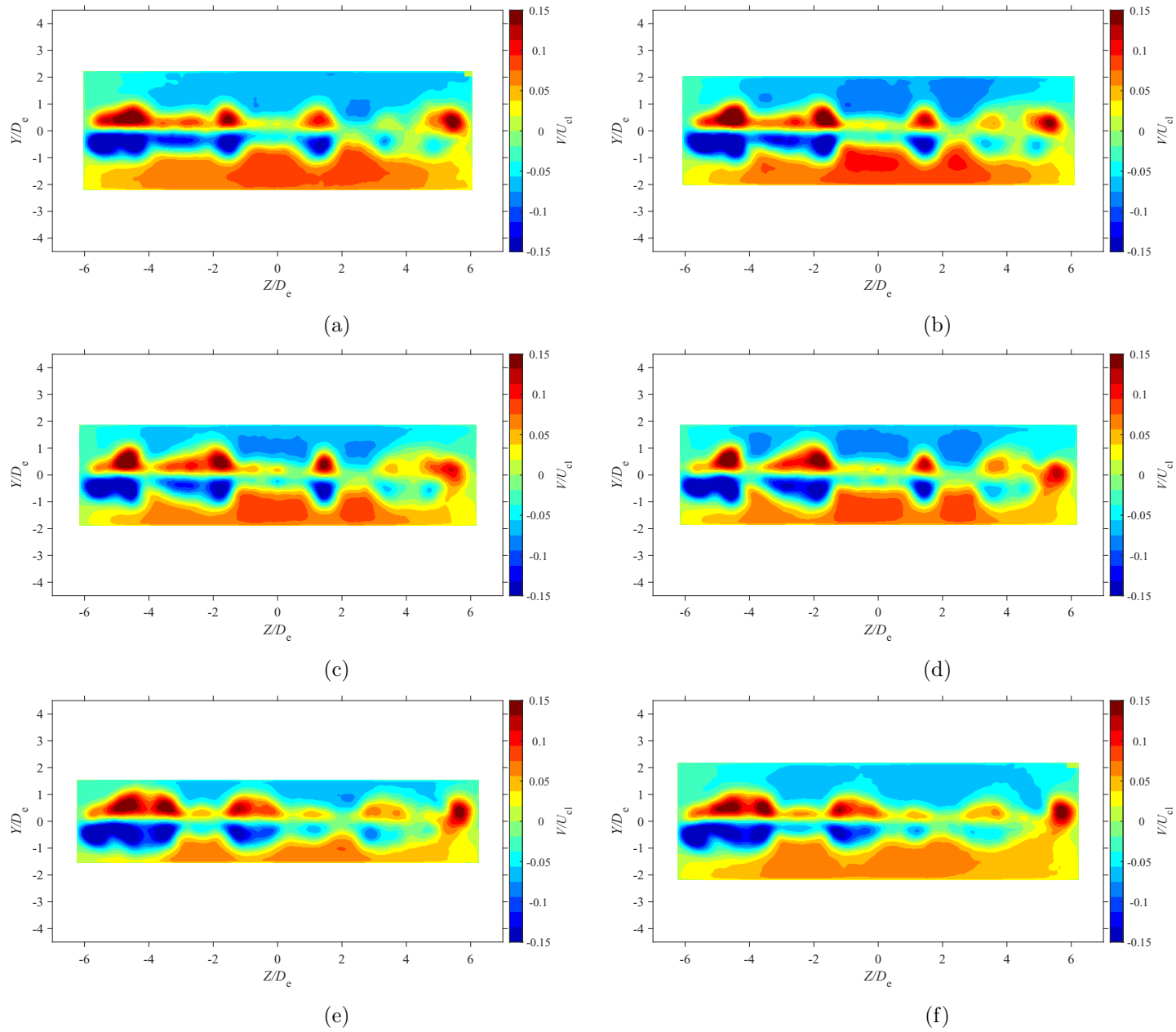


Figure E.22: Average lateral velocity map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

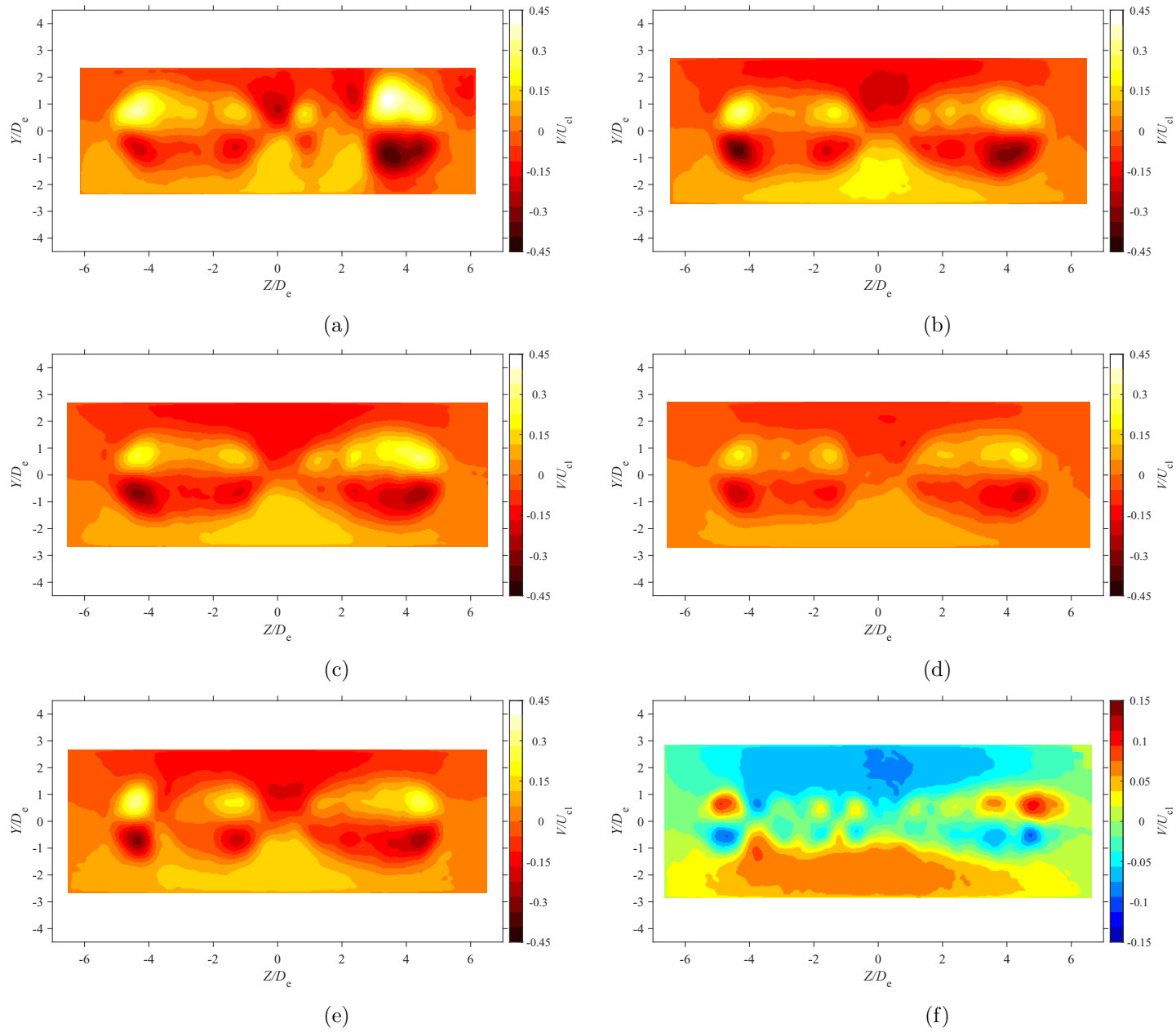


Figure E.23: Average lateral velocity map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

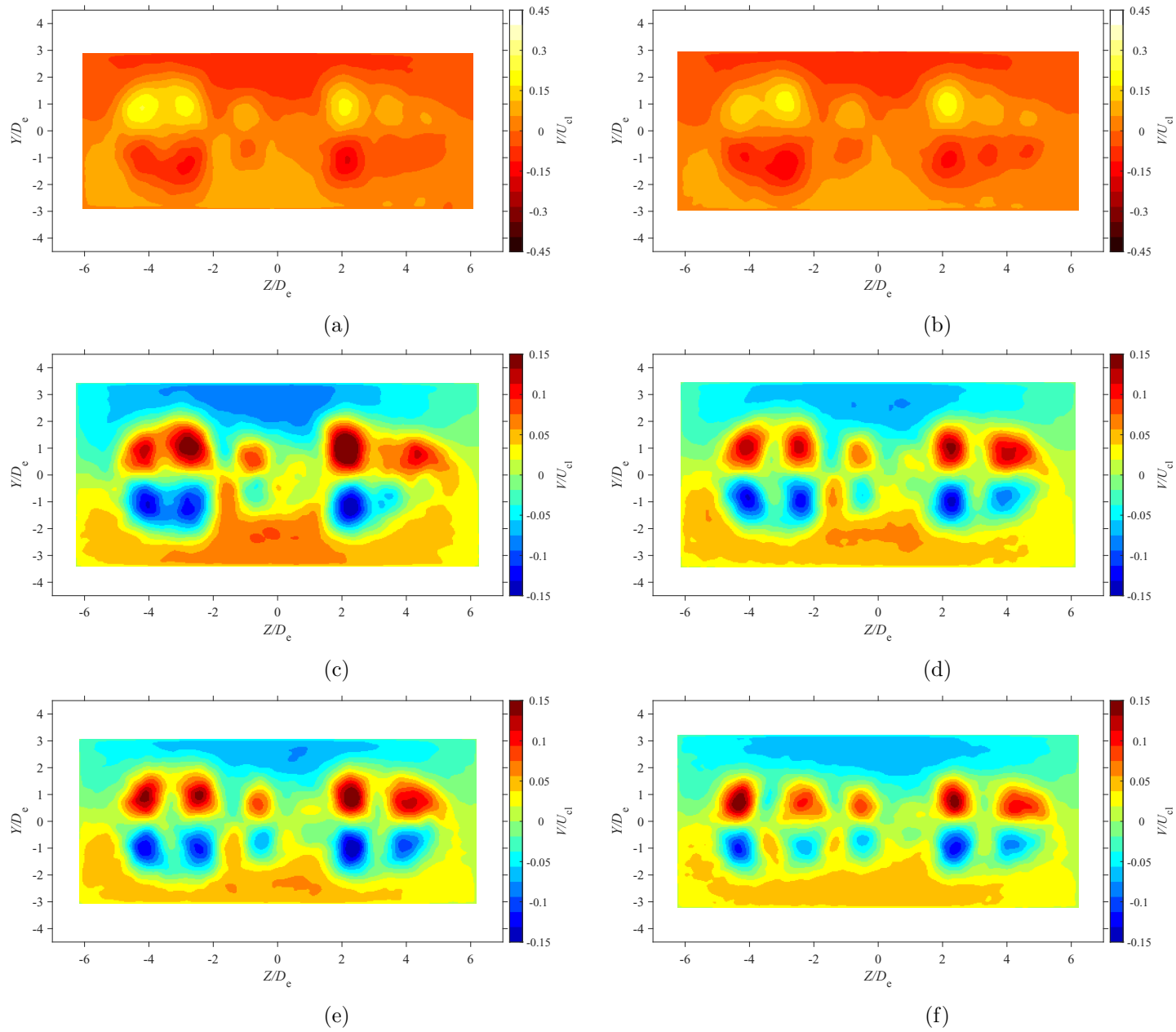


Figure E.24: Average lateral velocity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

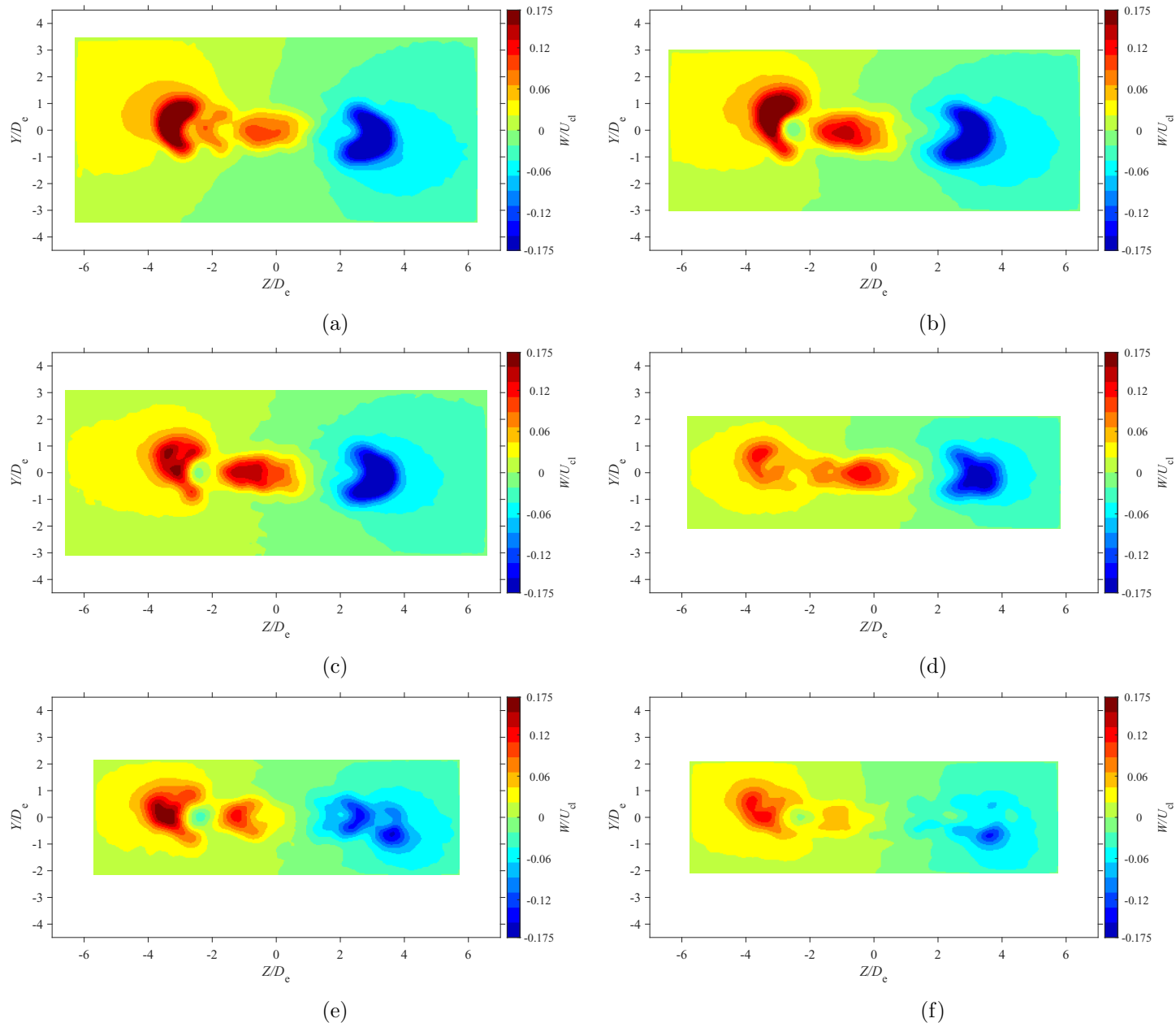


Figure E.25: Average spanwise velocity map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

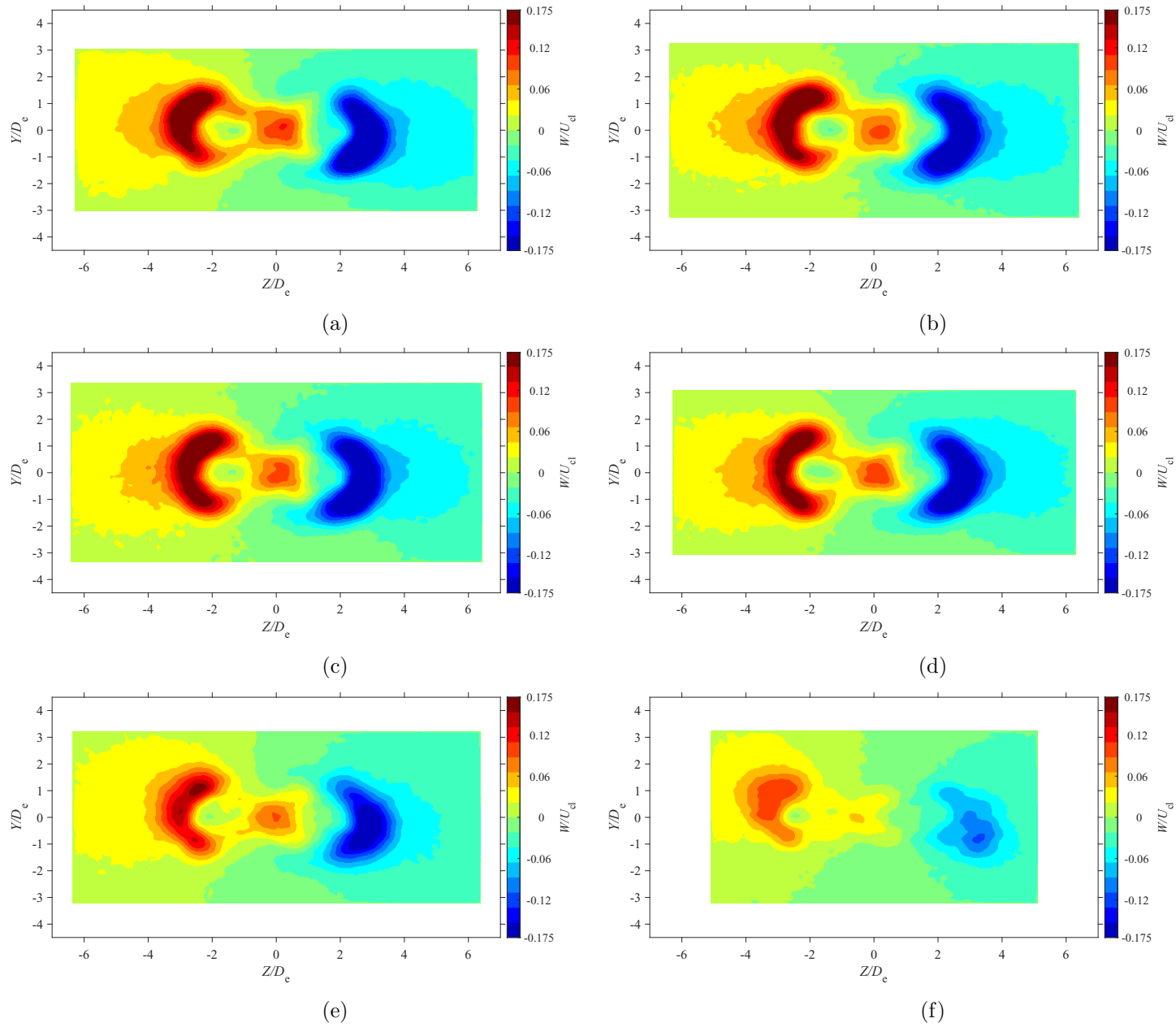


Figure E.26: Average spanwise velocity map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

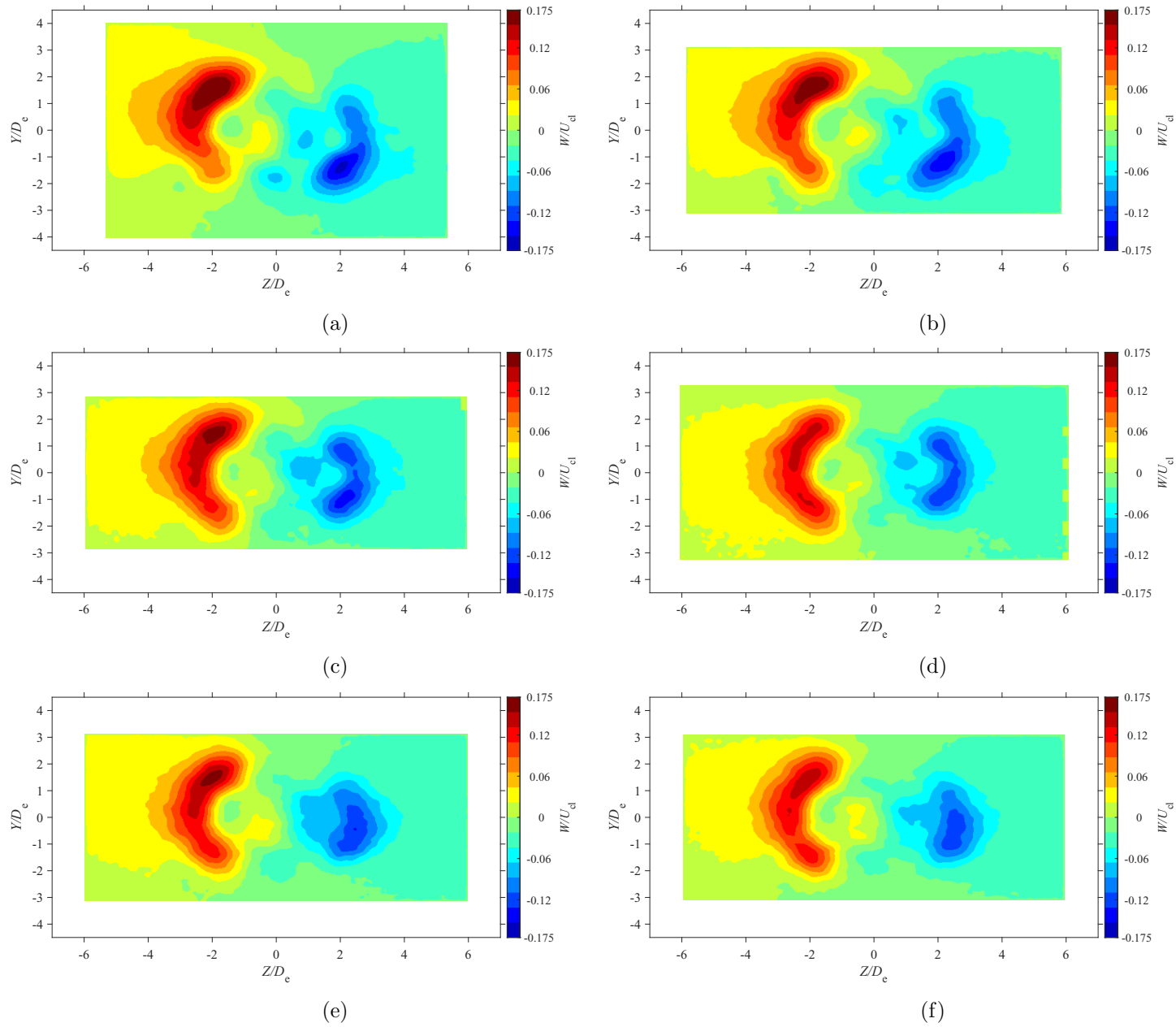


Figure E.27: Average spanwise velocity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

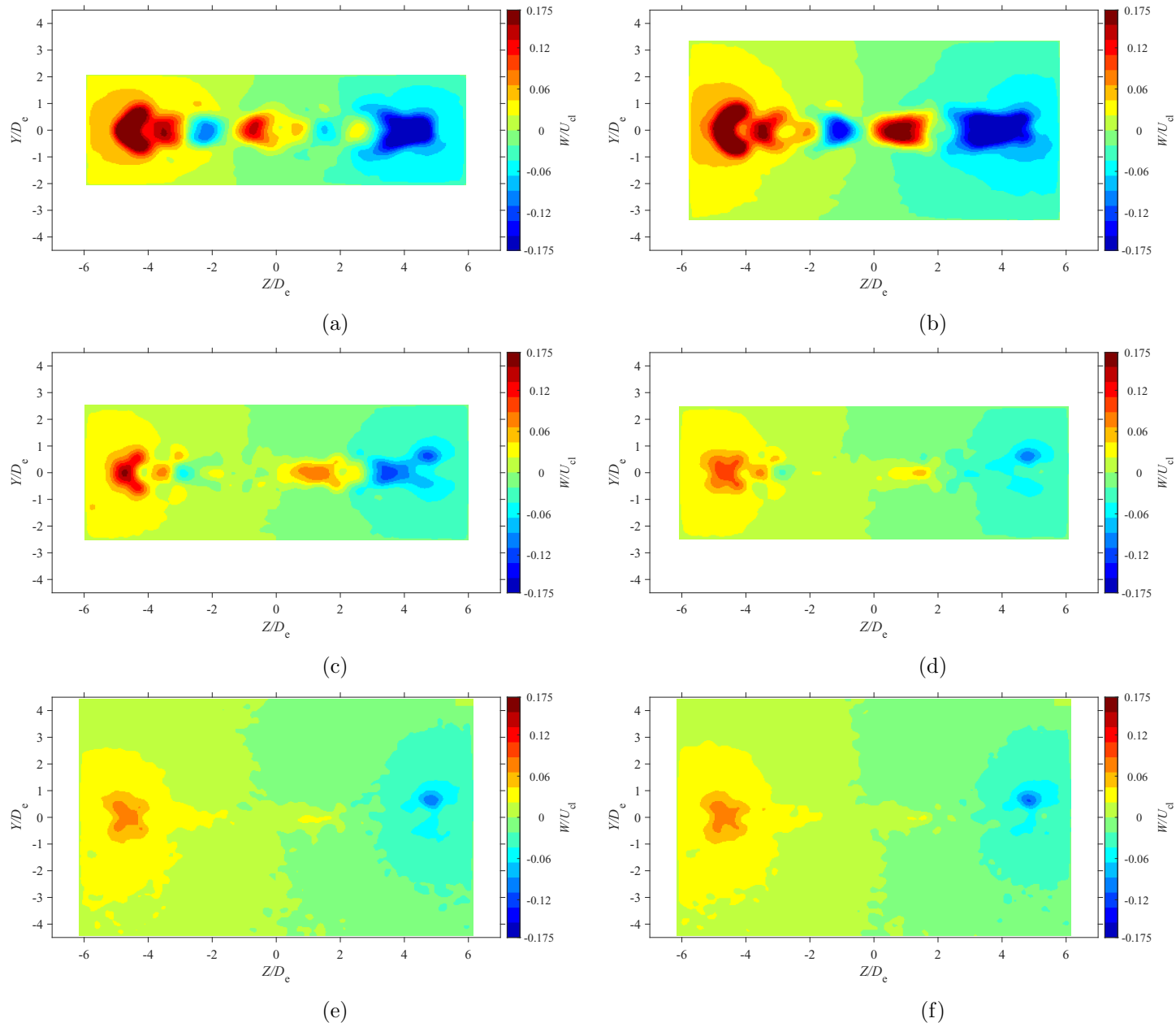


Figure E.28: Average spanwise velocity map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

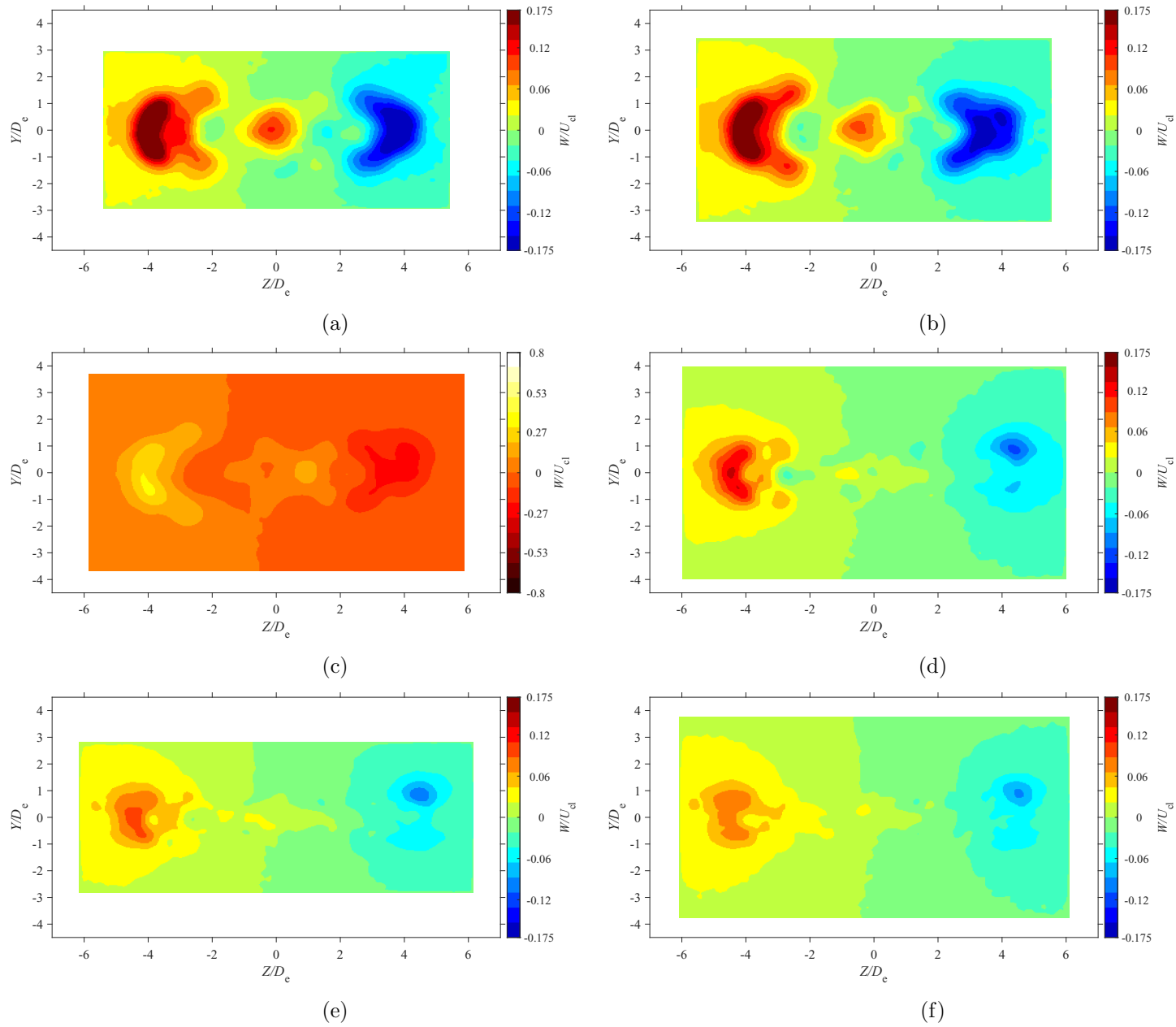
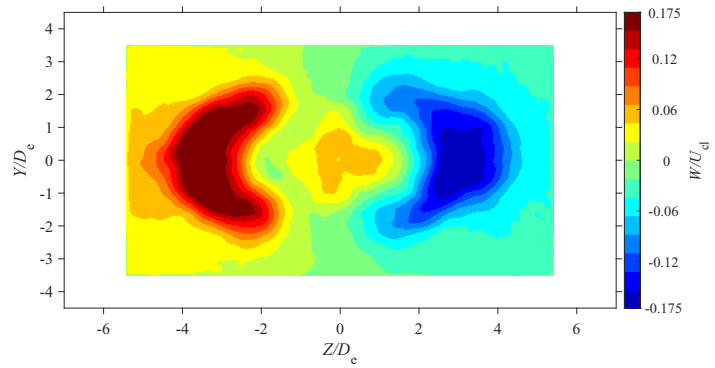
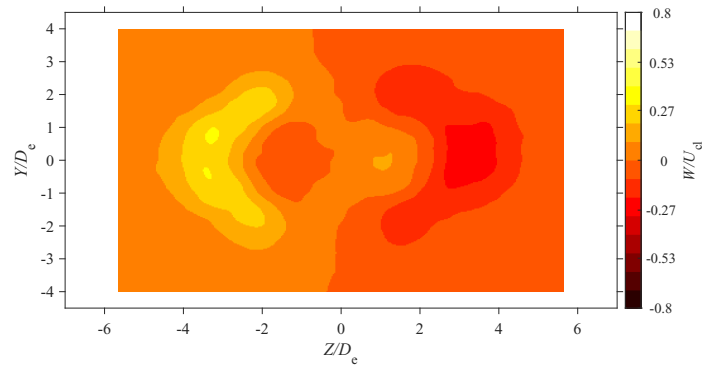


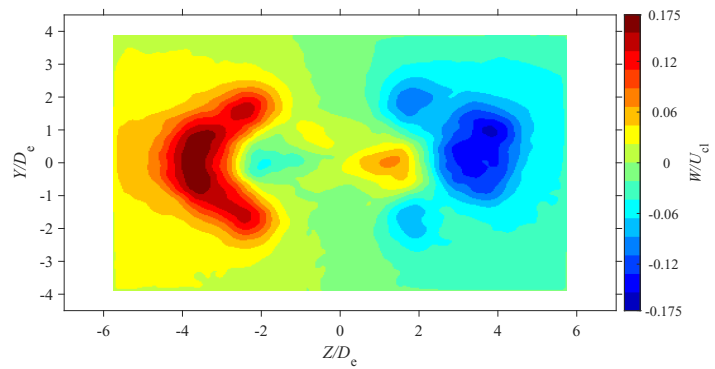
Figure E.29: Average spanwise velocity map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



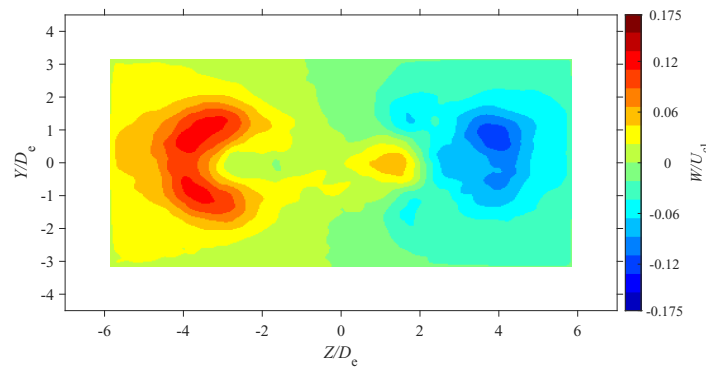
(a)



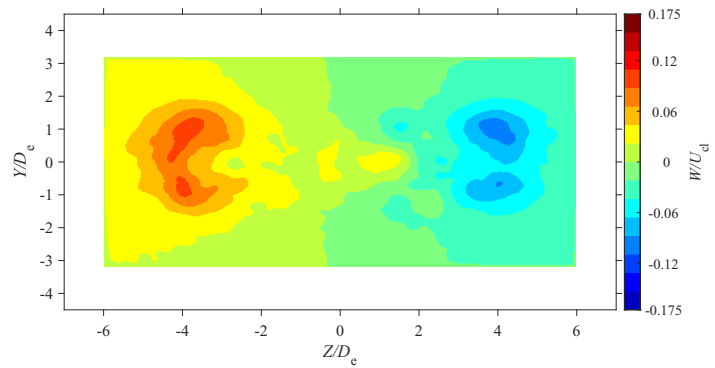
(b)



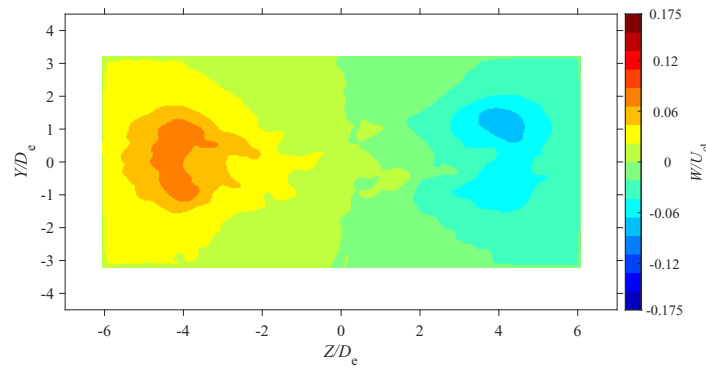
(c)



(d)



(e)



(f)

Figure E.30: Average spanwise velocity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

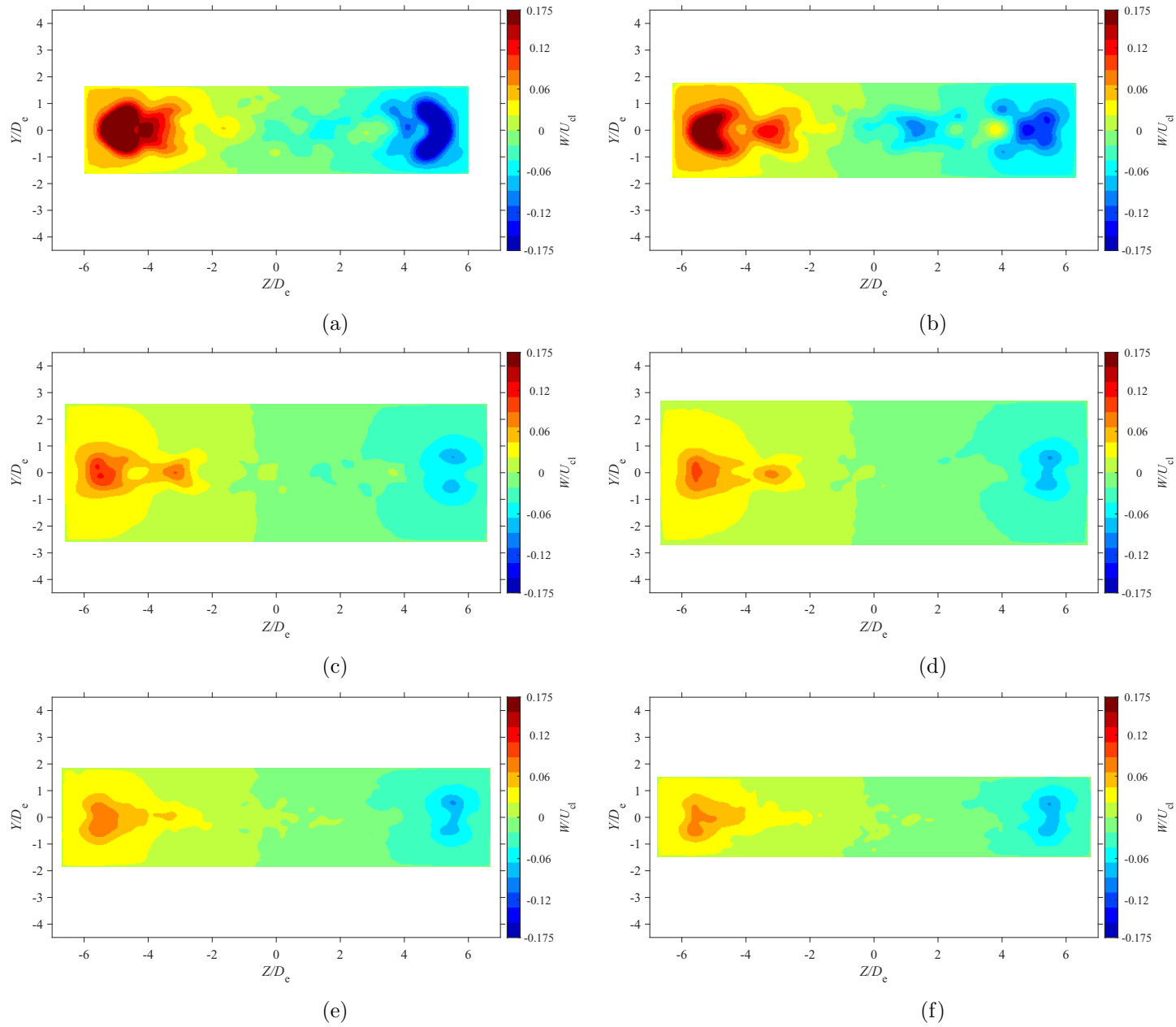


Figure E.31: Average spanwise velocity map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

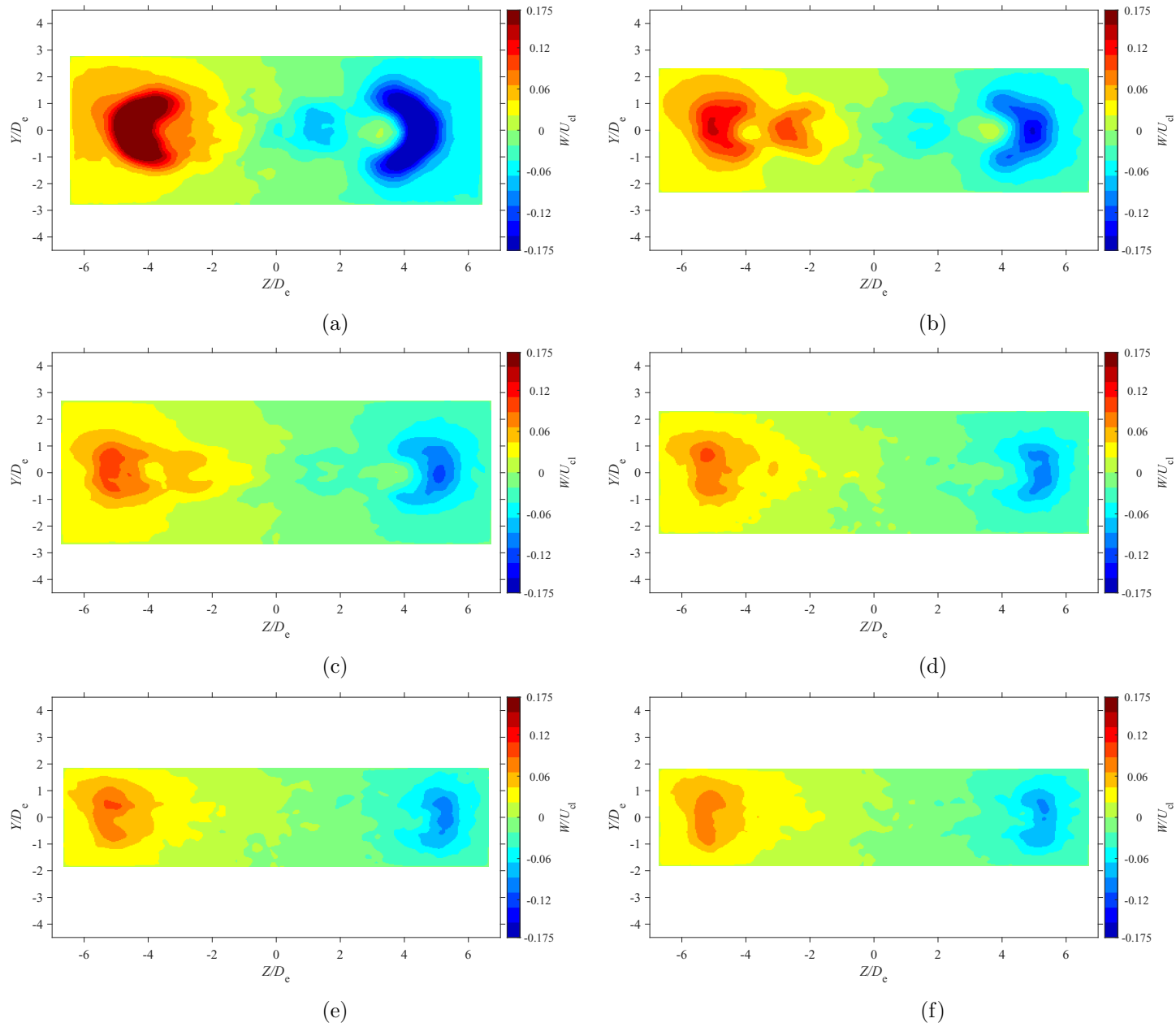


Figure E.32: Average spanwise velocity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

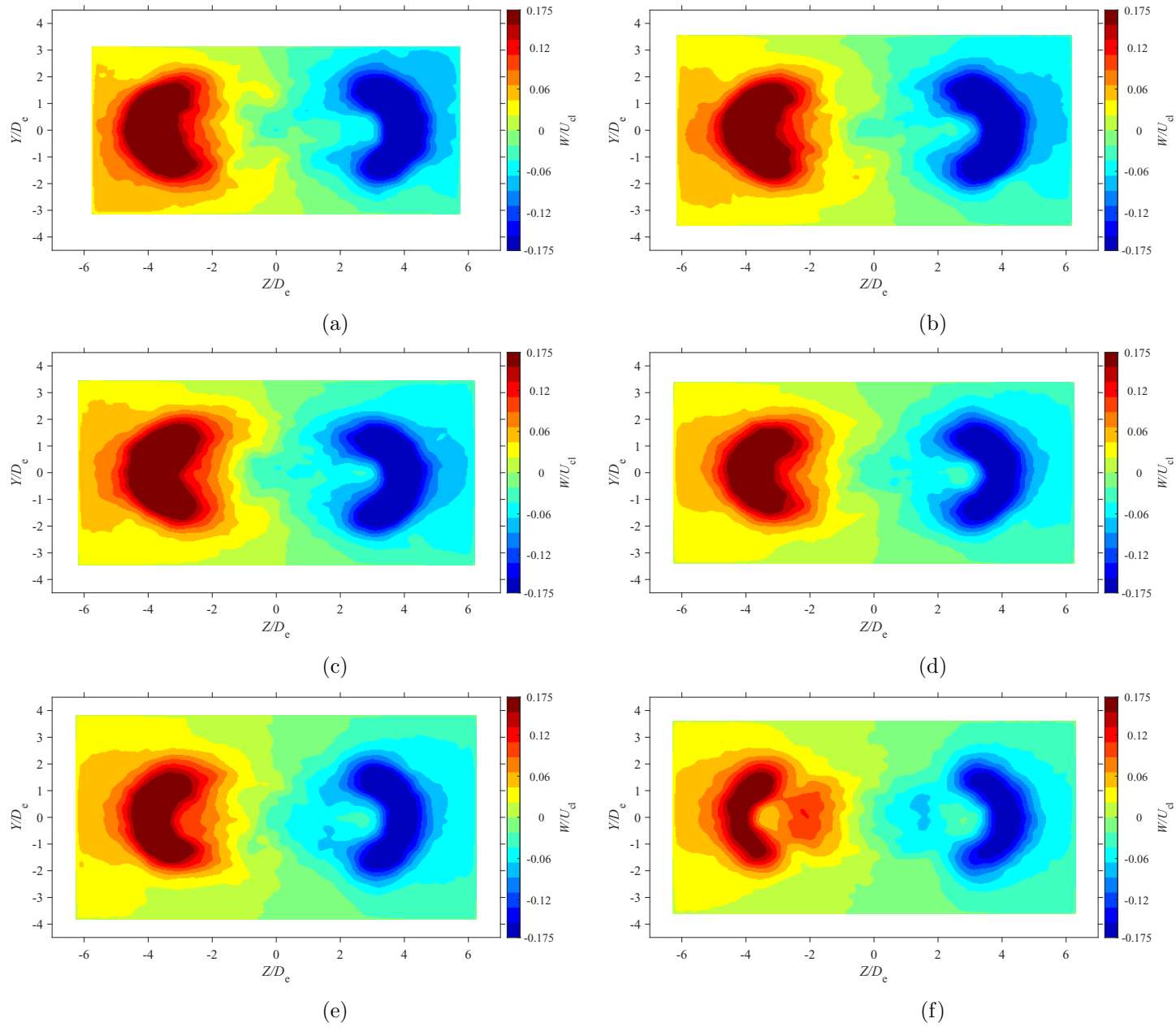


Figure E.33: Average spanwise velocity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

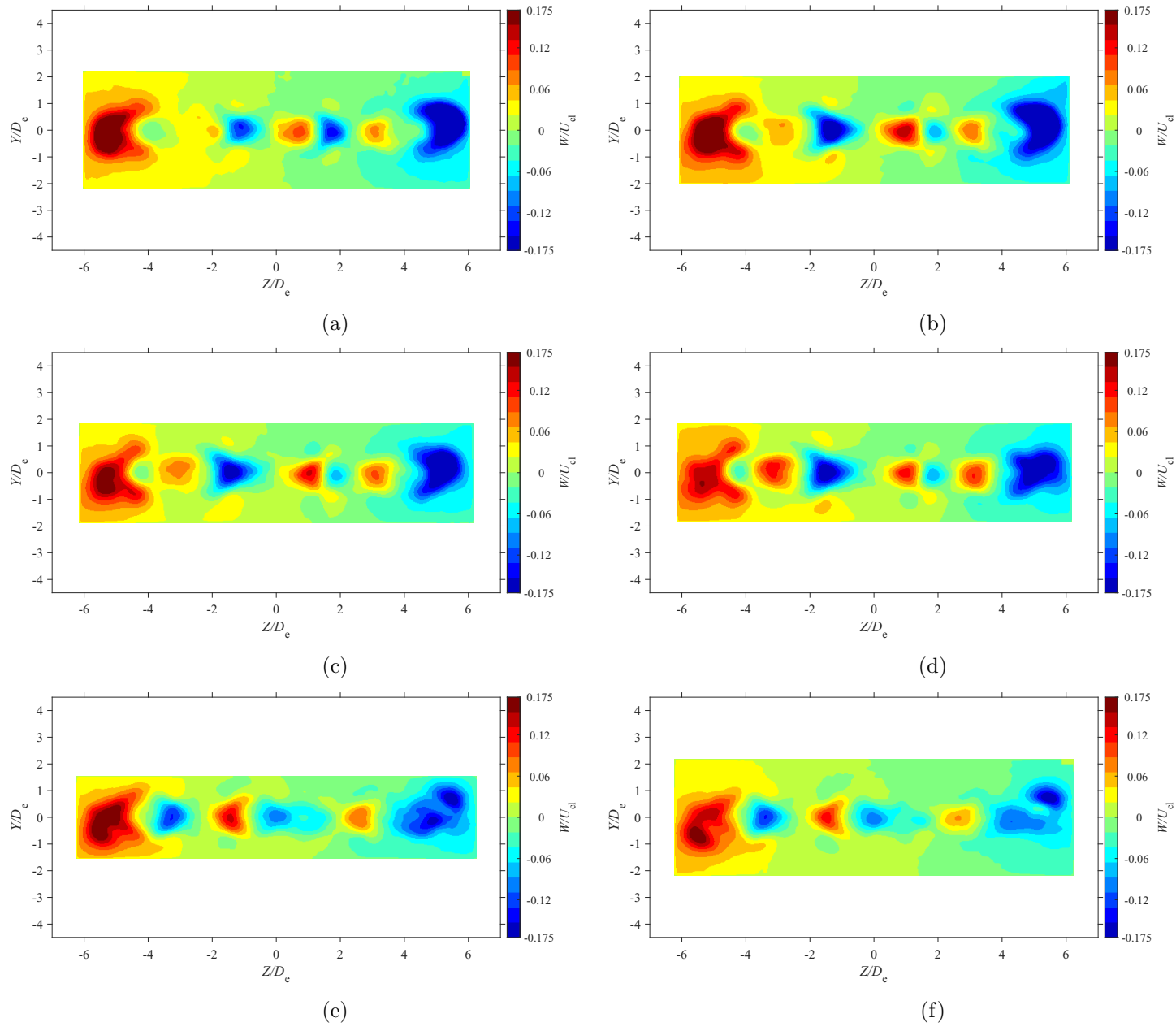


Figure E.34: Average spanwise velocity map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

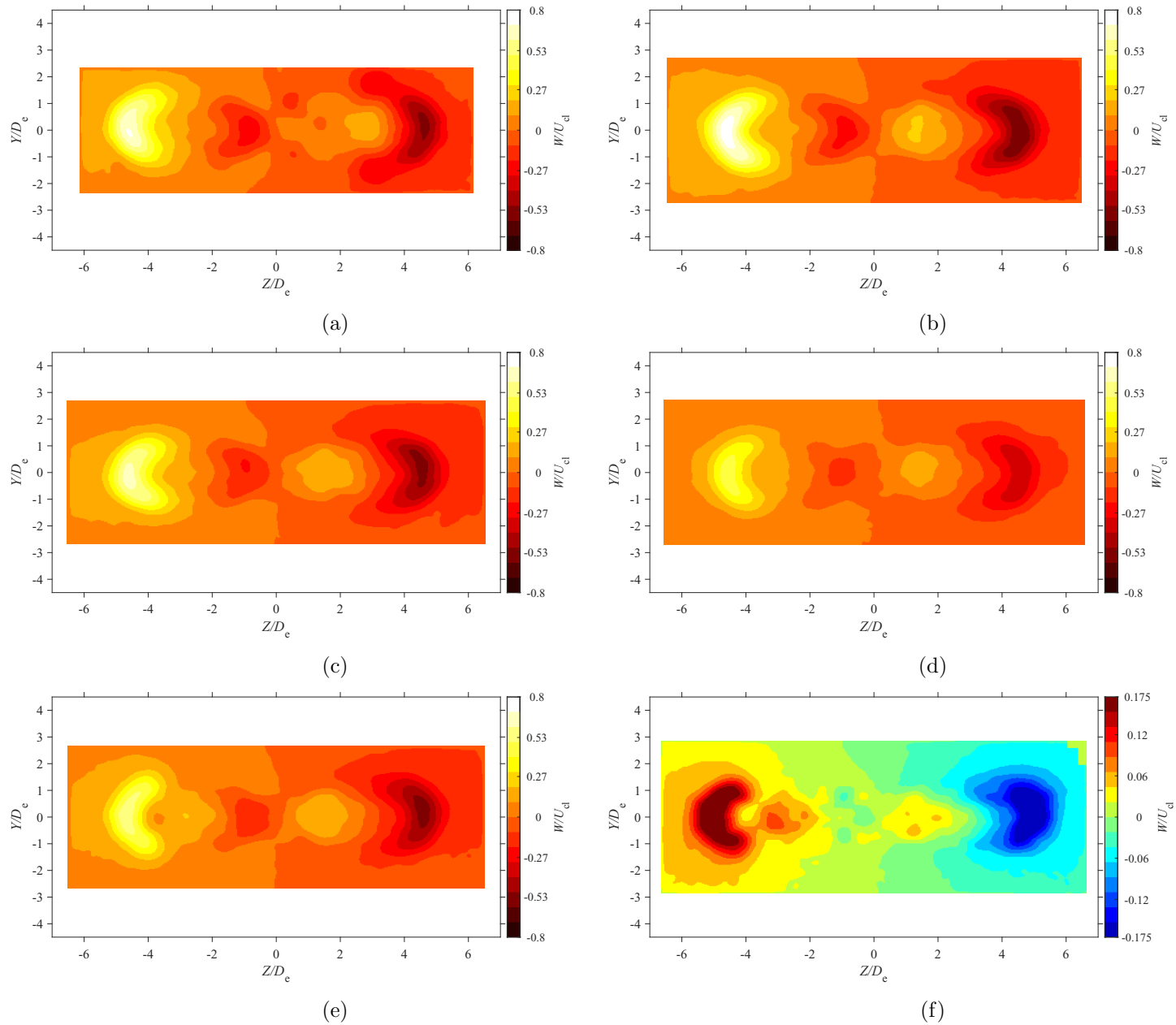


Figure E.35: Average spanwise velocity map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

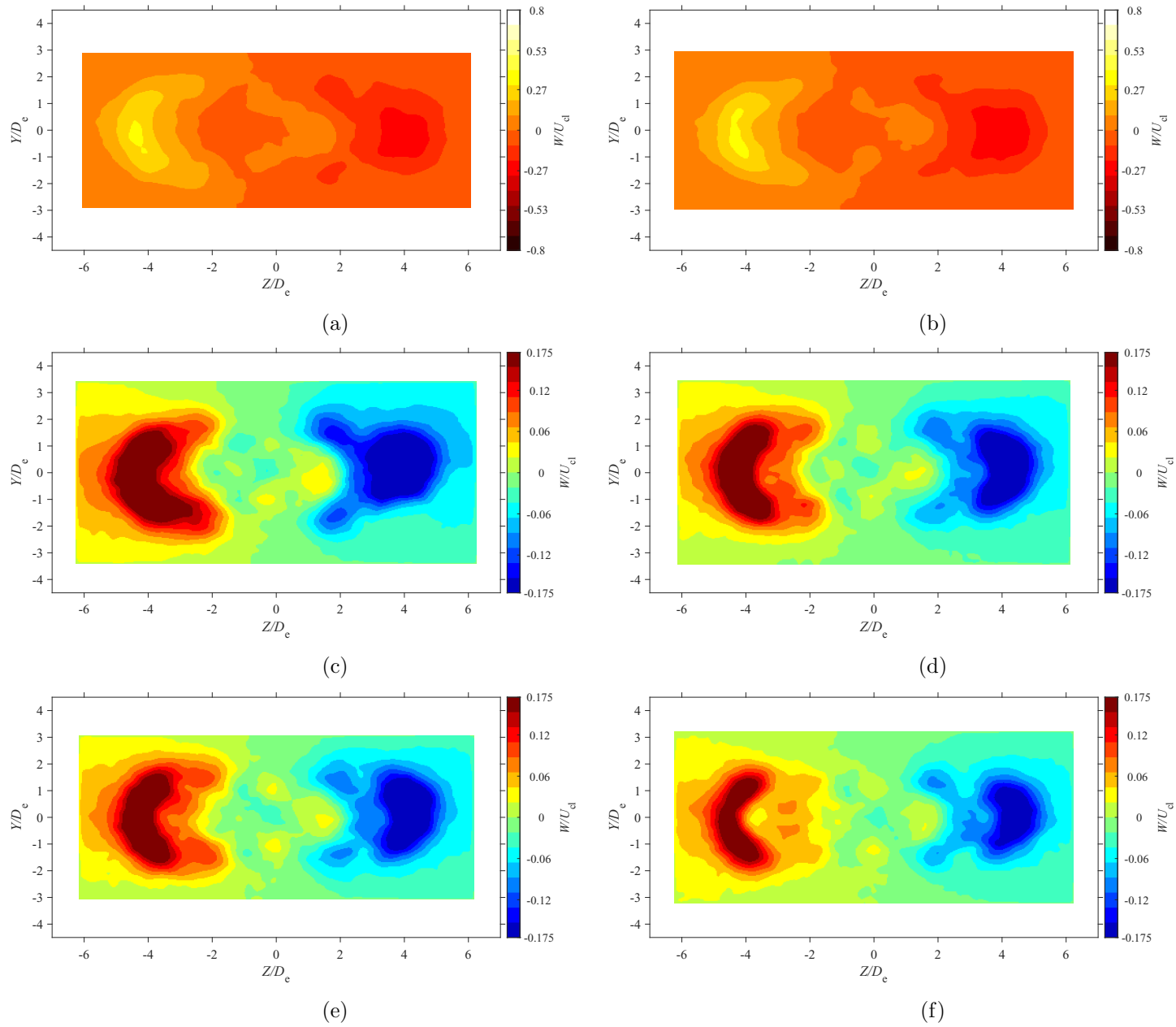


Figure E.36: Average spanwise velocity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

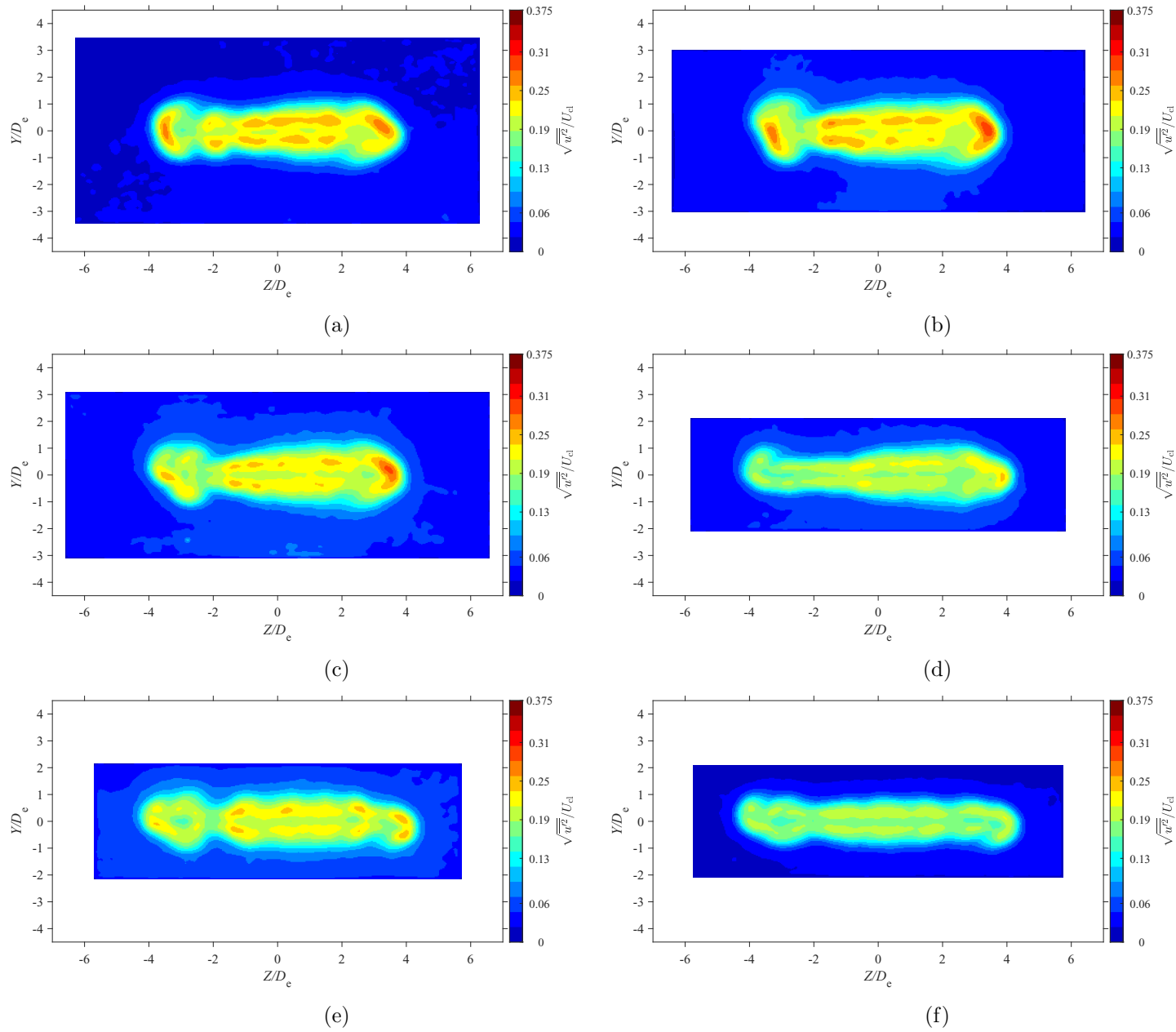


Figure E.37: Average streamwise turbulence intensity map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

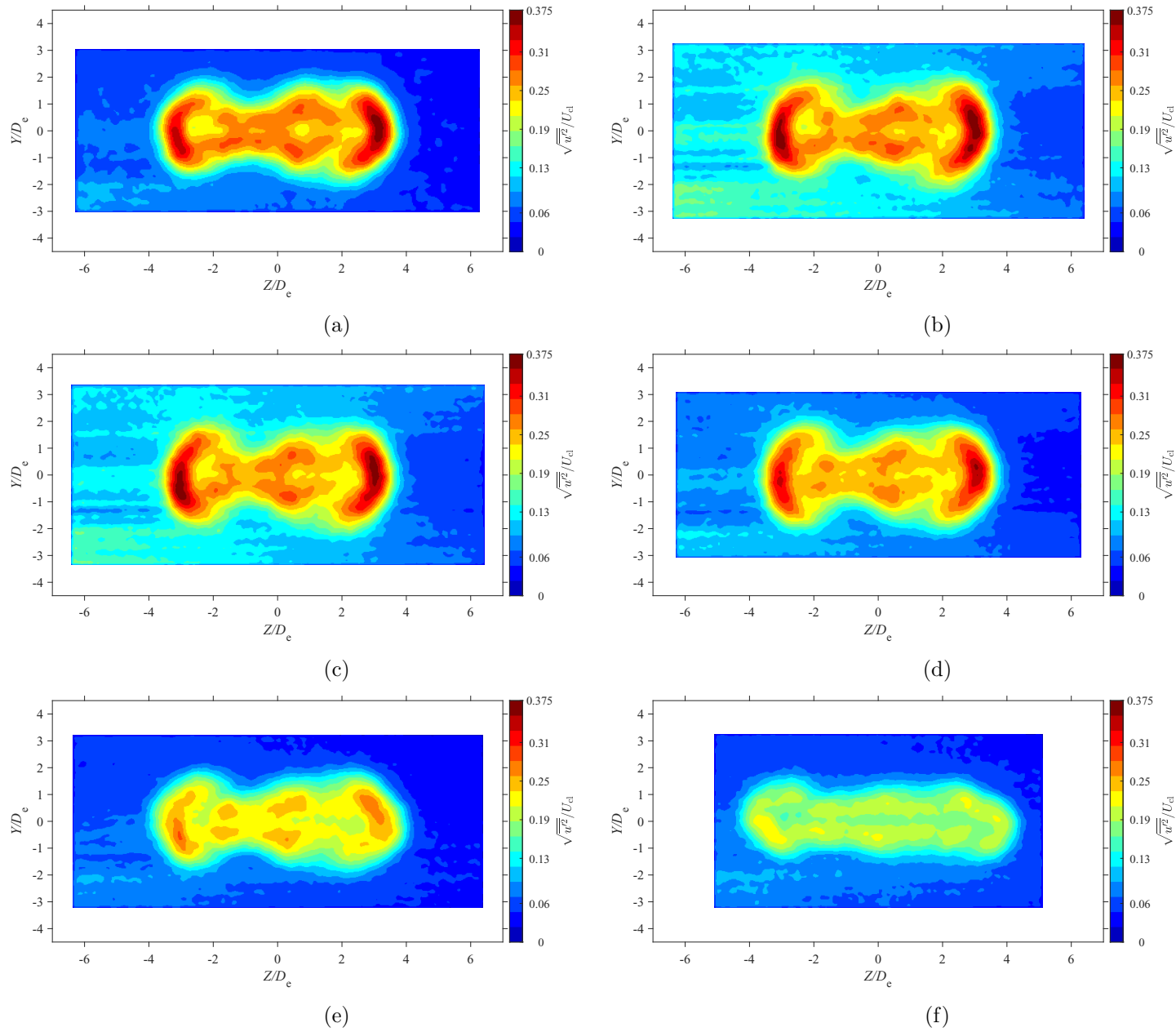


Figure E.38: Average streamwise turbulence intensity map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

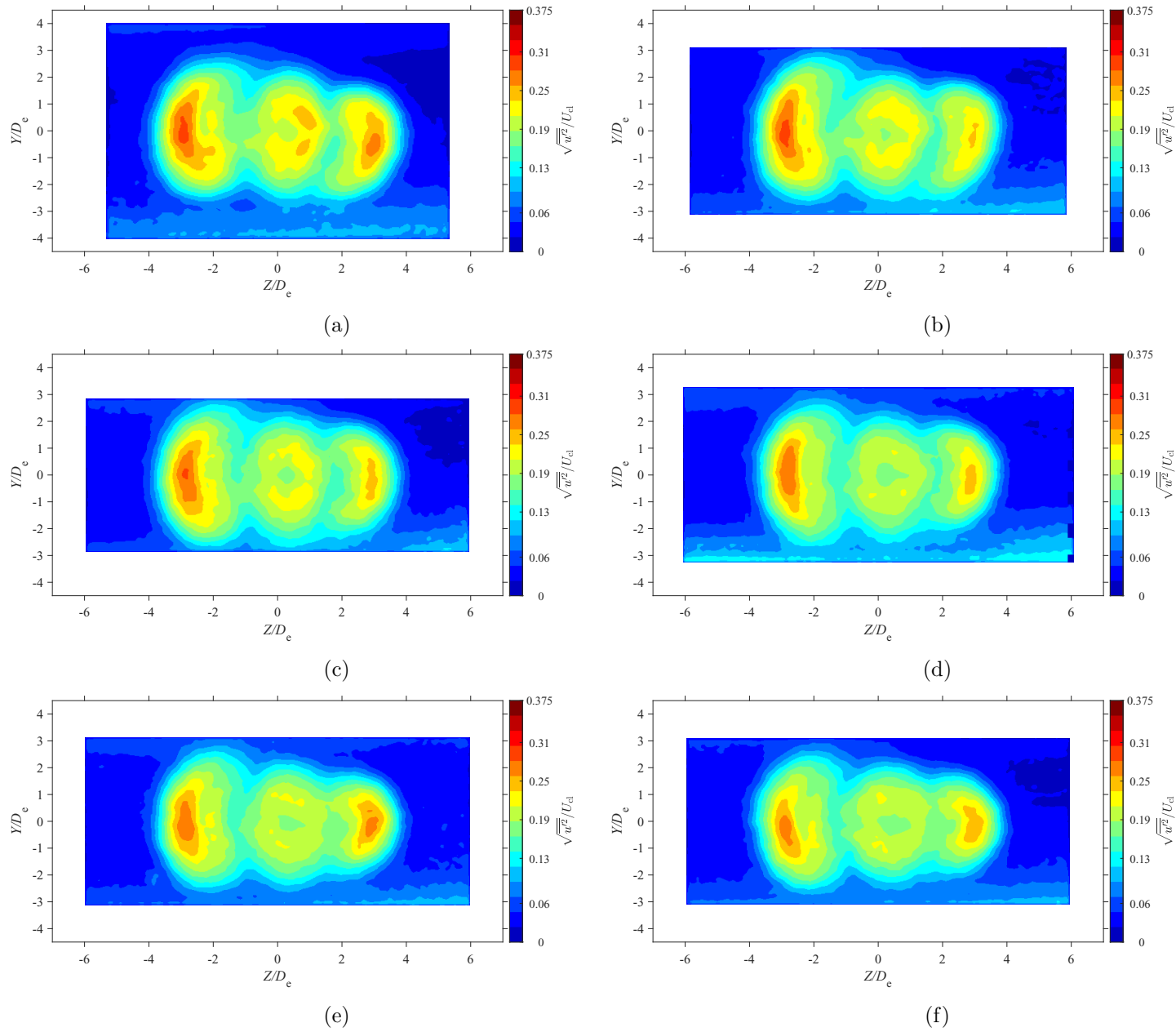


Figure E.39: Average streamwise turbulence intensity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

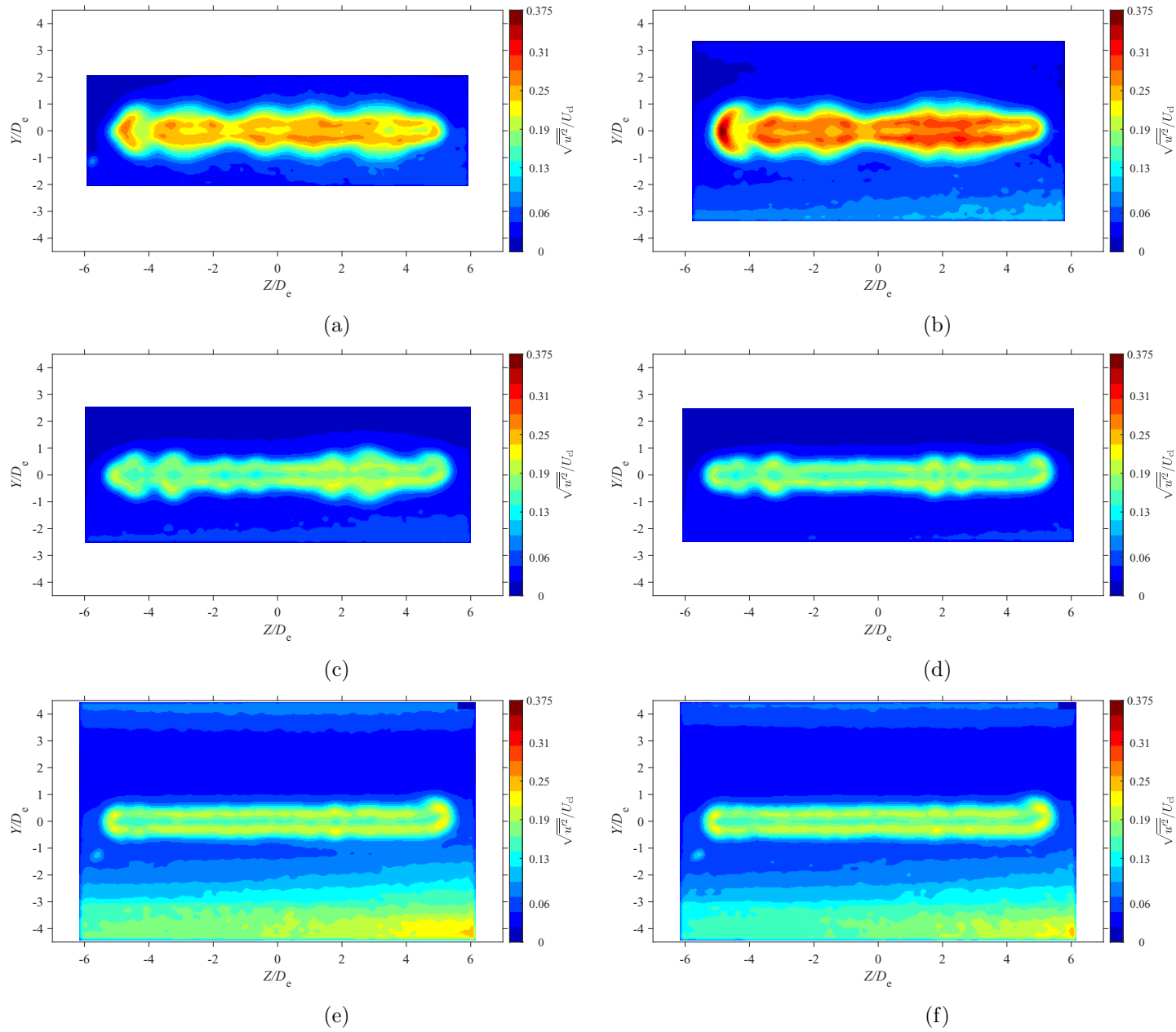


Figure E.40: Average streamwise turbulence intensity map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

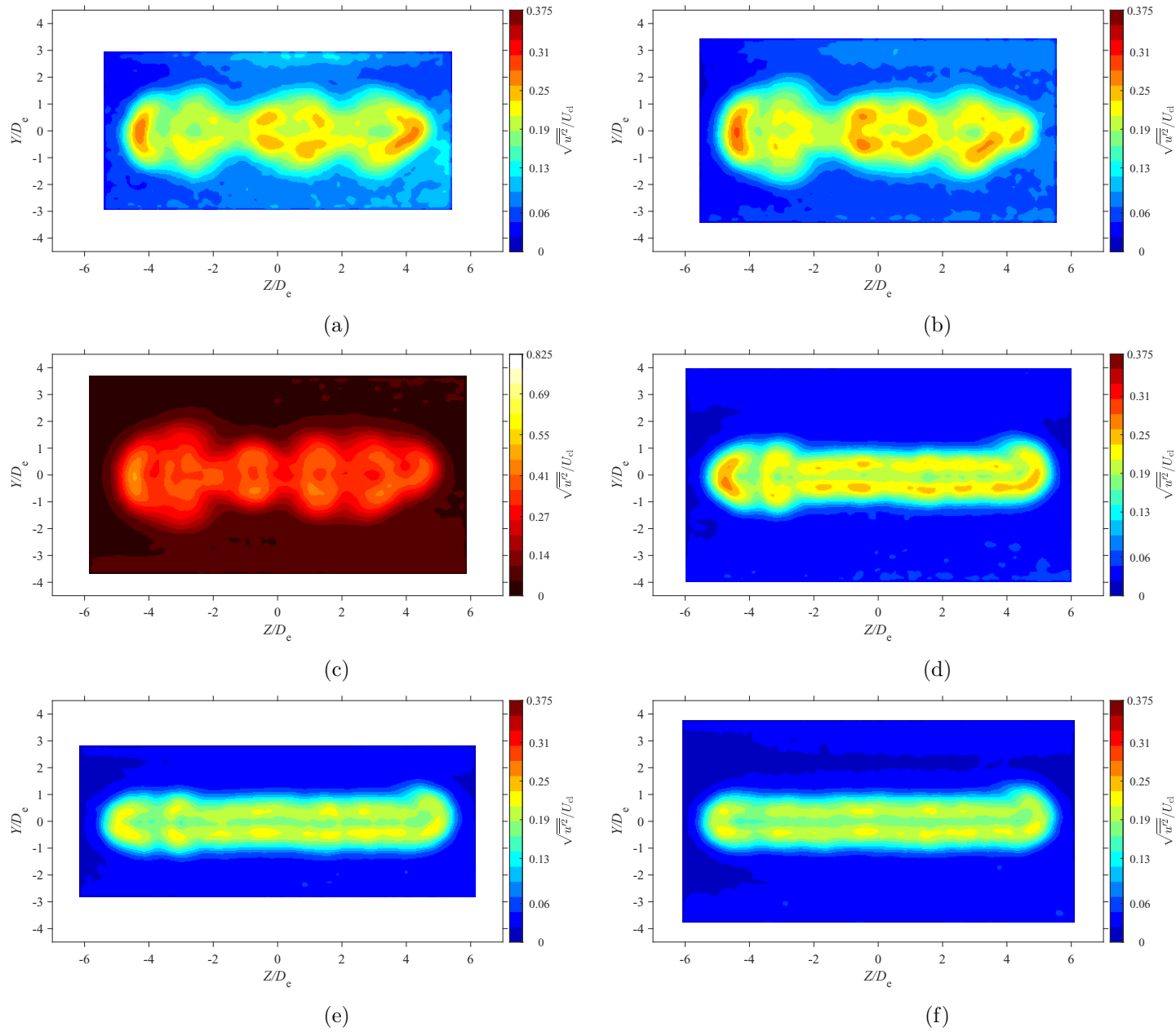


Figure E.41: Average streamwise turbulence intensity map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

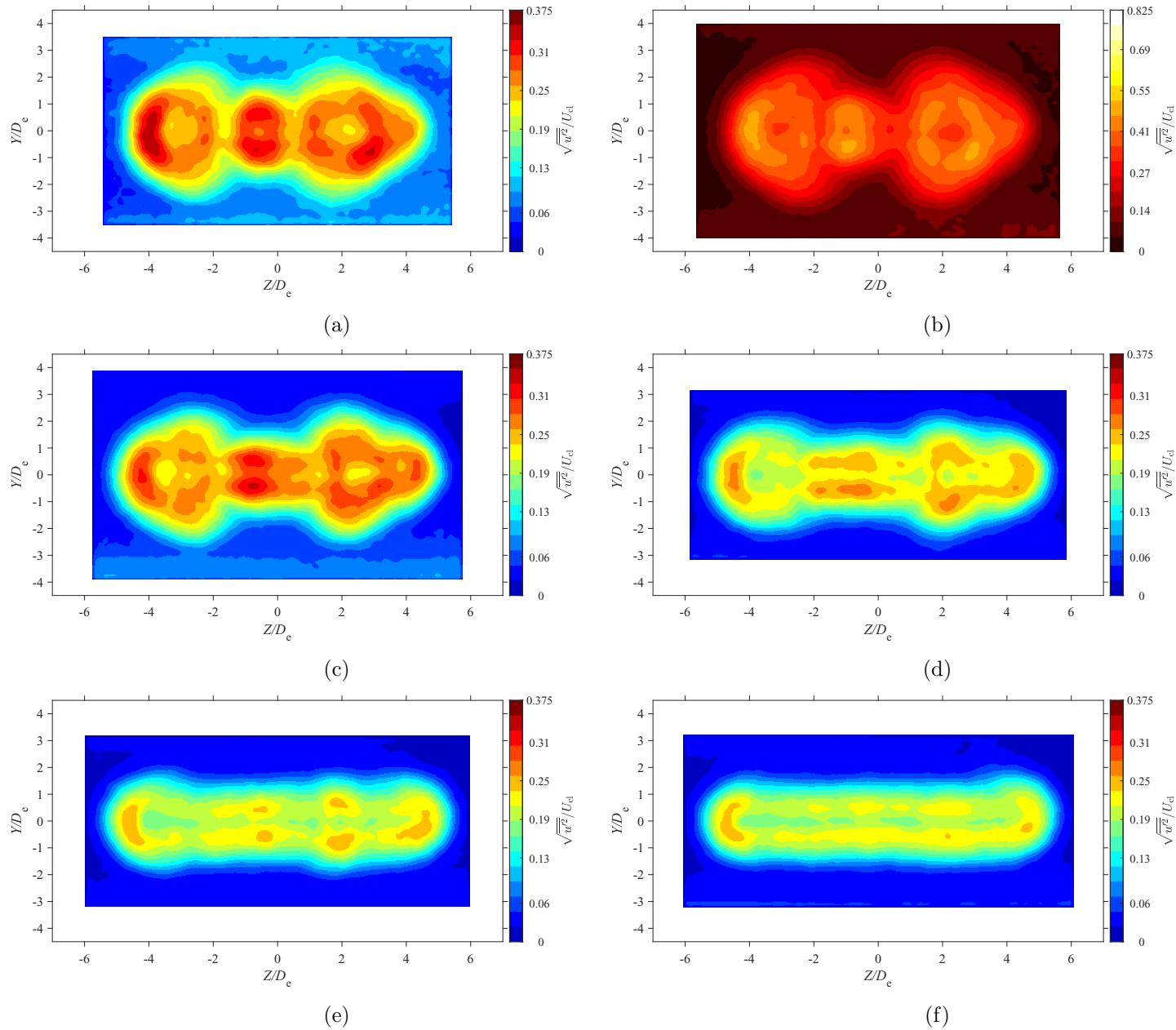


Figure E.42: Average streamwise turbulence intensity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

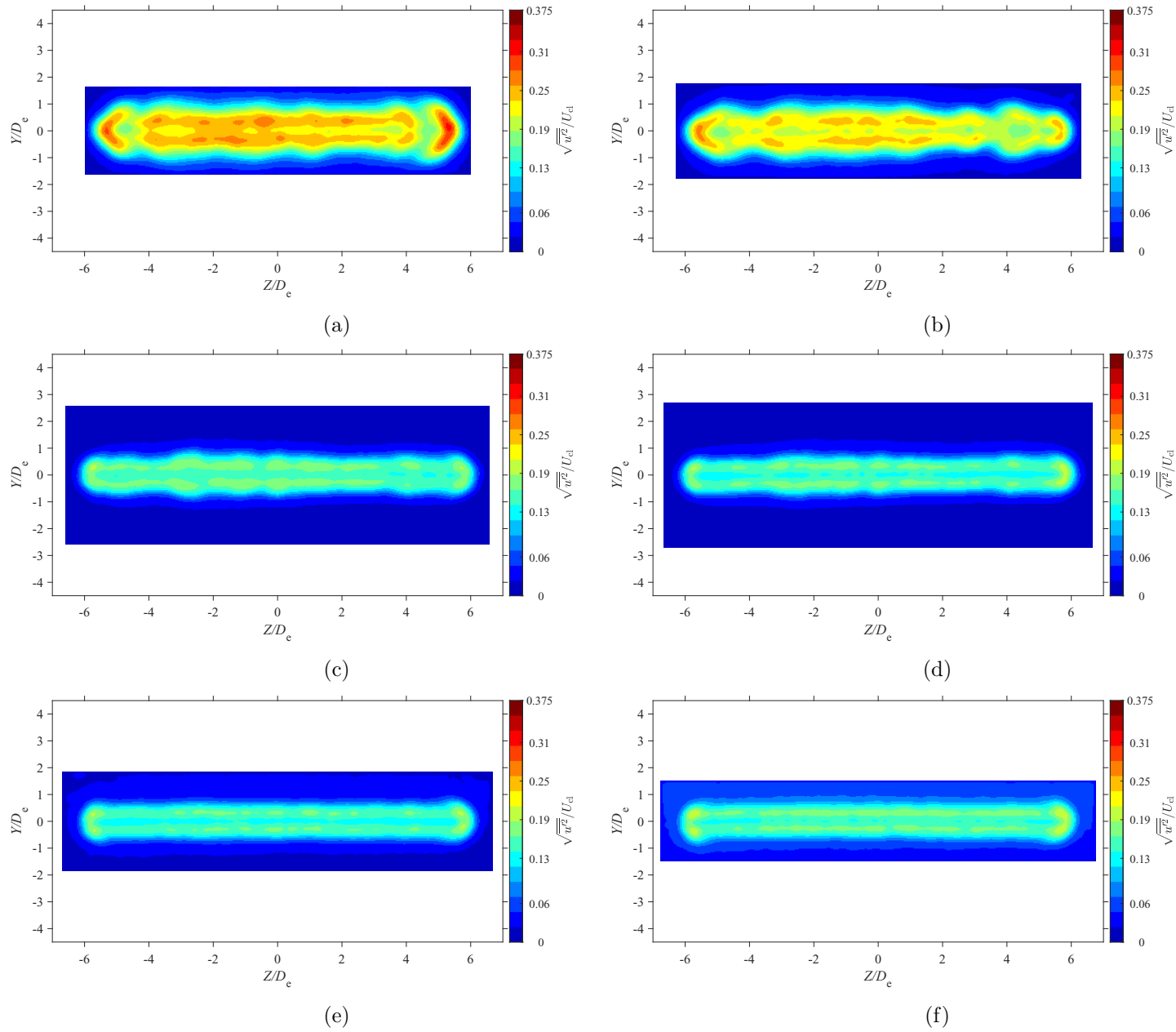


Figure E.43: Average streamwise turbulence intensity map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

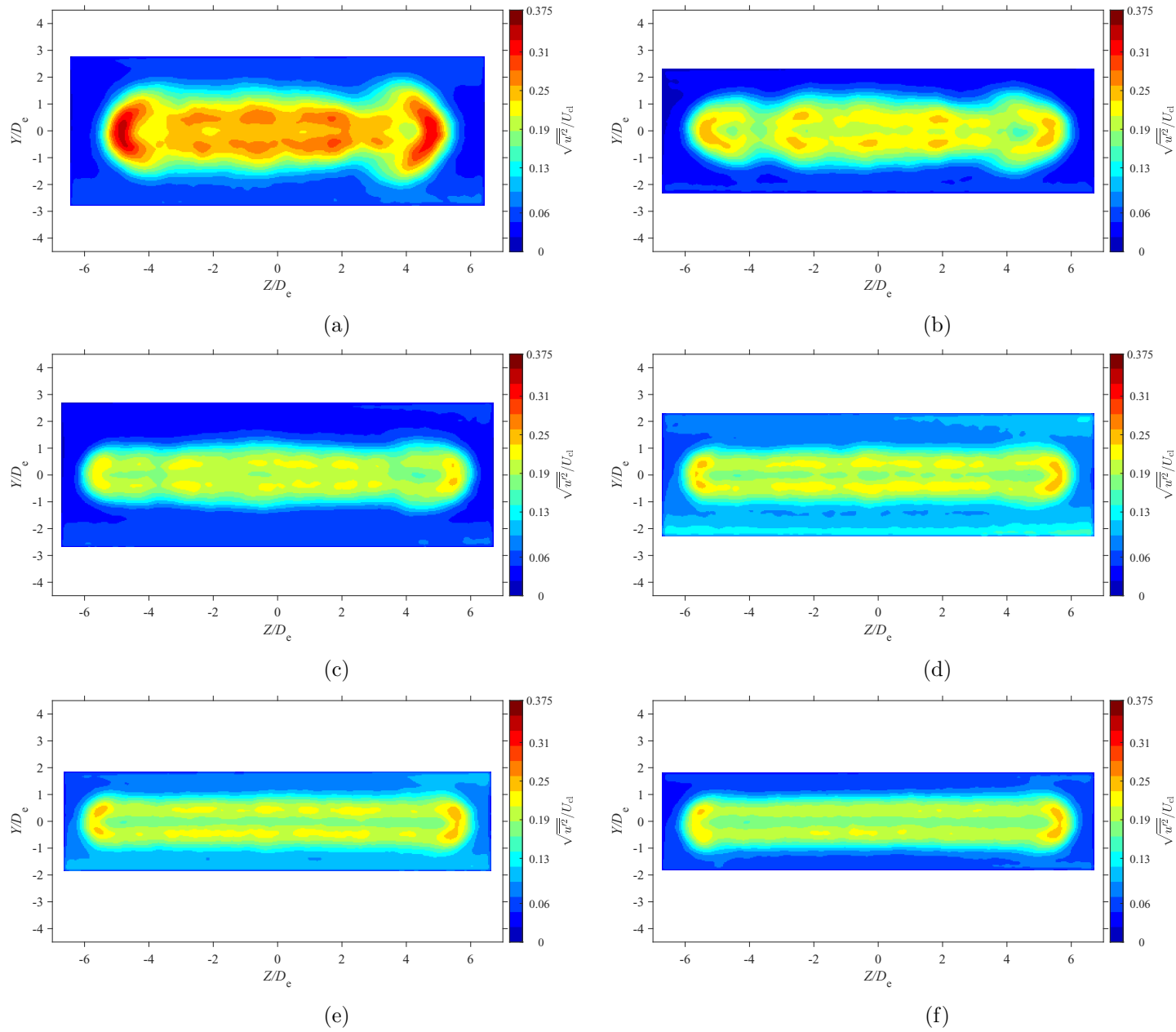


Figure E.44: Average streamwise turbulence intensity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

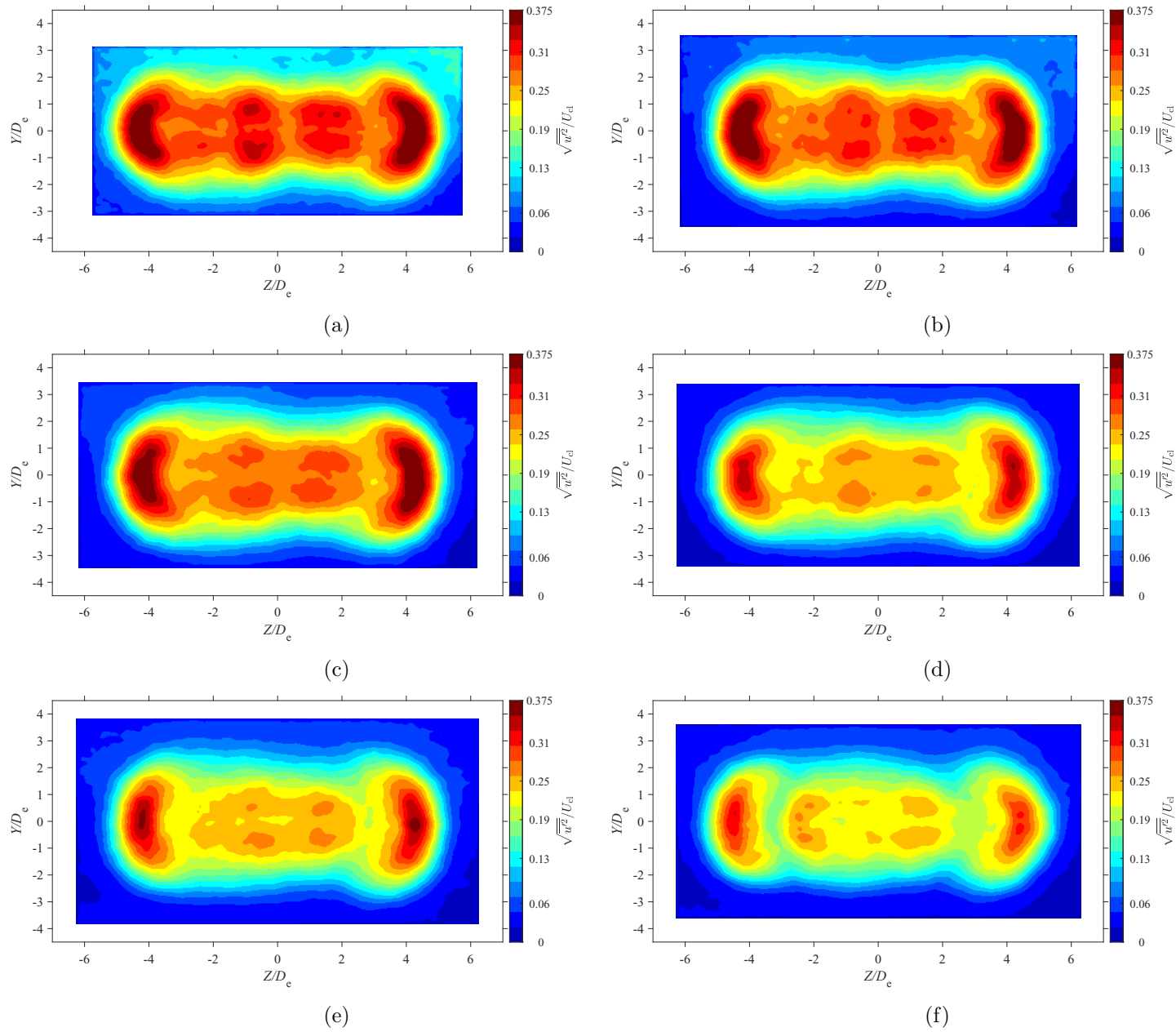


Figure E.45: Average streamwise turbulence intensity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

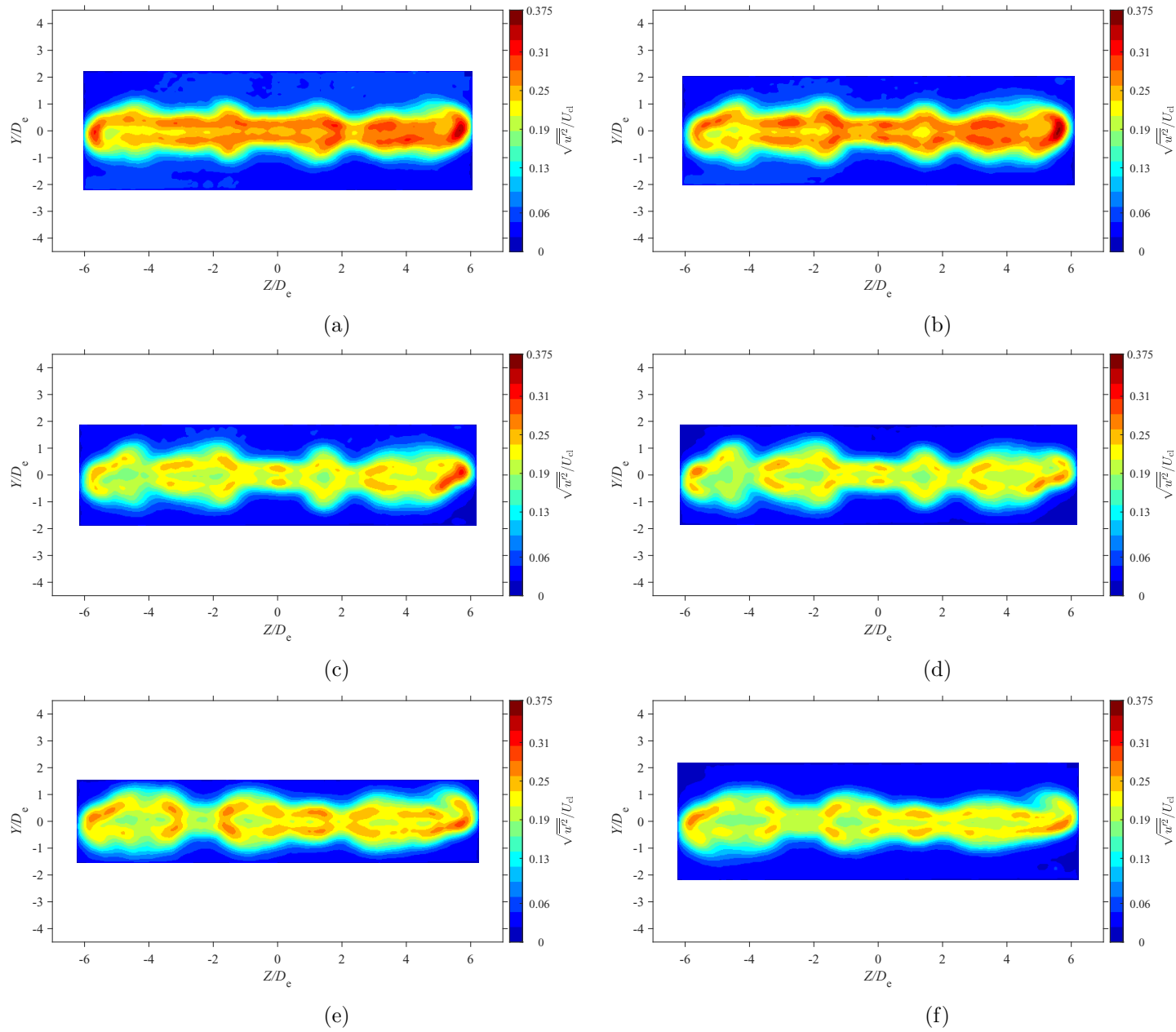


Figure E.46: Average streamwise turbulence intensity map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

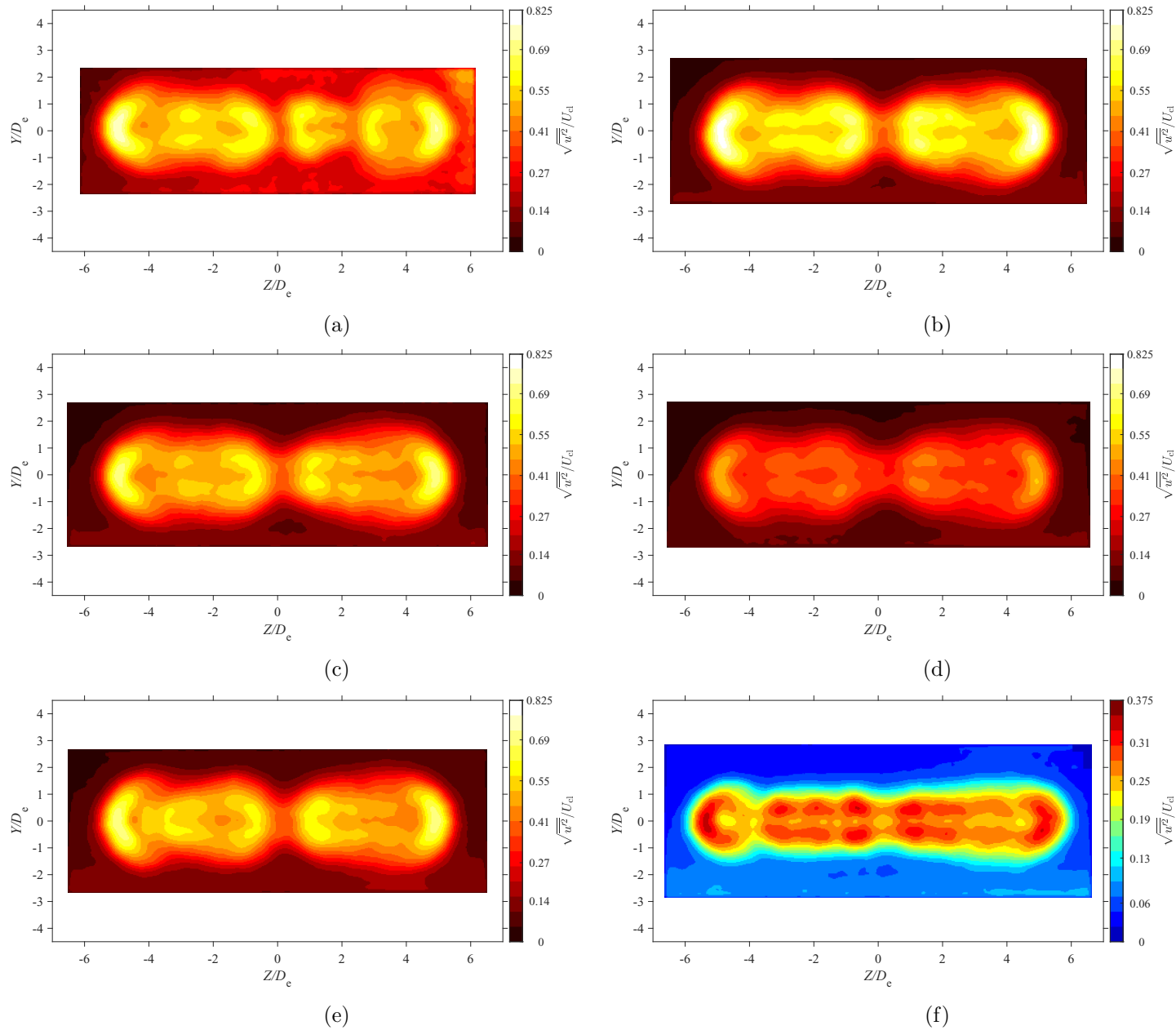


Figure E.47: Average streamwise turbulence intensity map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

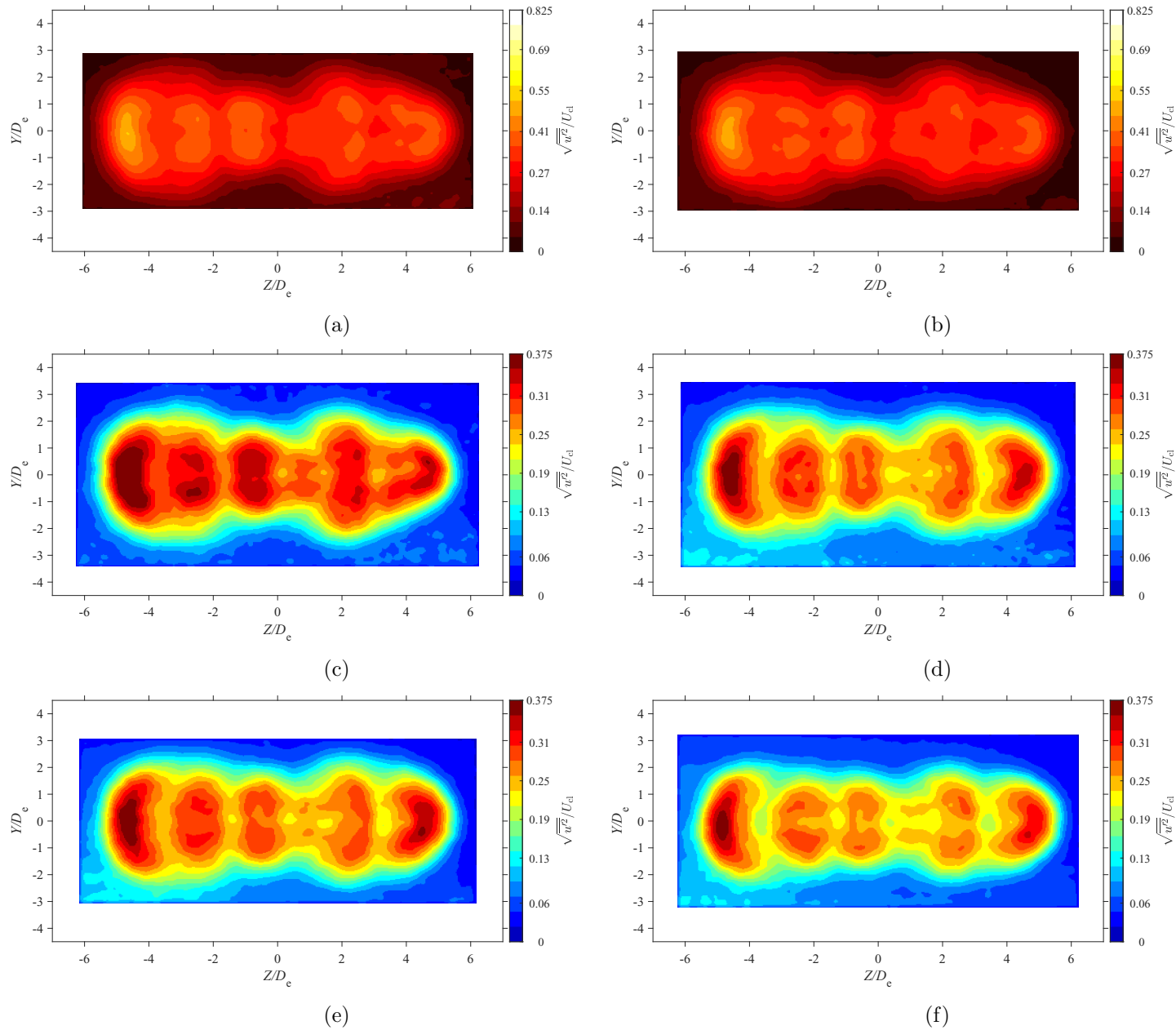


Figure E.48: Average streamwise turbulence intensity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

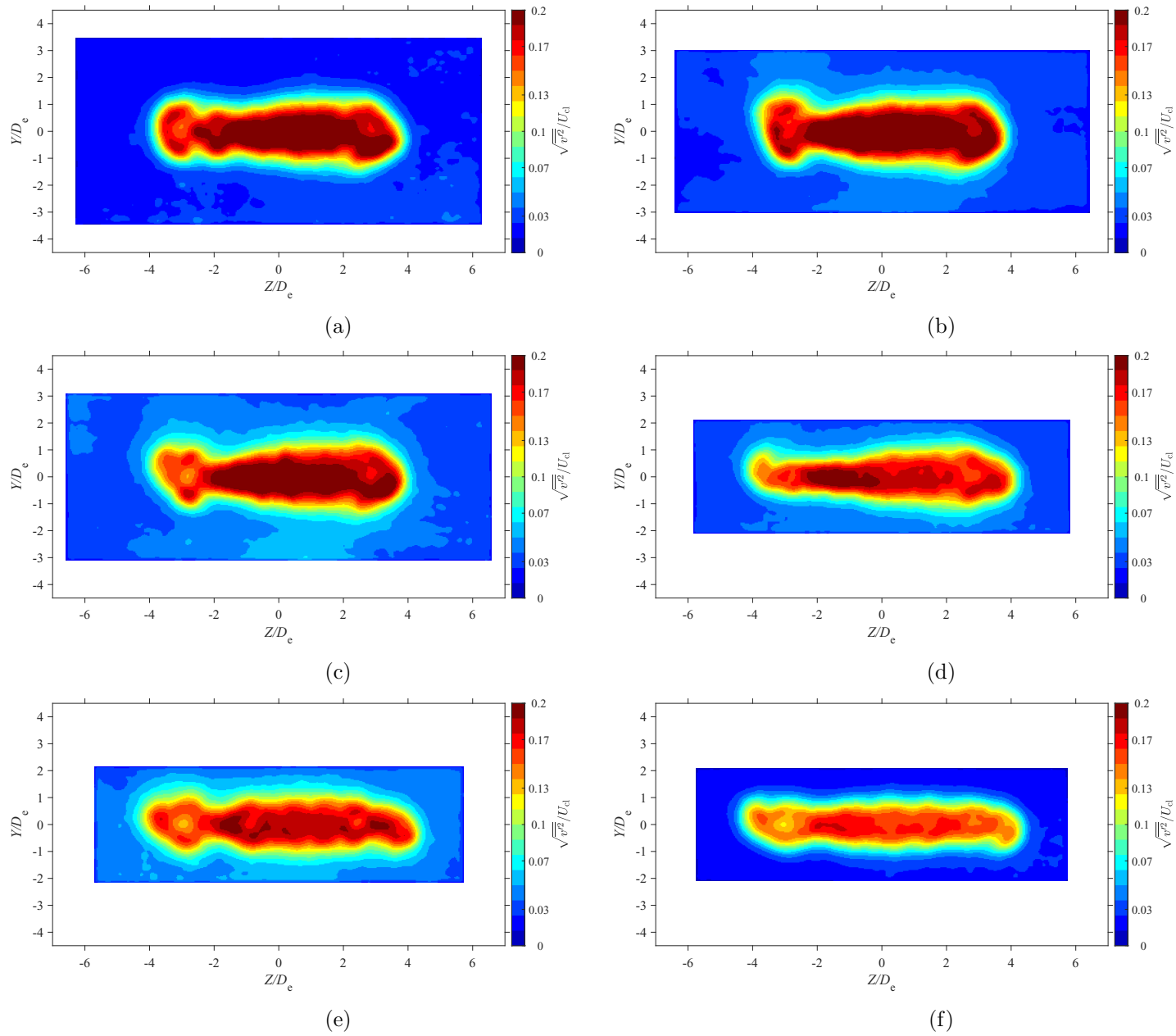


Figure E.49: Average lateral turbulence intensity map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

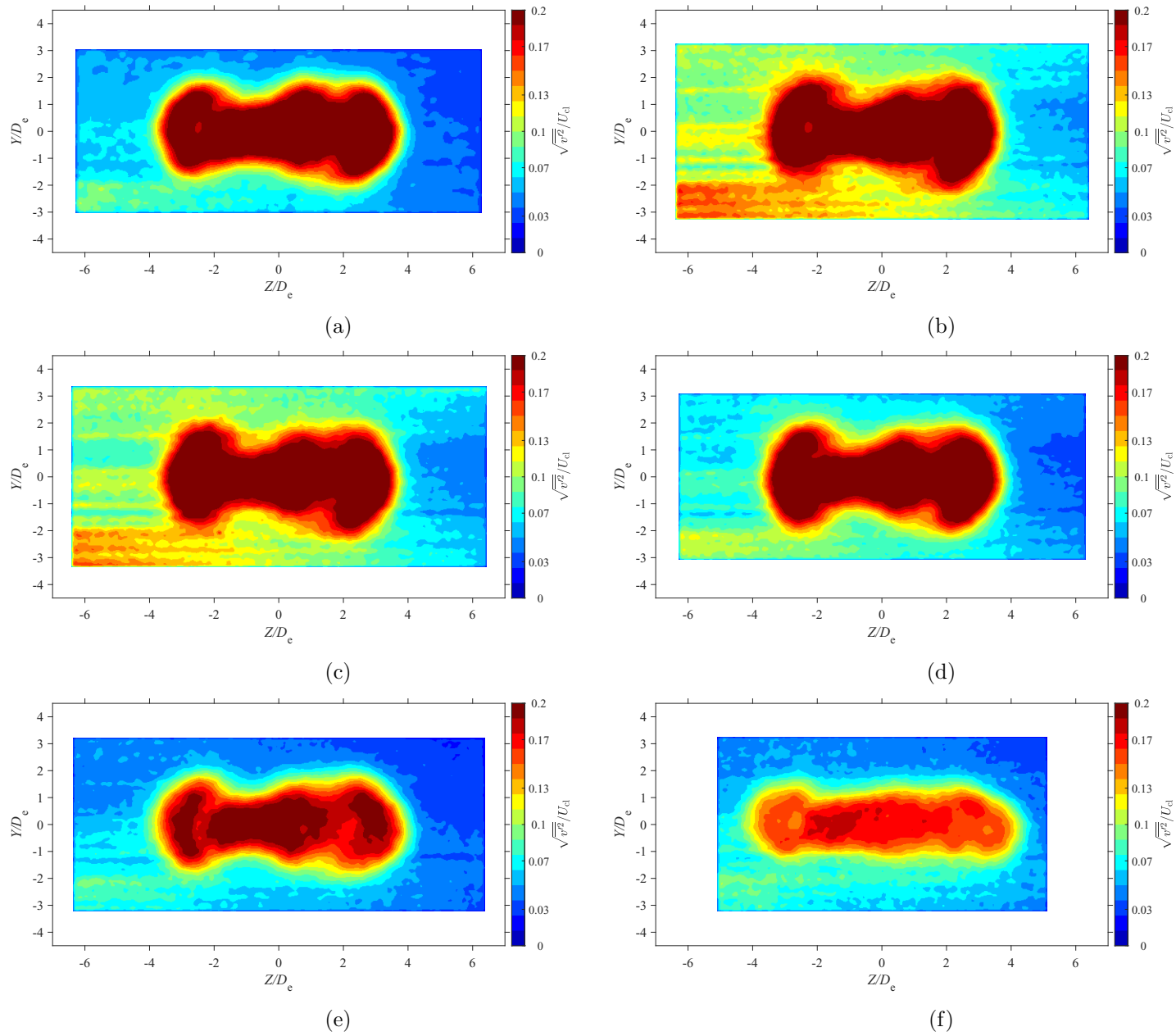


Figure E.50: Average lateral turbulence intensity map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

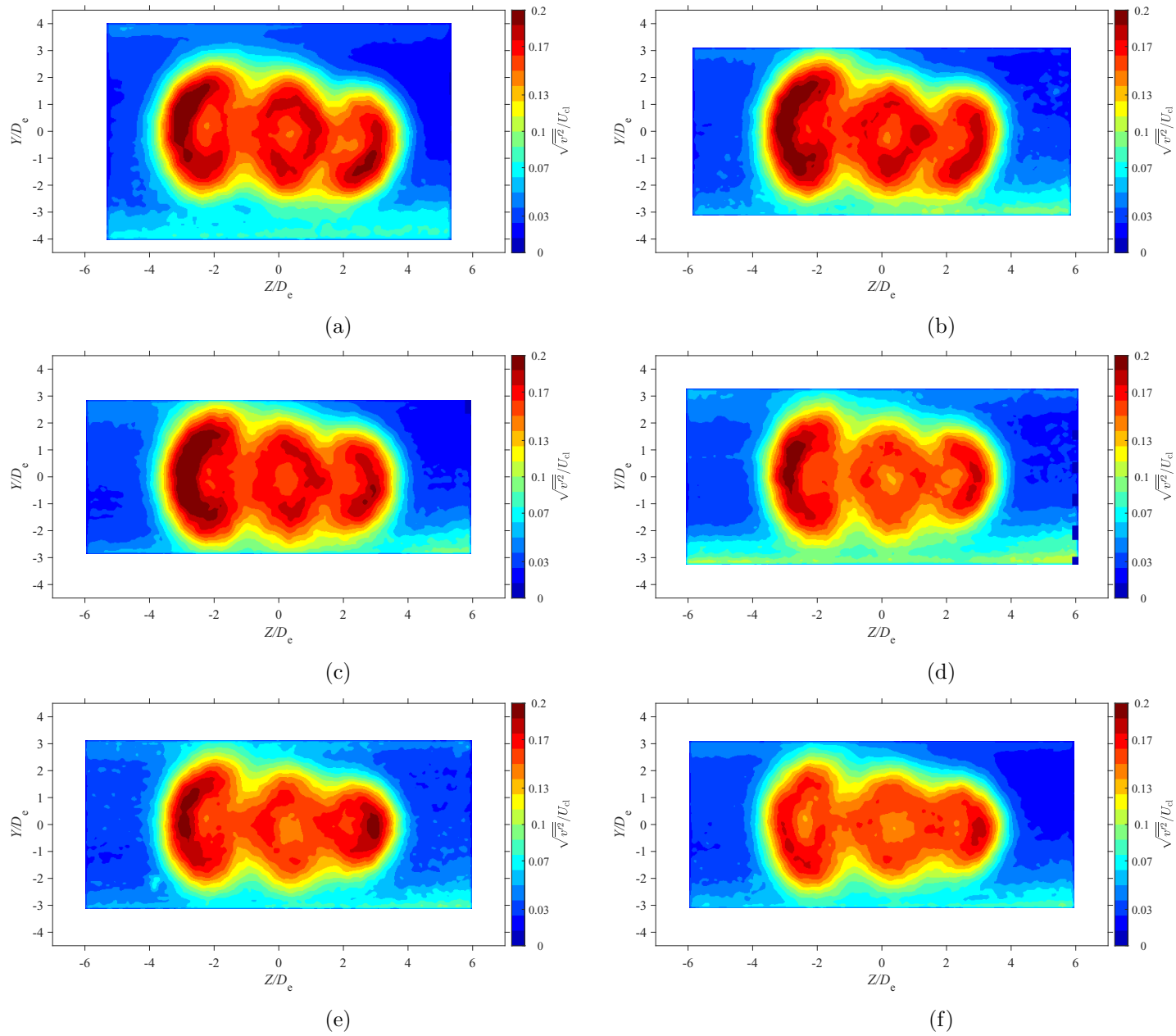


Figure E.51: Average lateral turbulence intensity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

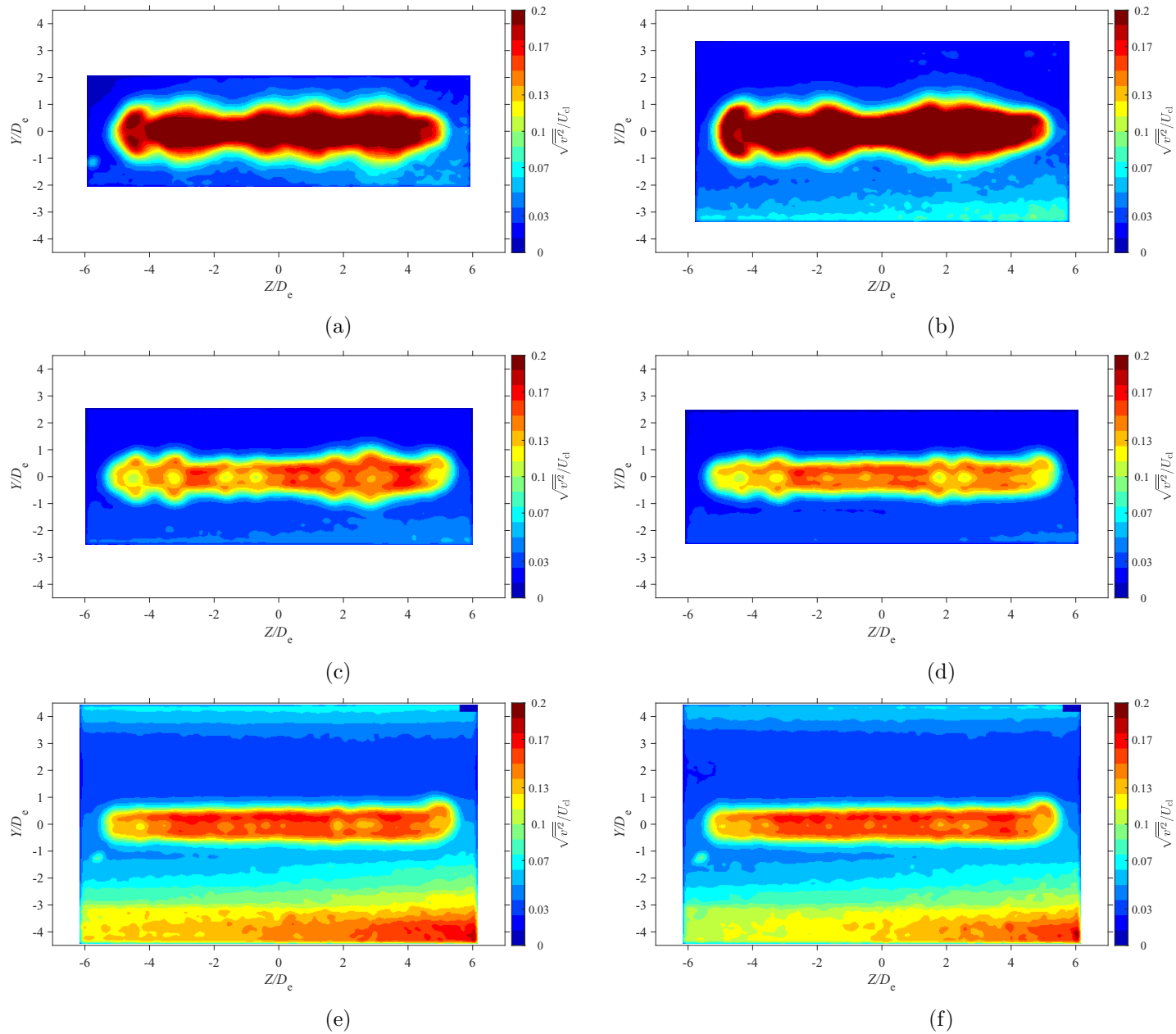


Figure E.52: Average lateral turbulence intensity map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

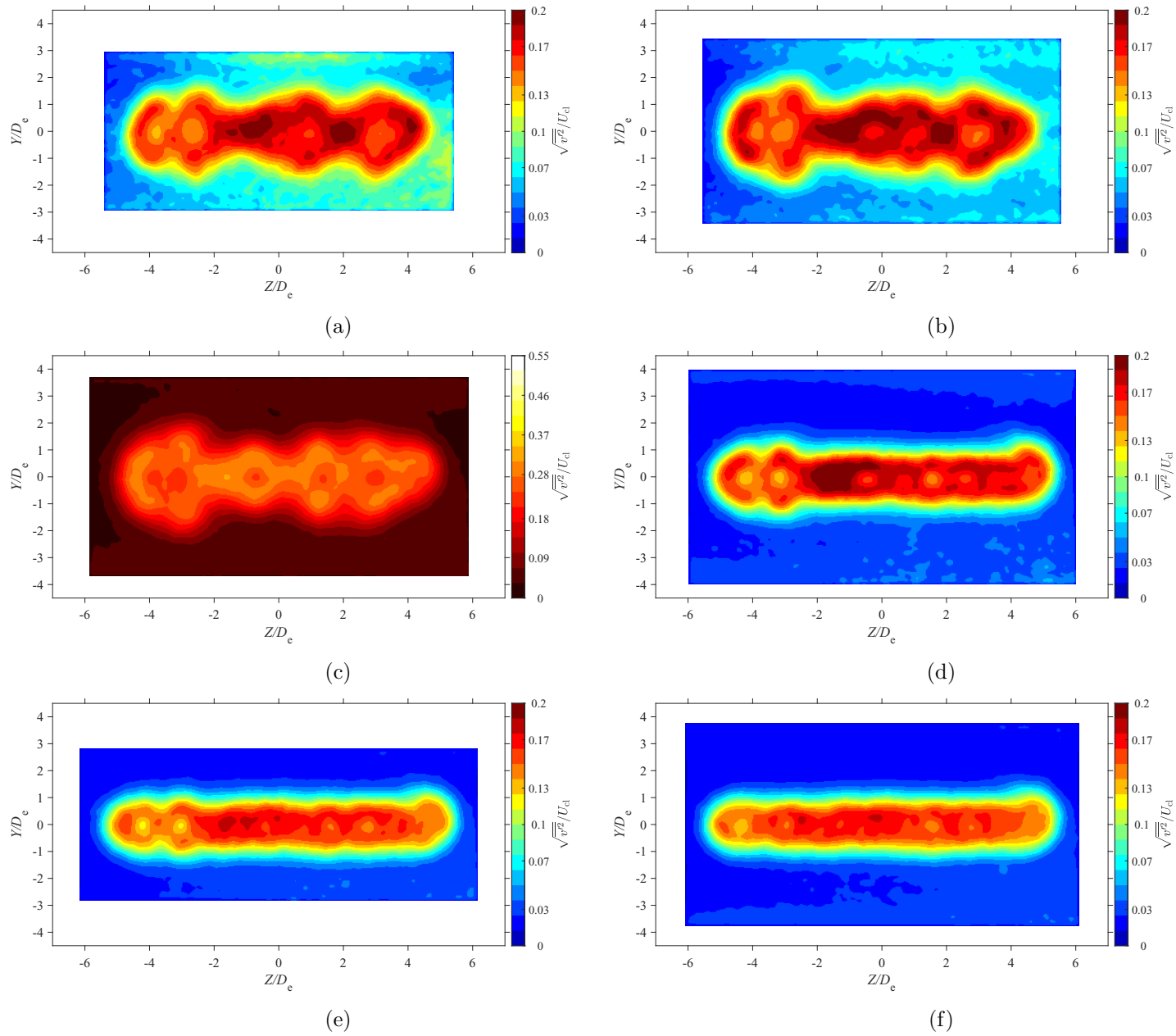


Figure E.53: Average lateral turbulence intensity map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

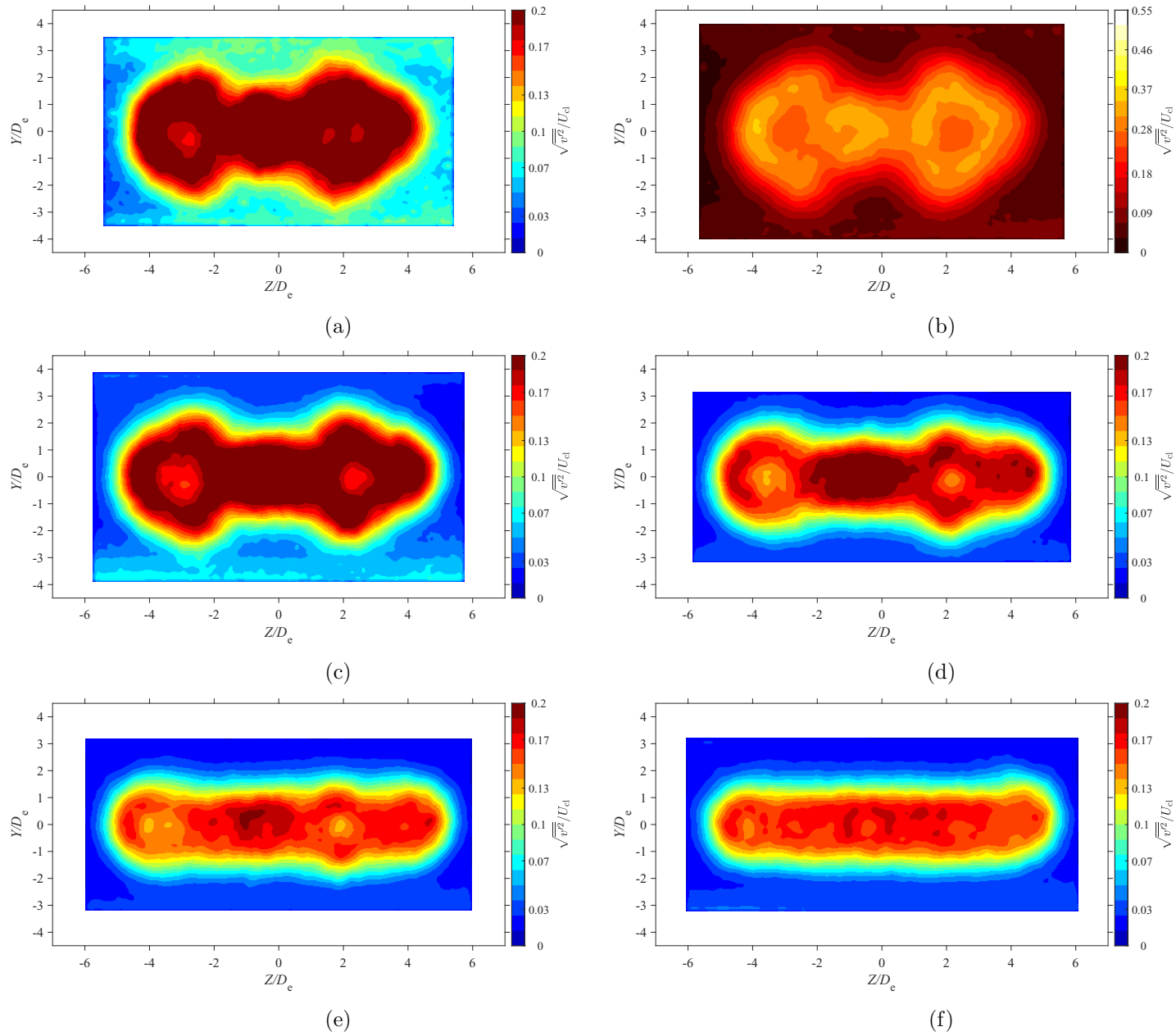


Figure E.54: Average lateral turbulence intensity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

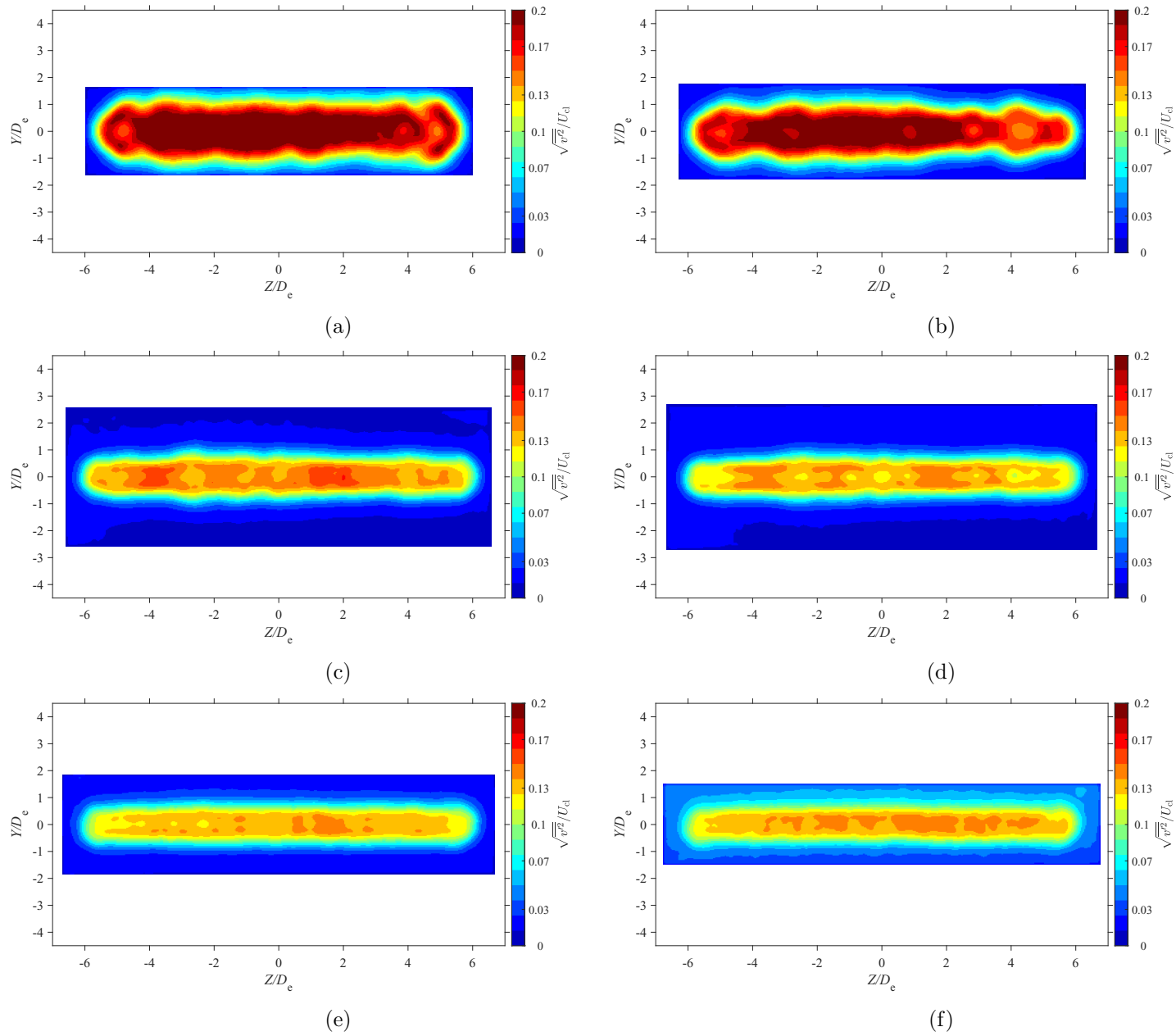


Figure E.55: Average lateral turbulence intensity map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

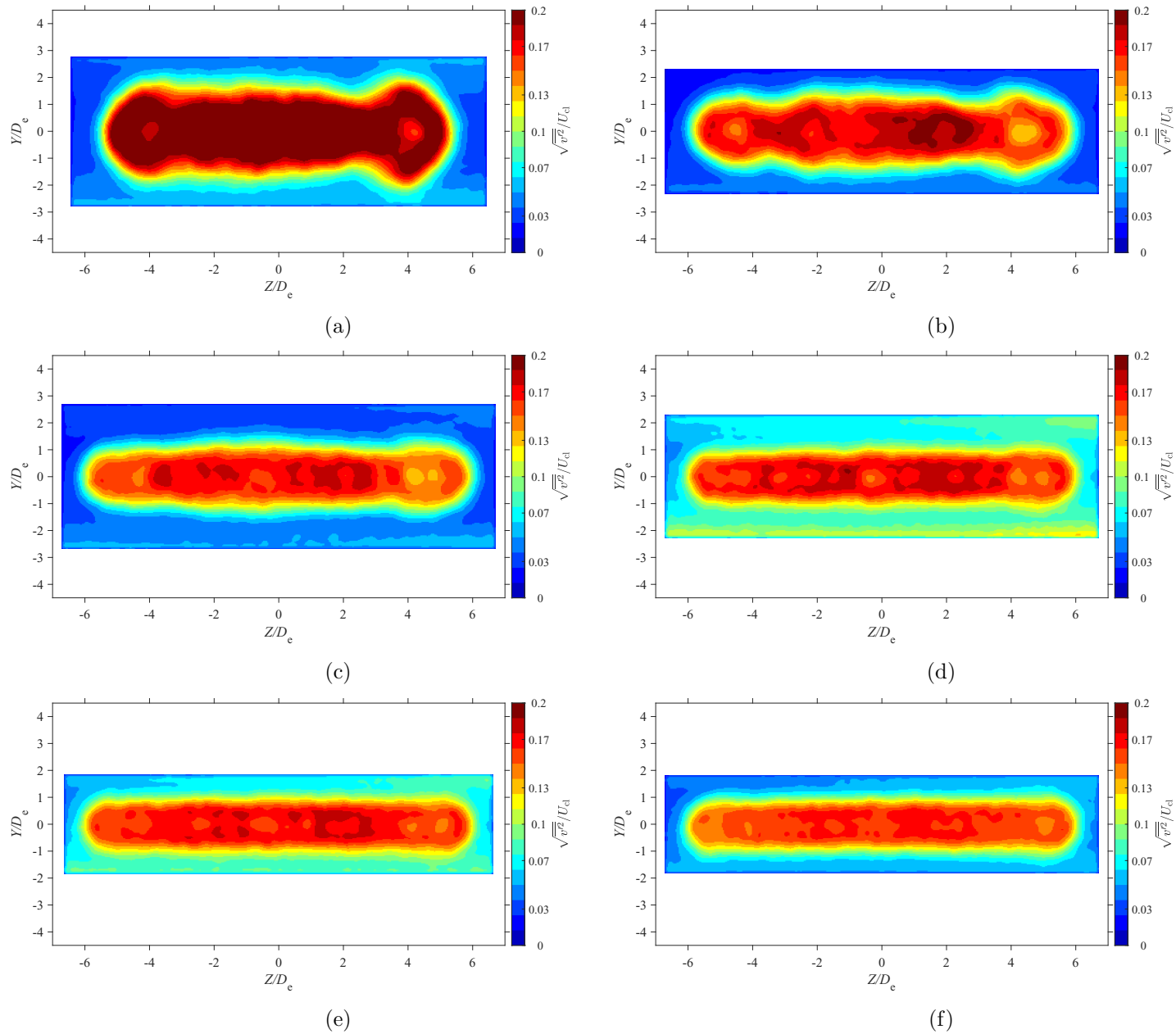


Figure E.56: Average lateral turbulence intensity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

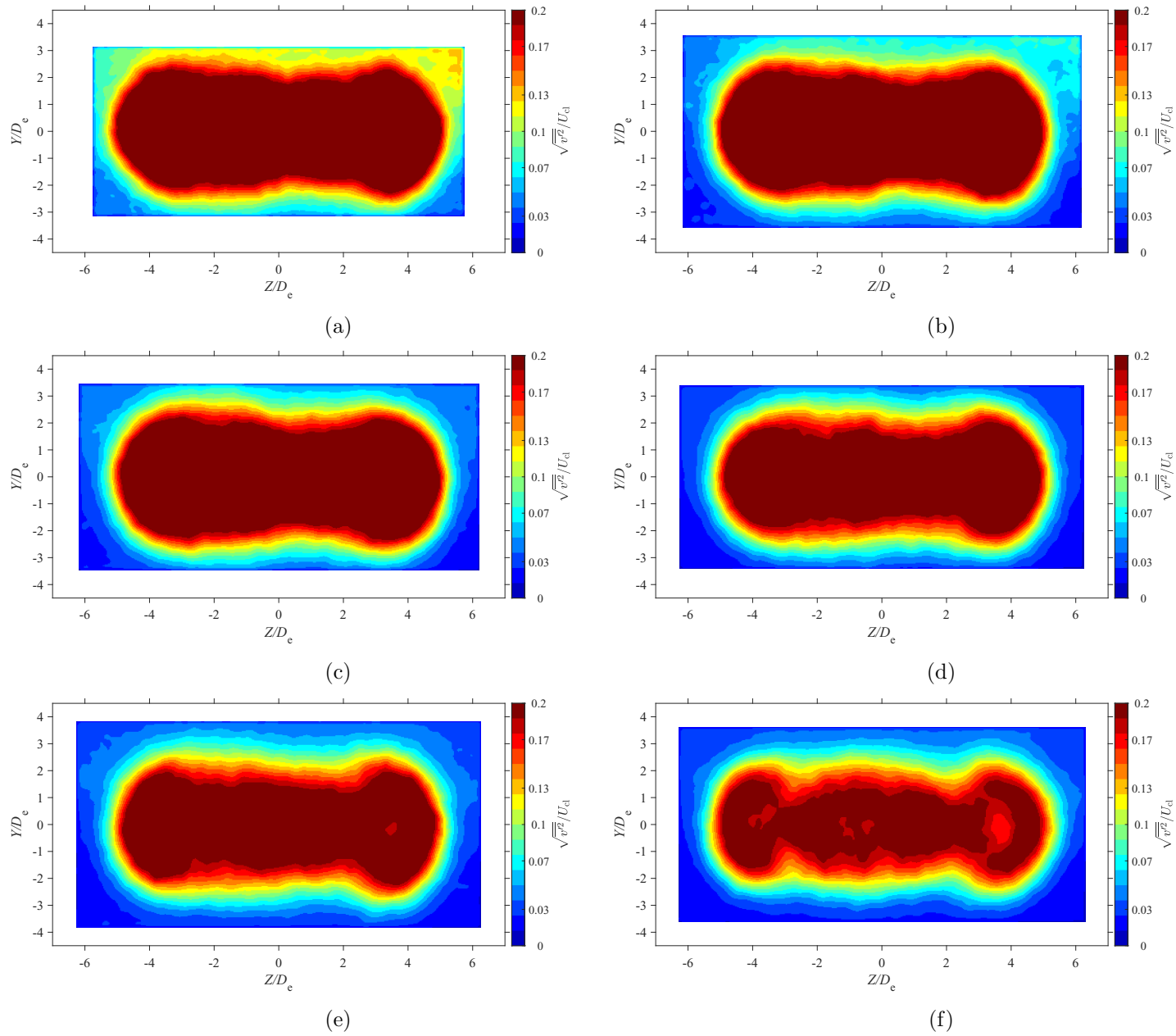


Figure E.57: Average lateral turbulence intensity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

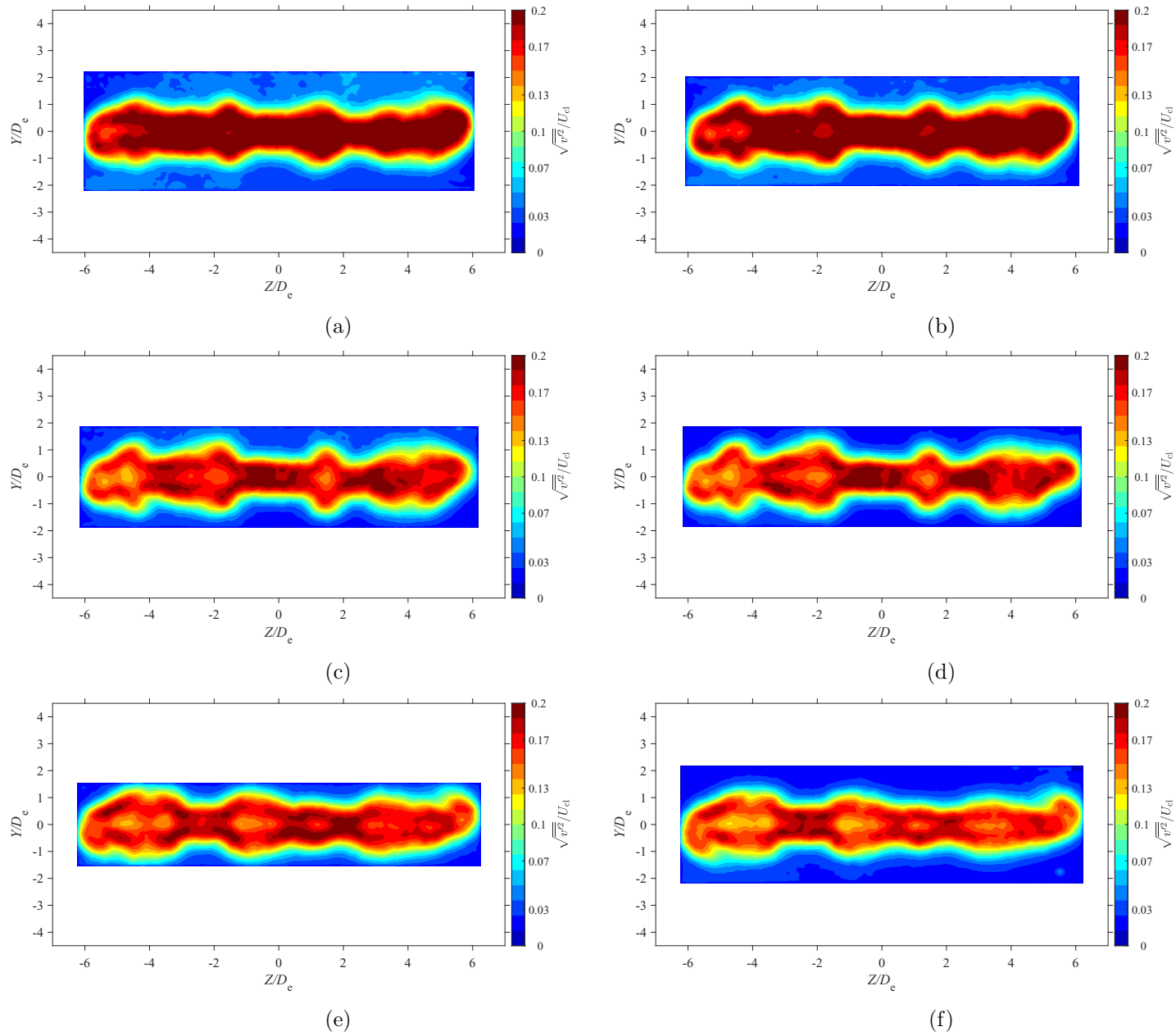


Figure E.58: Average lateral turbulence intensity map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

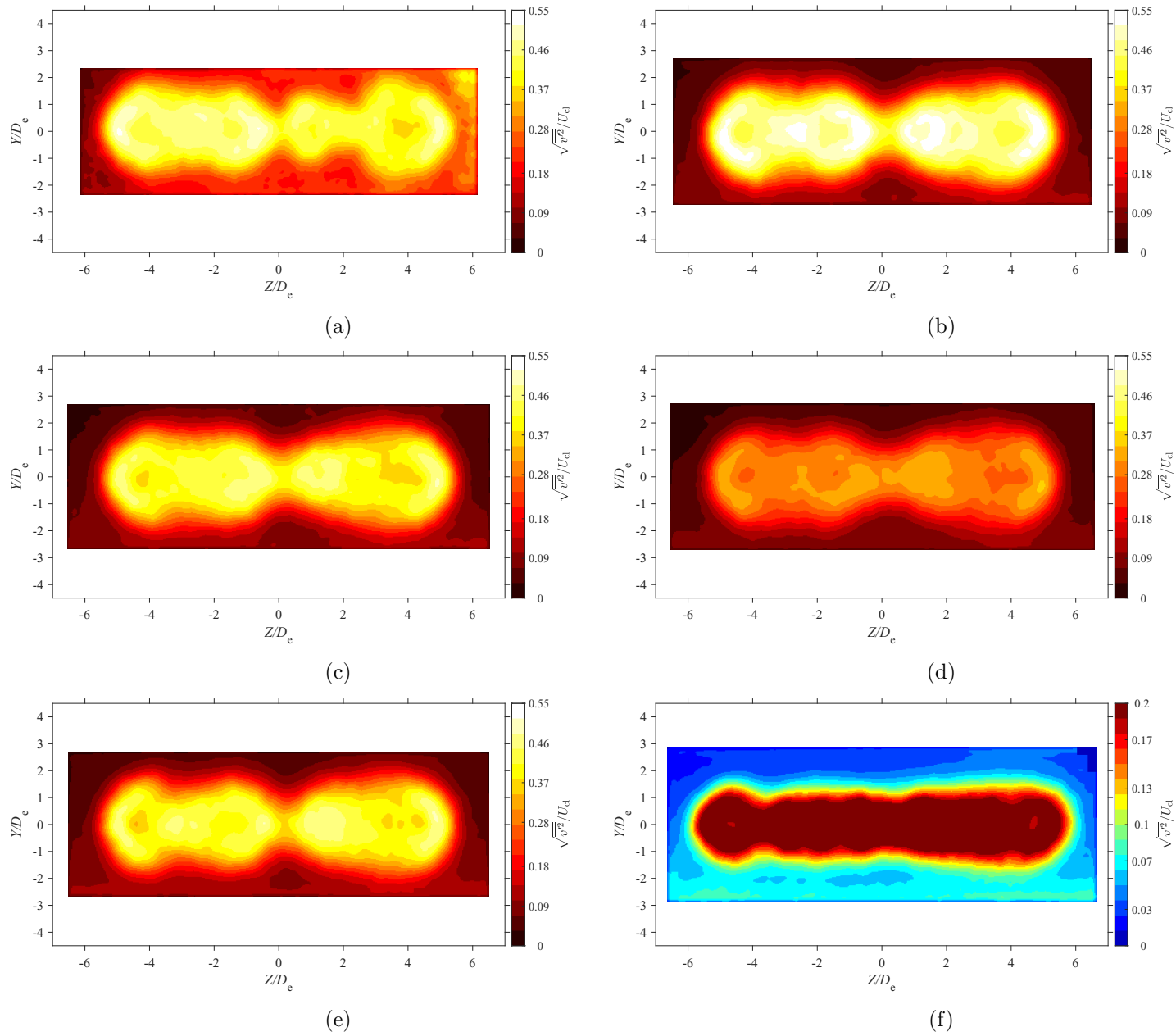


Figure E.59: Average lateral turbulence intensity map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

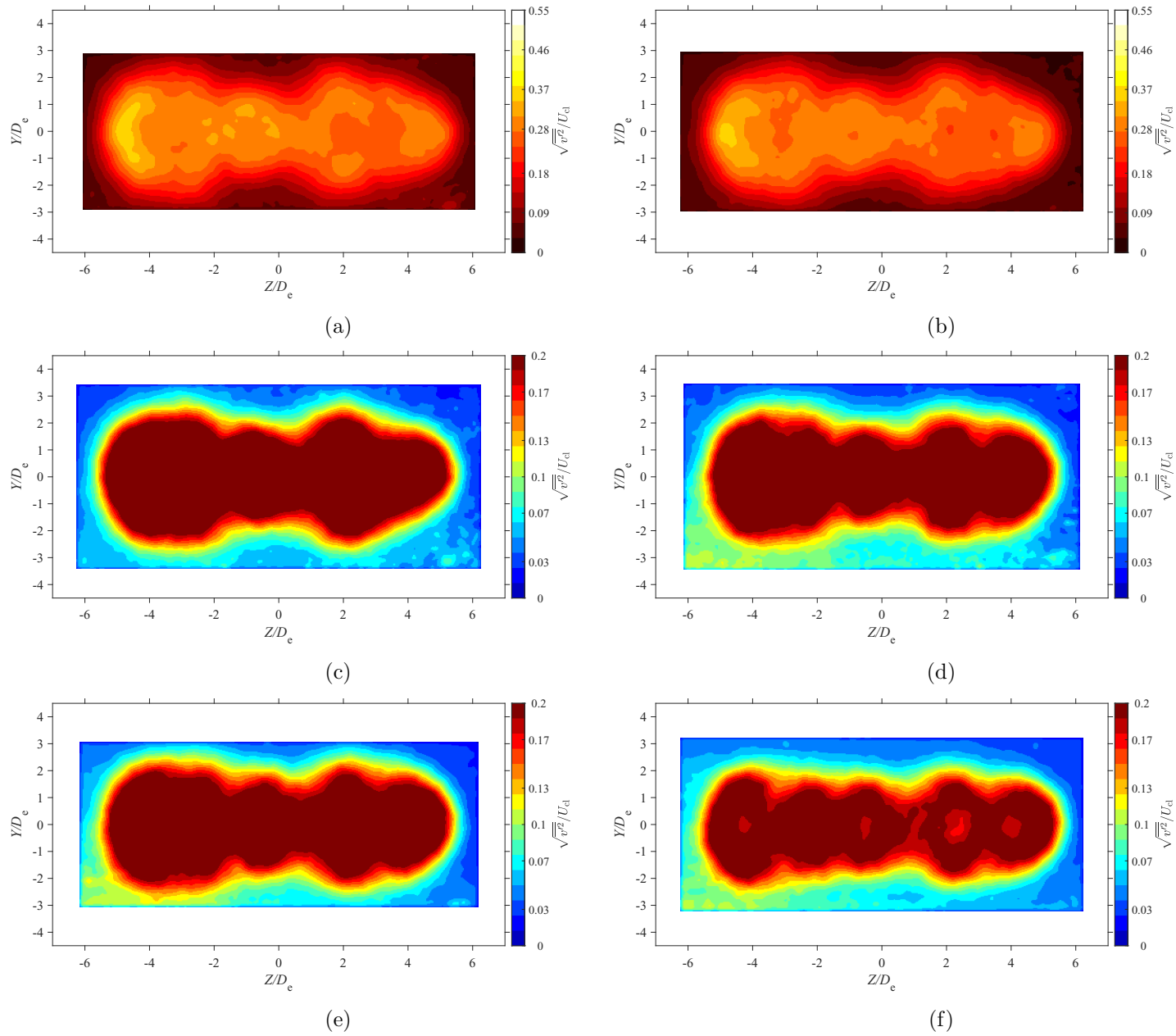


Figure E.60: Average lateral turbulence intensity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

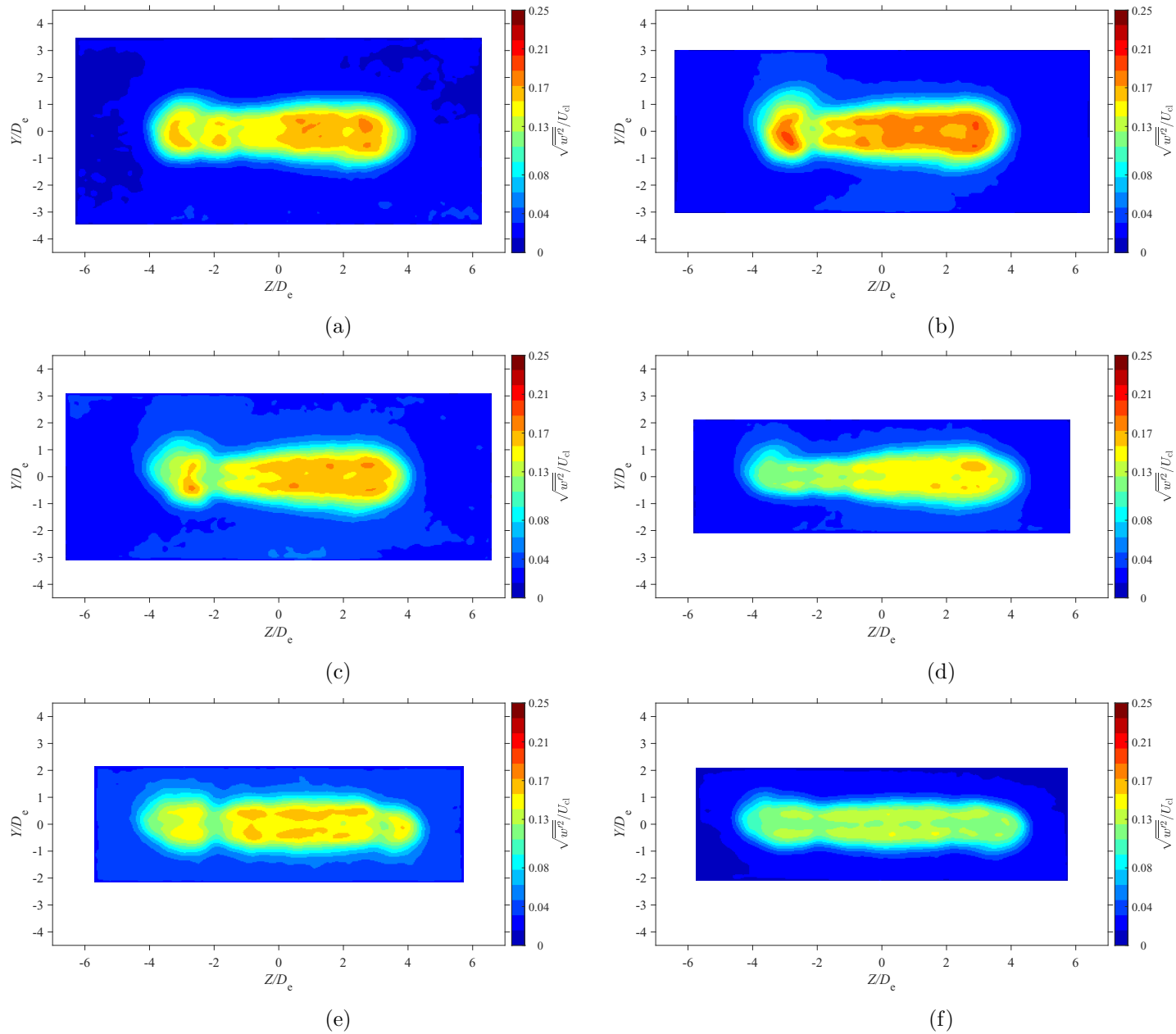


Figure E.61: Average spanwise turbulence intensity map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

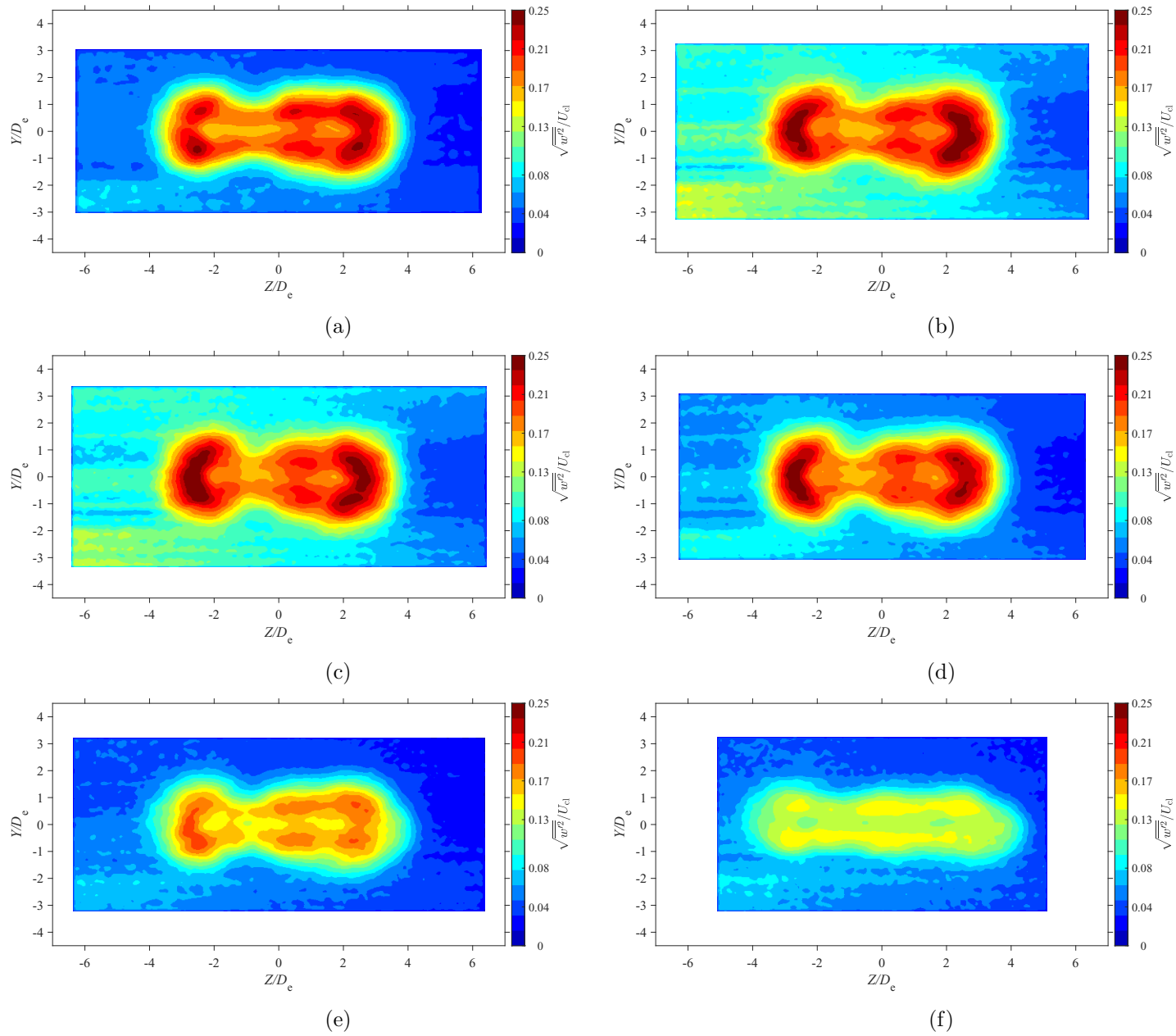


Figure E.62: Average spanwise turbulence intensity map at location $X = 5De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

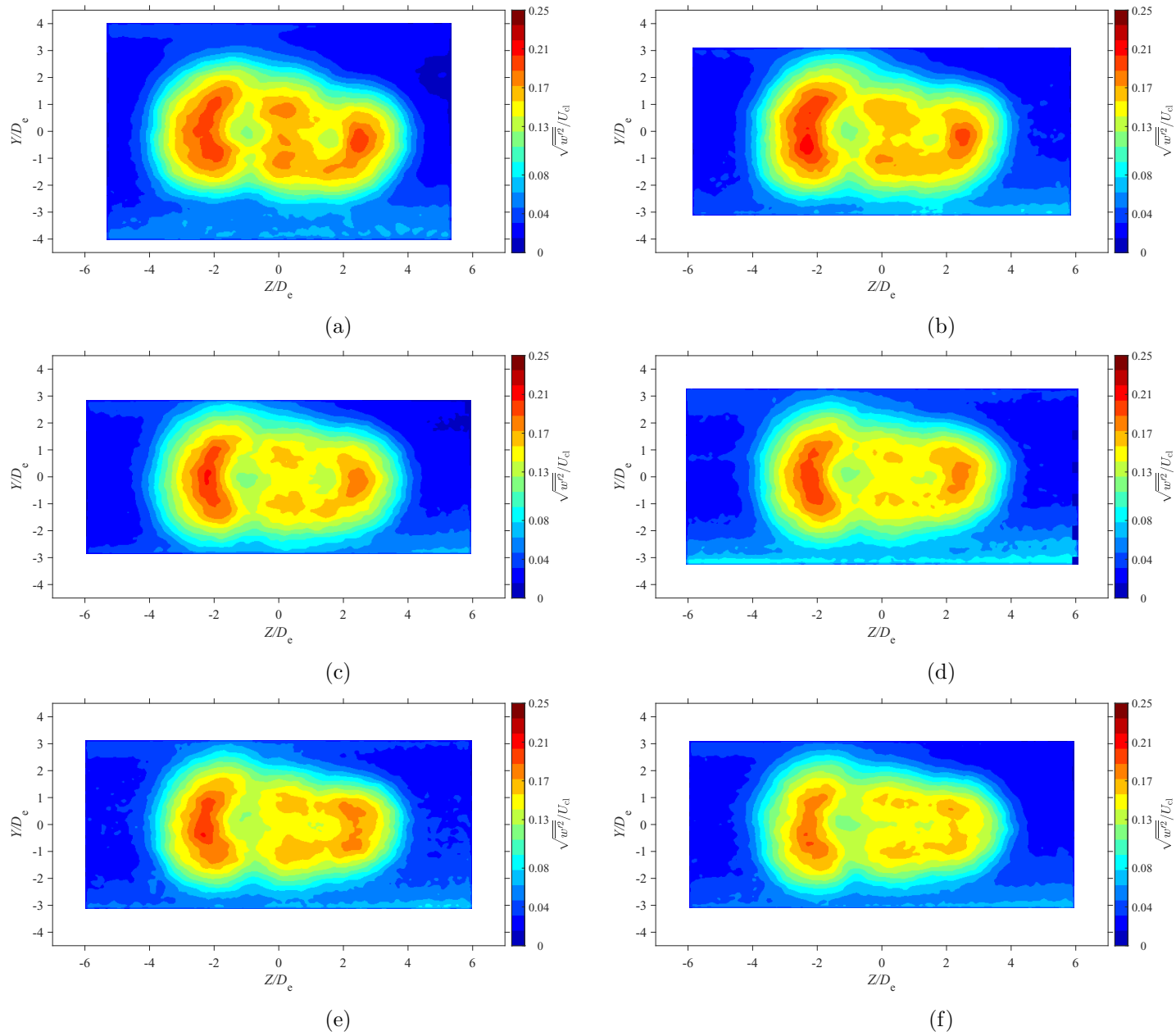


Figure E.63: Average spanwise turbulence intensity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

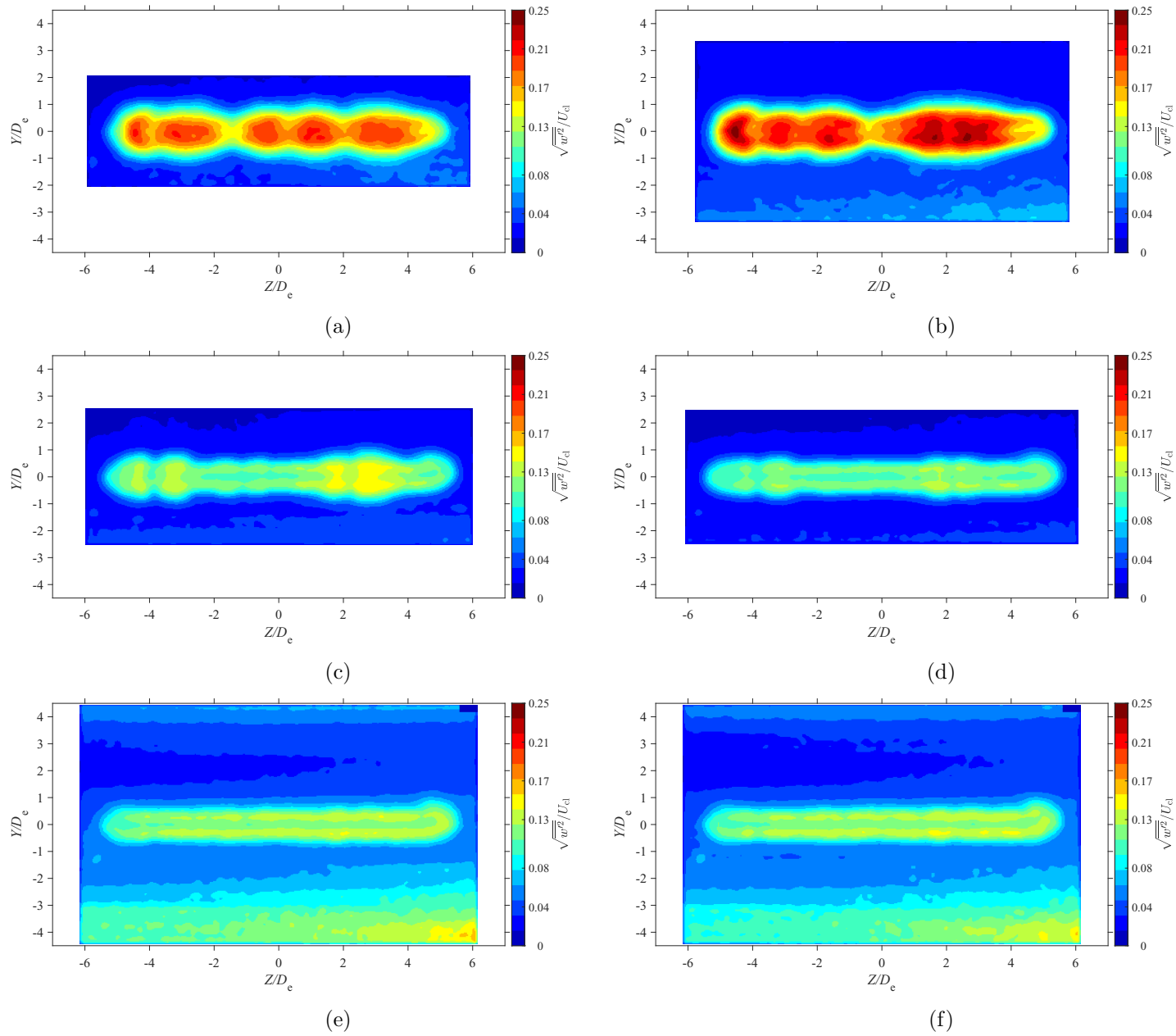


Figure E.64: Average spanwise turbulence intensity map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

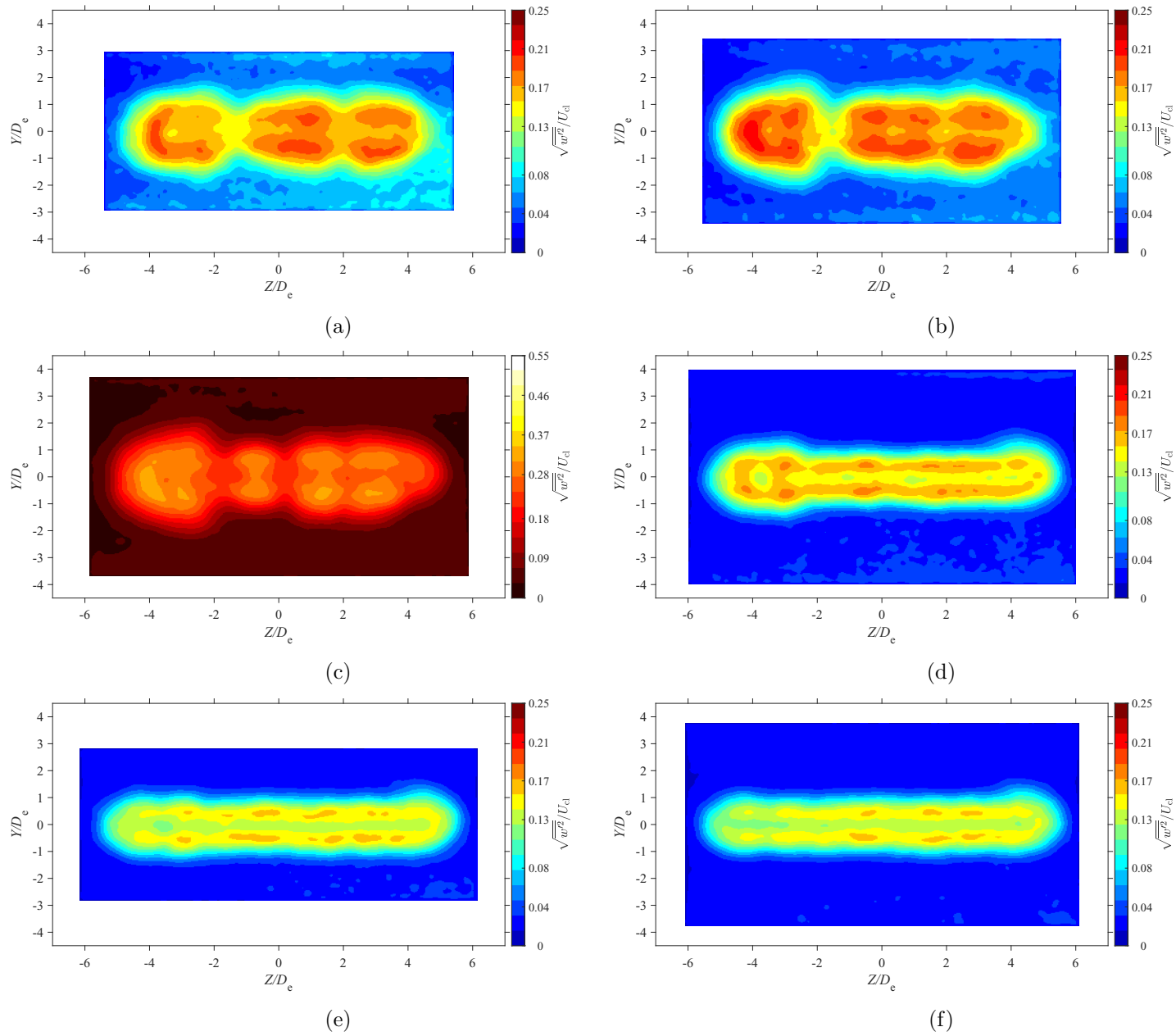


Figure E.65: Average spanwise turbulence intensity map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

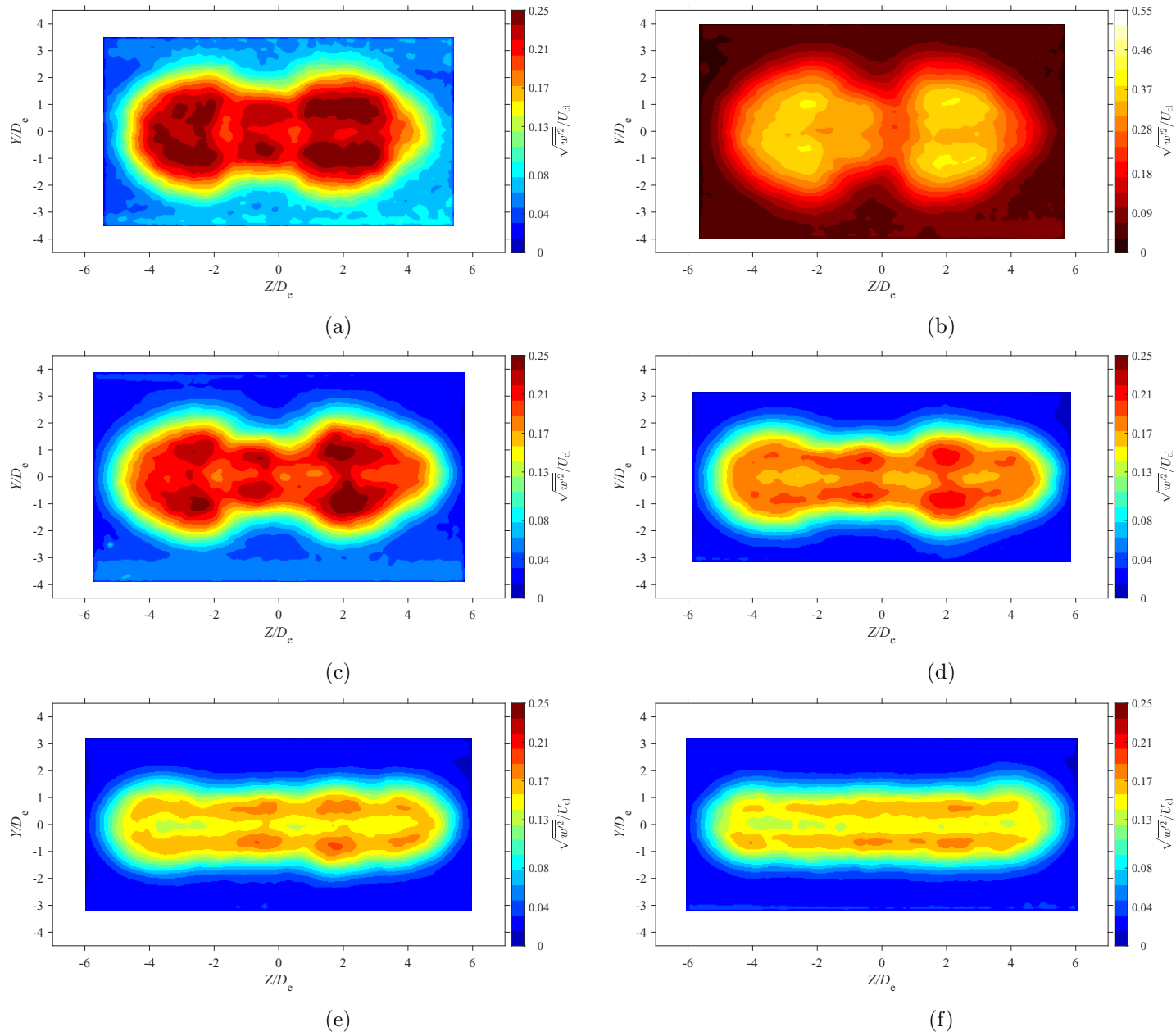


Figure E.66: Average spanwise turbulence intensity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

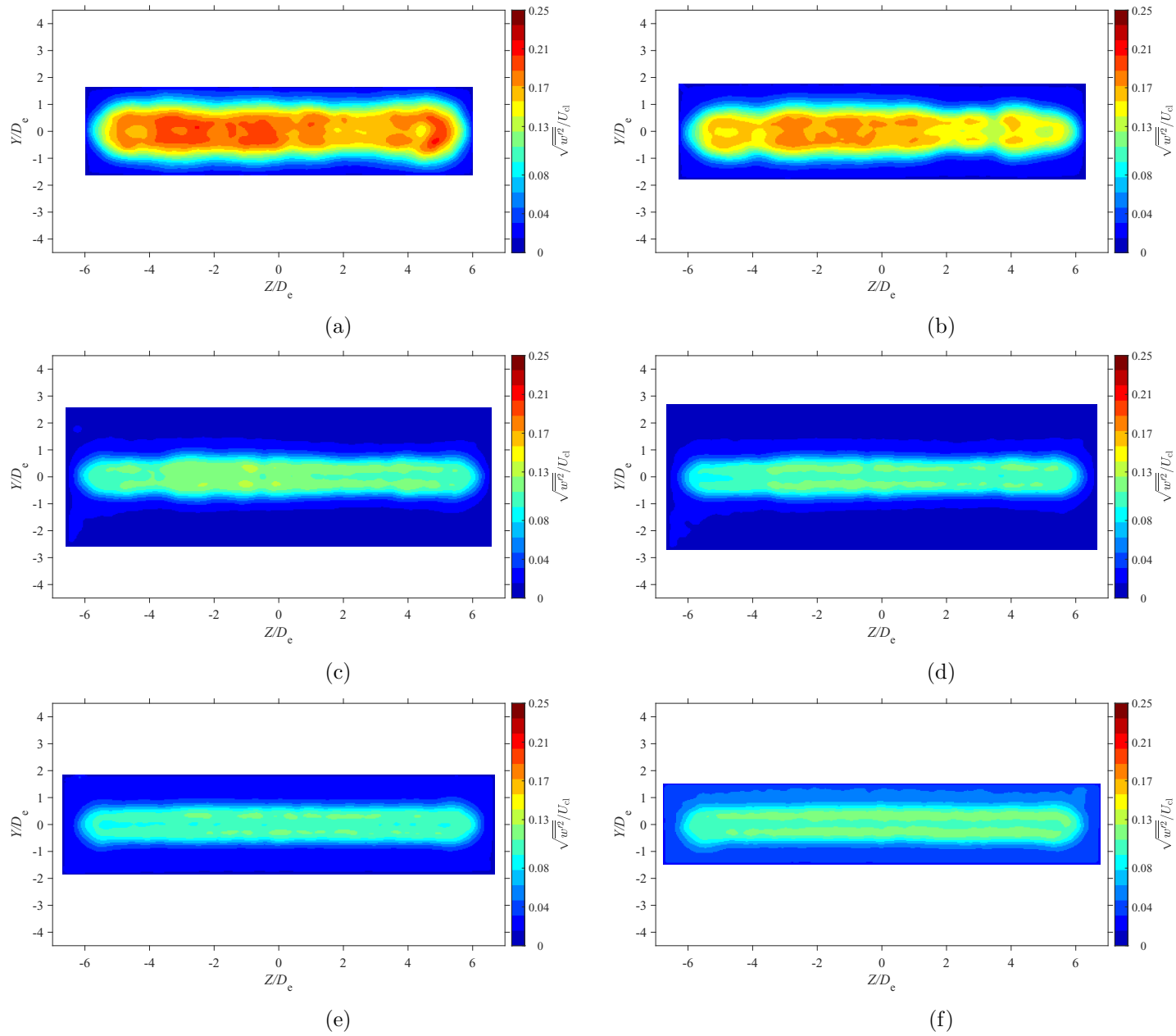


Figure E.67: Average spanwise turbulence intensity map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

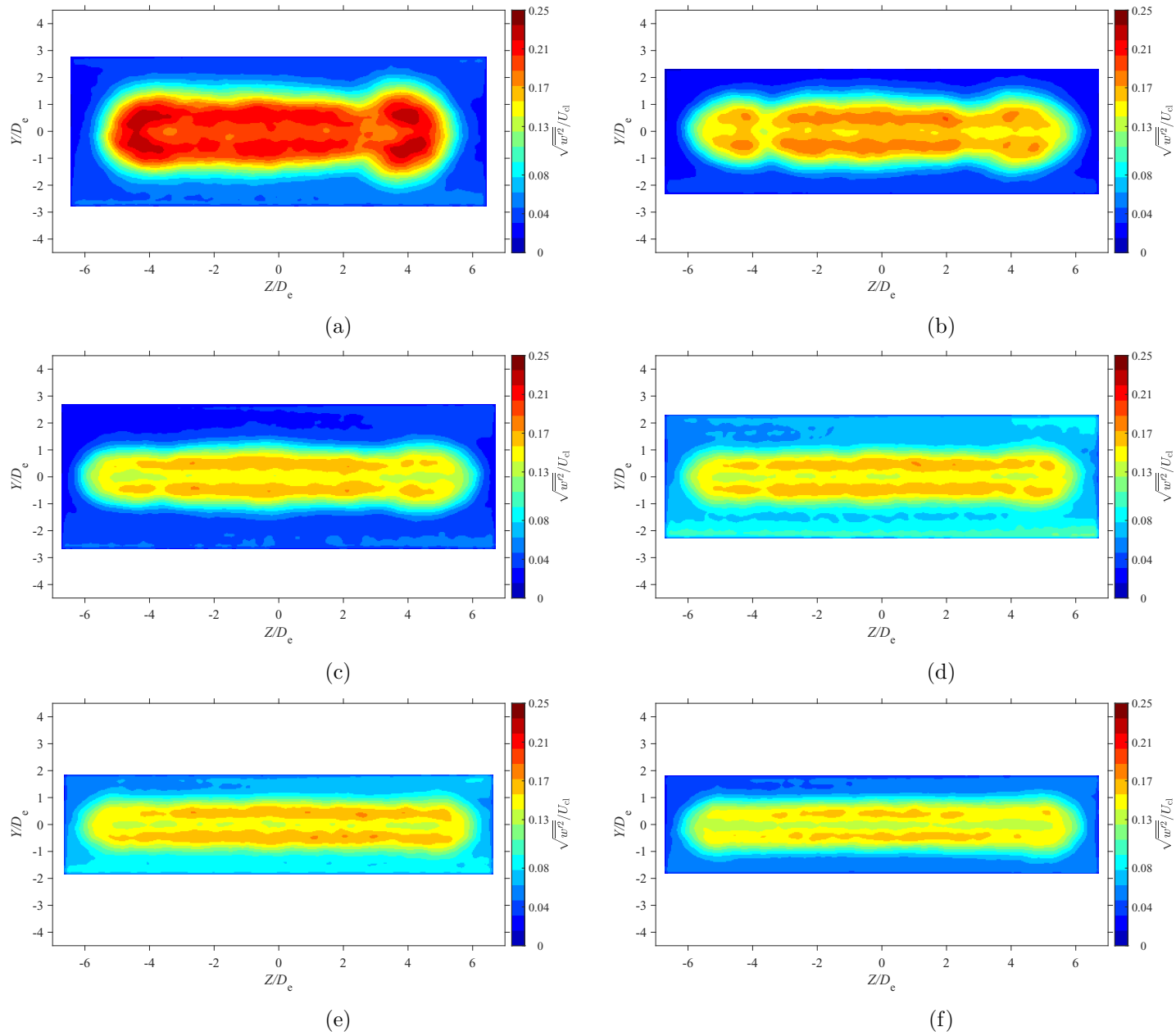


Figure E.68: Average spanwise turbulence intensity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

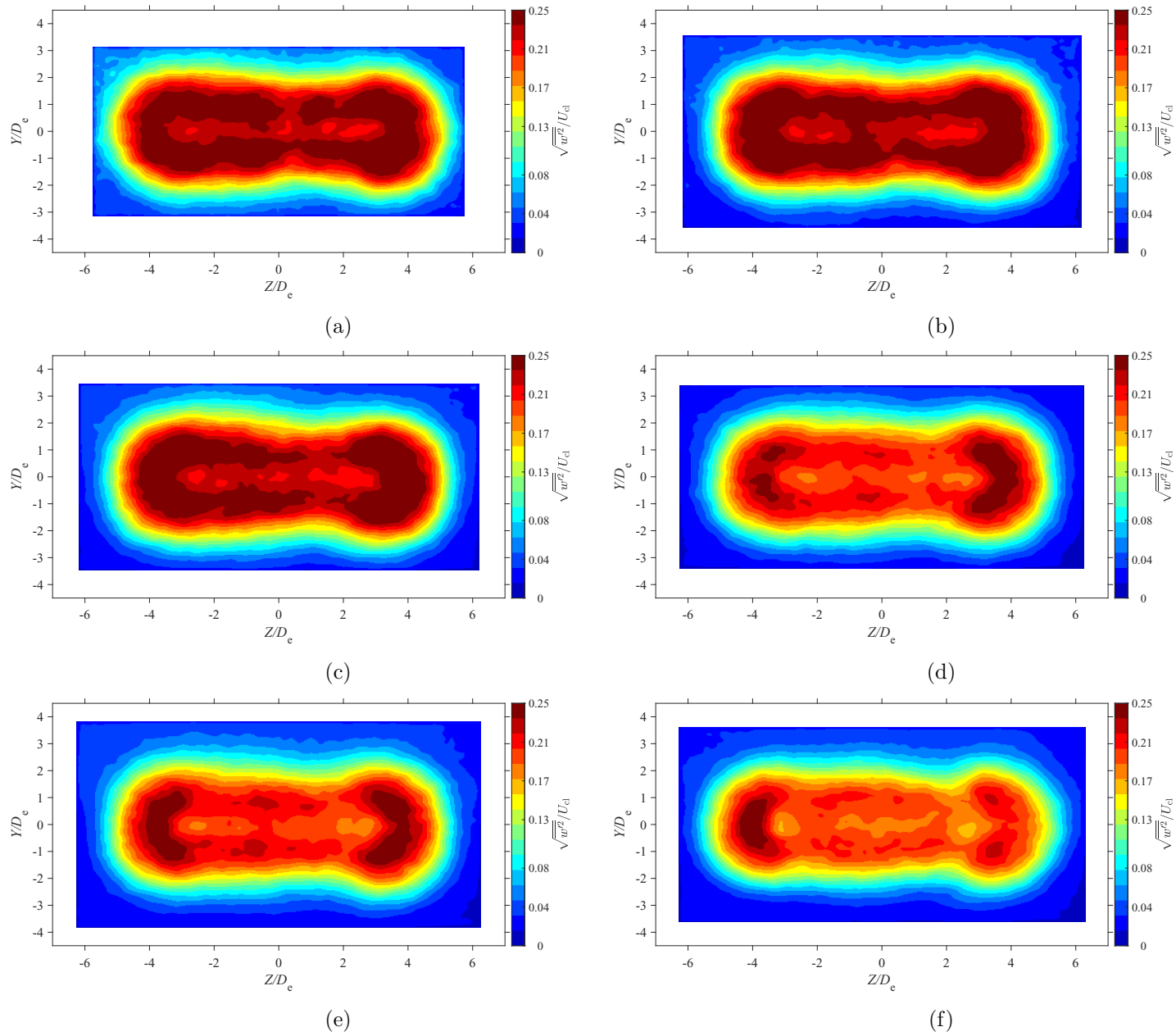


Figure E.69: Average spanwise turbulence intensity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

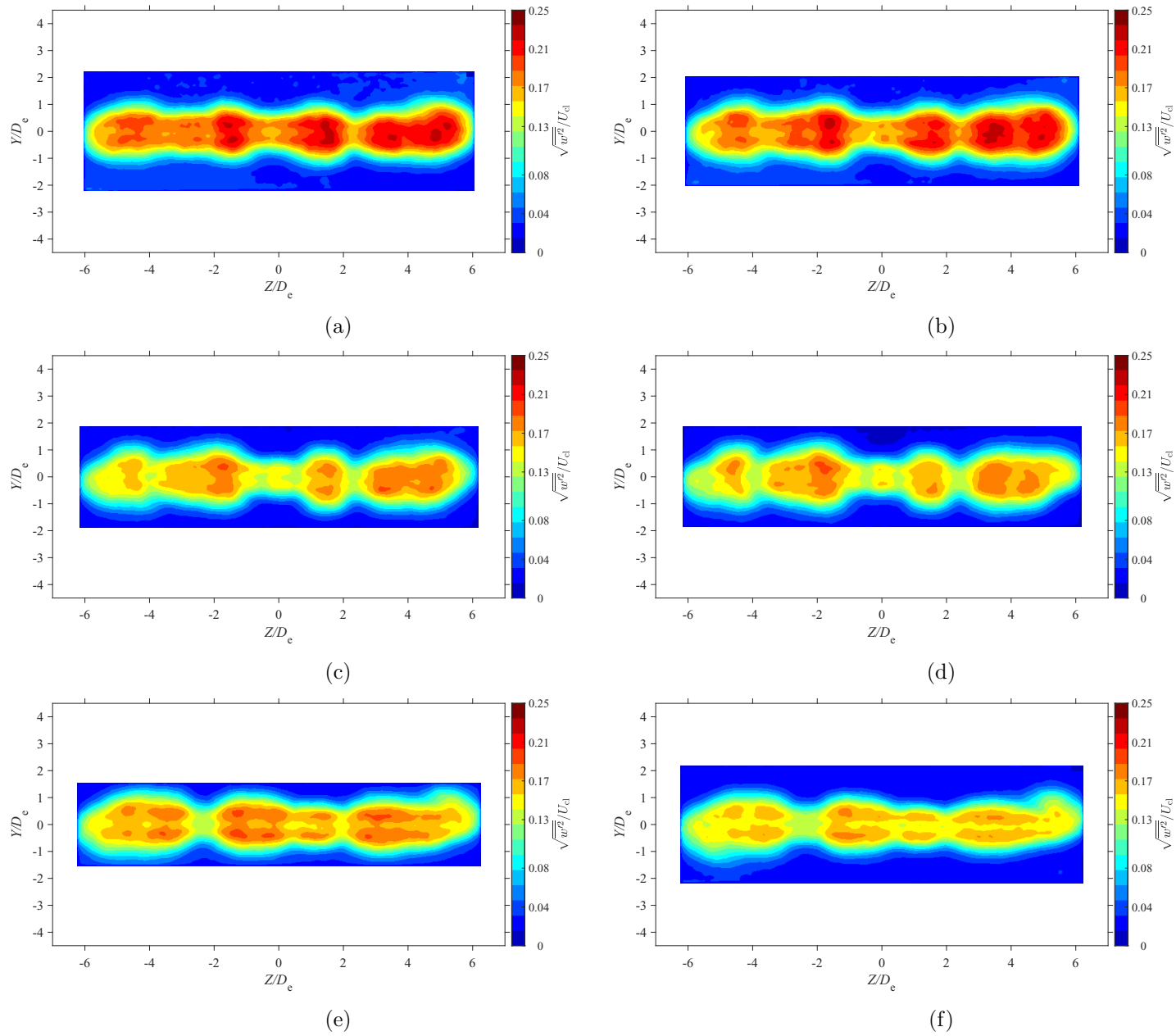


Figure E.70: Average spanwise turbulence intensity map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

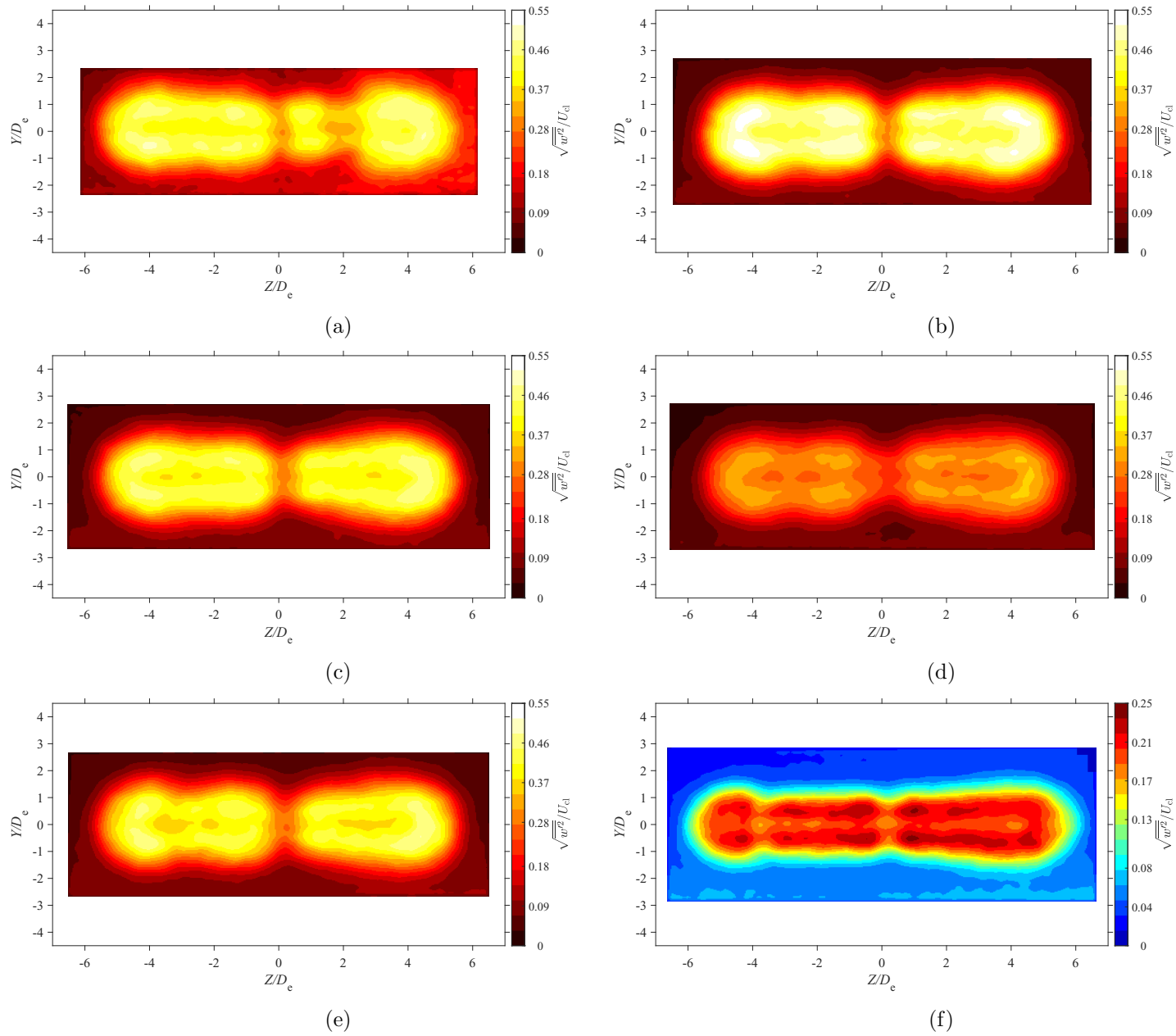


Figure E.71: Average spanwise turbulence intensity map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

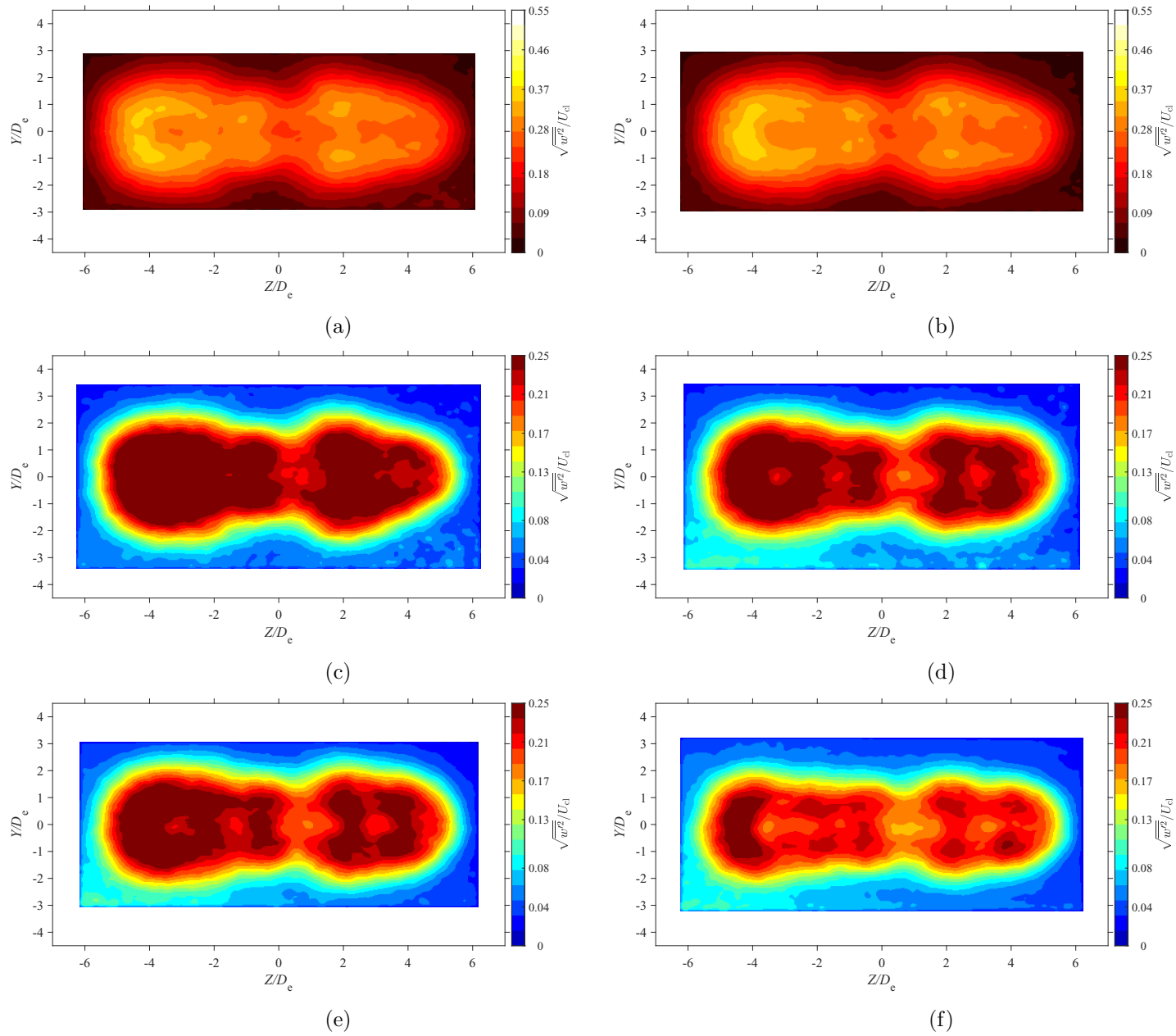


Figure E.72: Average spanwise turbulence intensity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

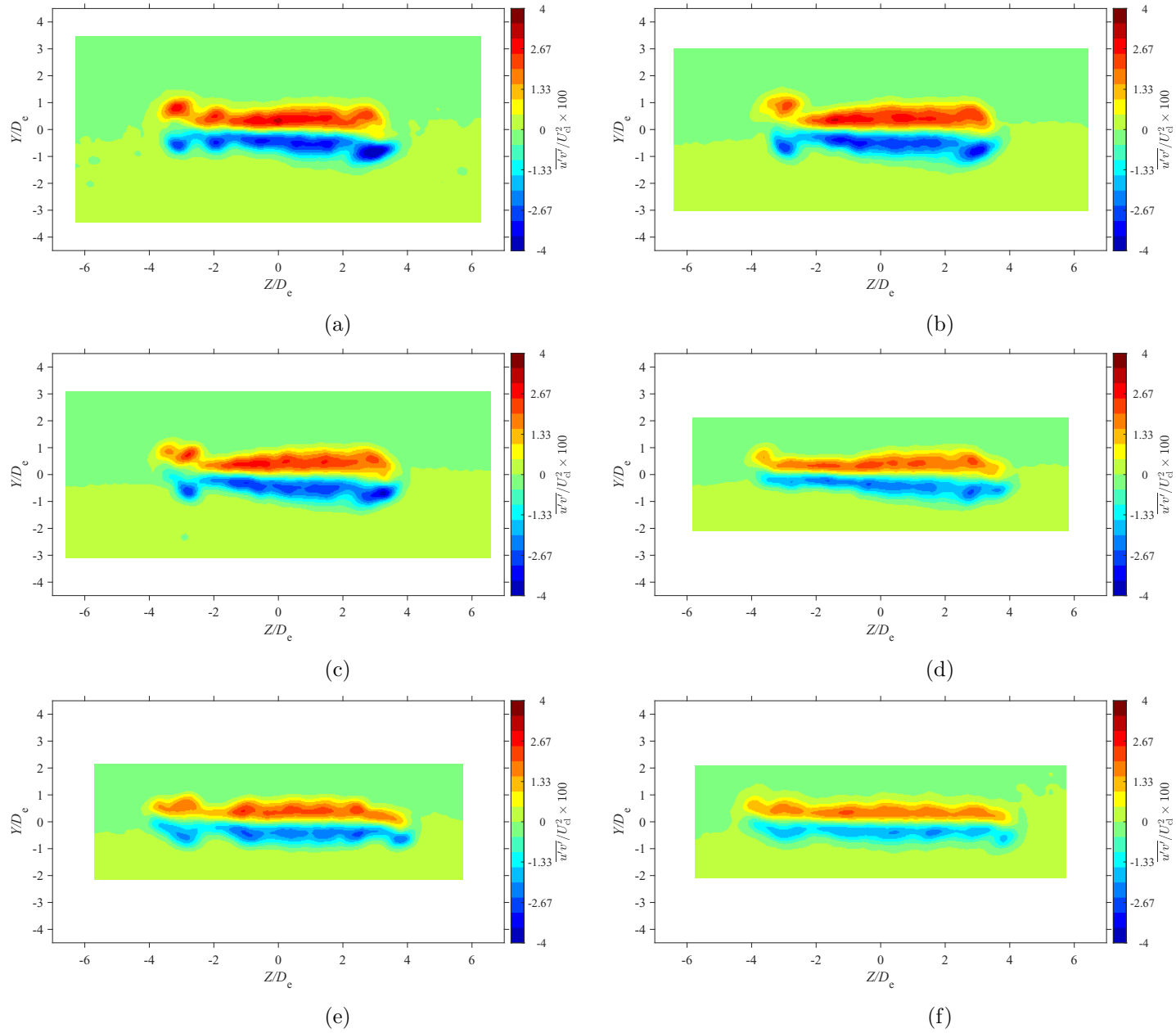


Figure E.73: Average lateral Reynolds shear stress map at location $X = 3D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

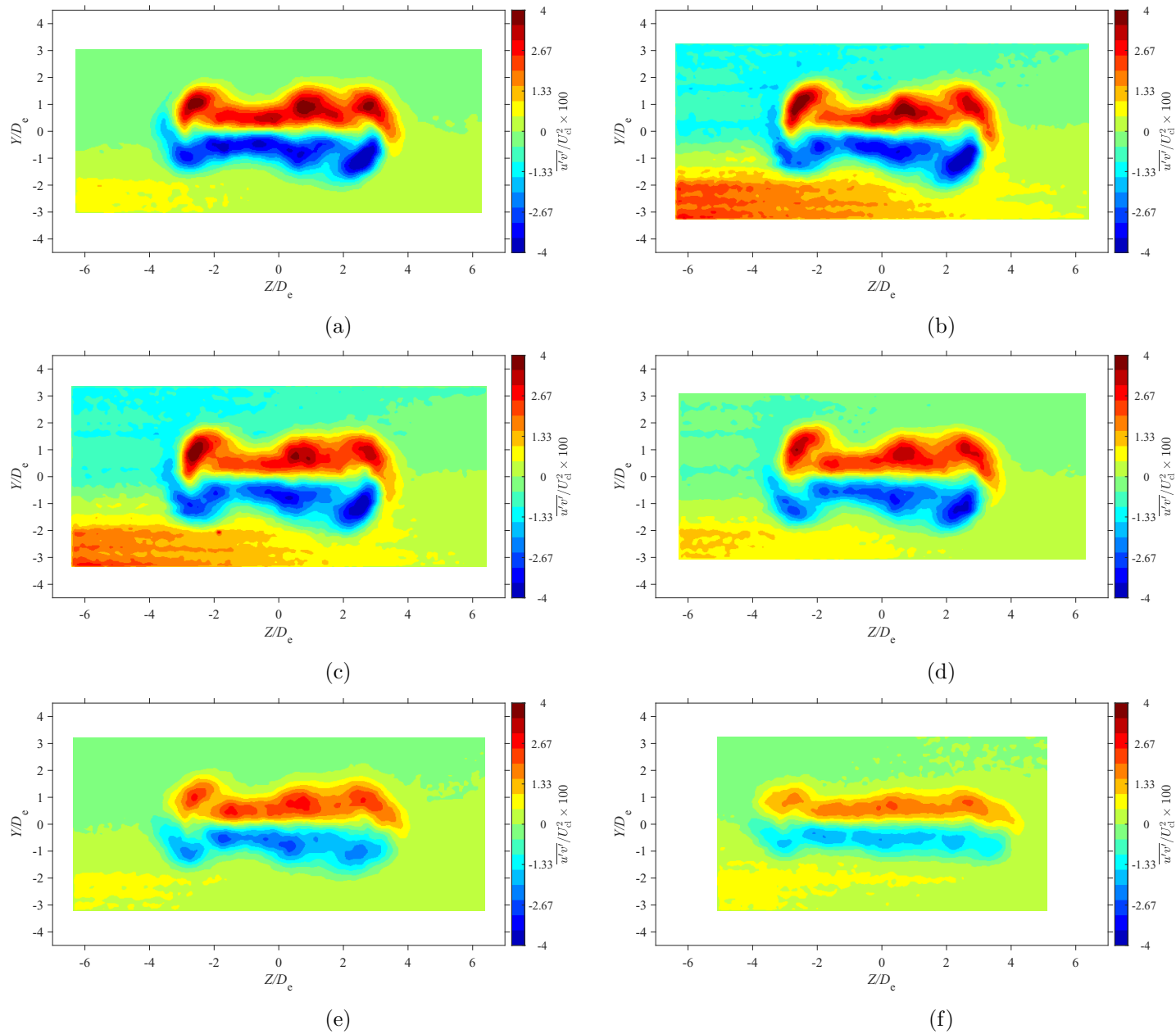


Figure E.74: Average lateral Reynolds shear stress map at location $X = 5De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

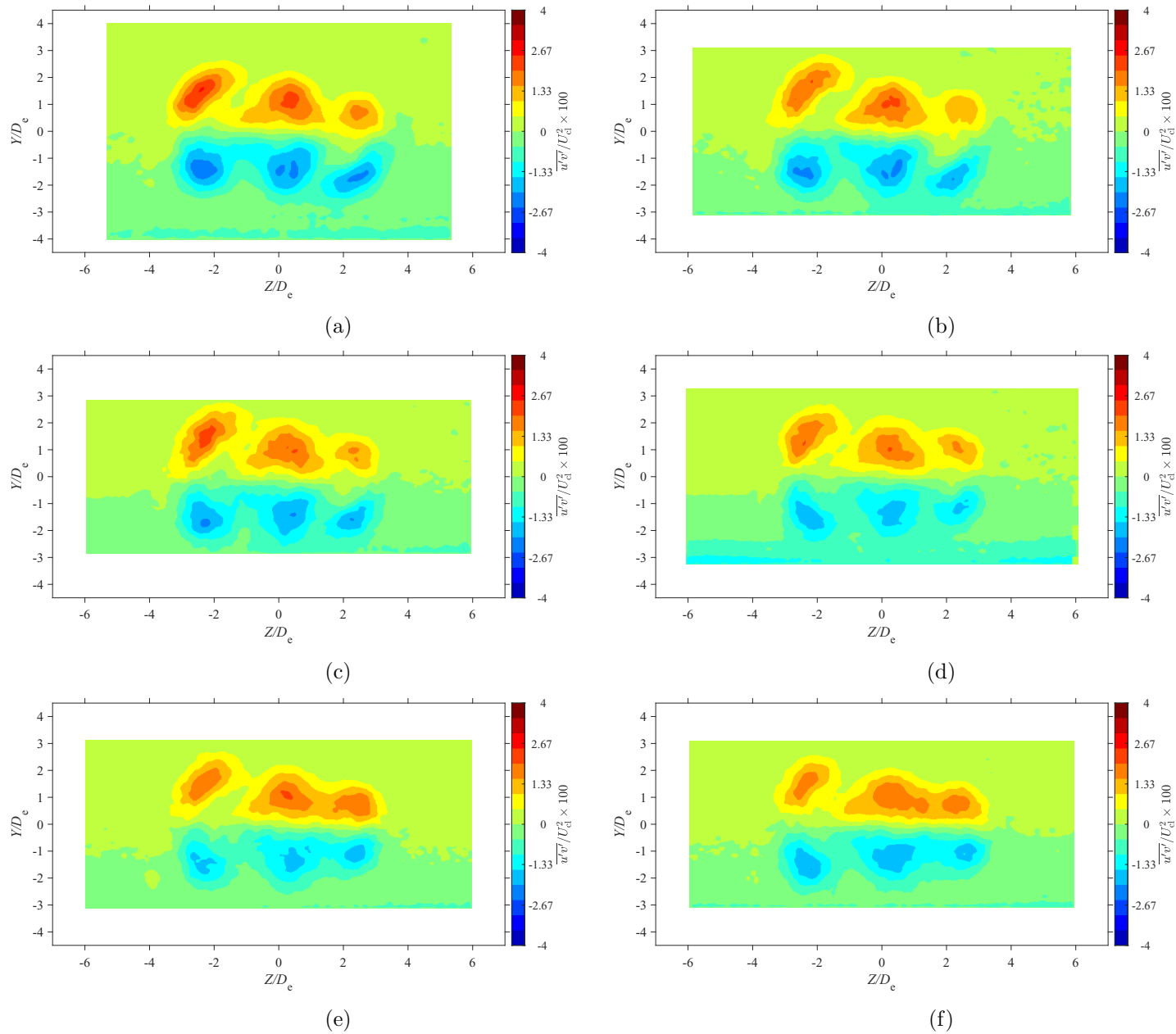
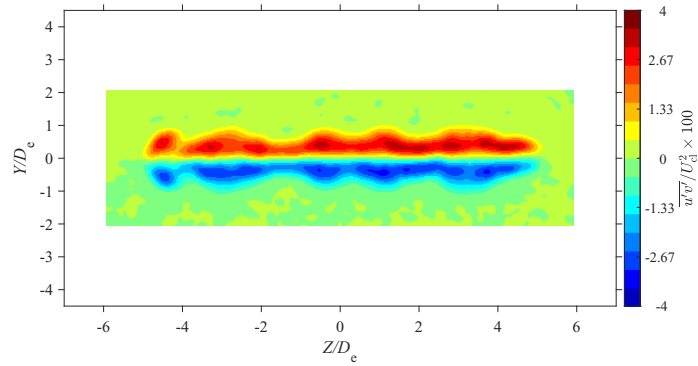
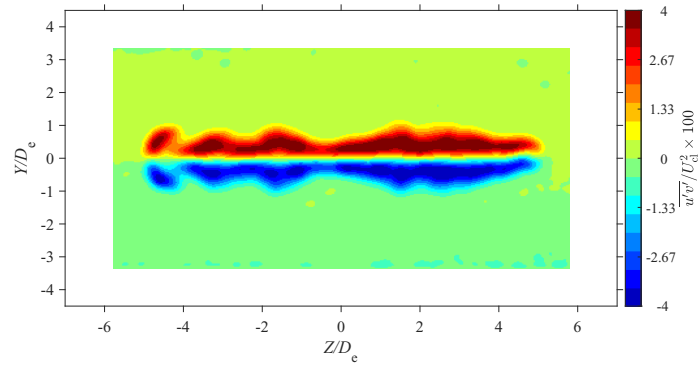


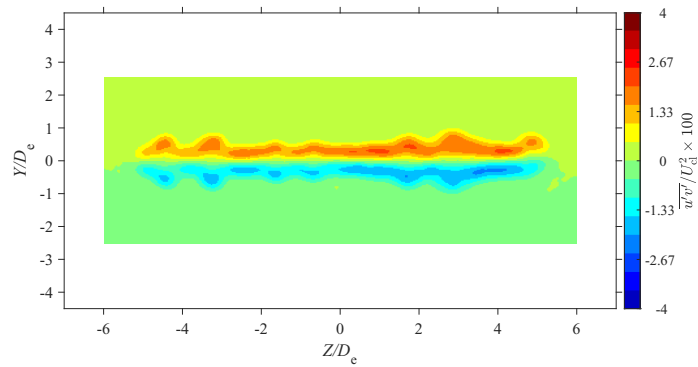
Figure E.75: Average lateral Reynolds shear stress map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



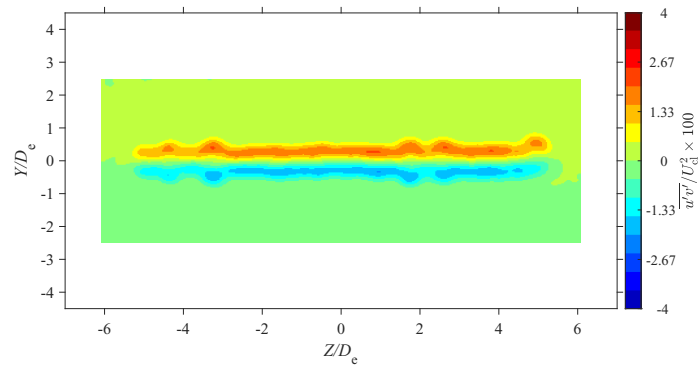
(a)



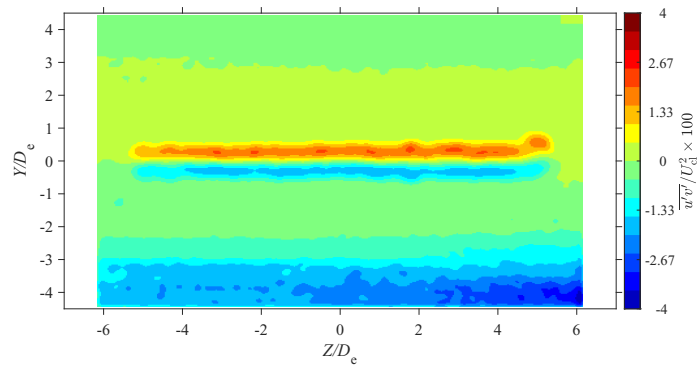
(b)



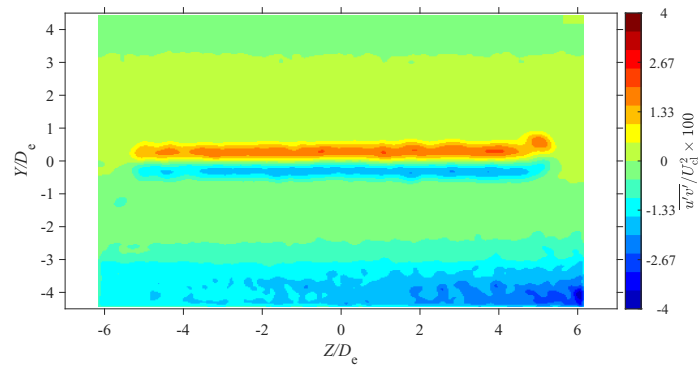
(c)



(d)



(e)



(f)

Figure E.76: Average lateral Reynolds shear stress map at location $X = 3De$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

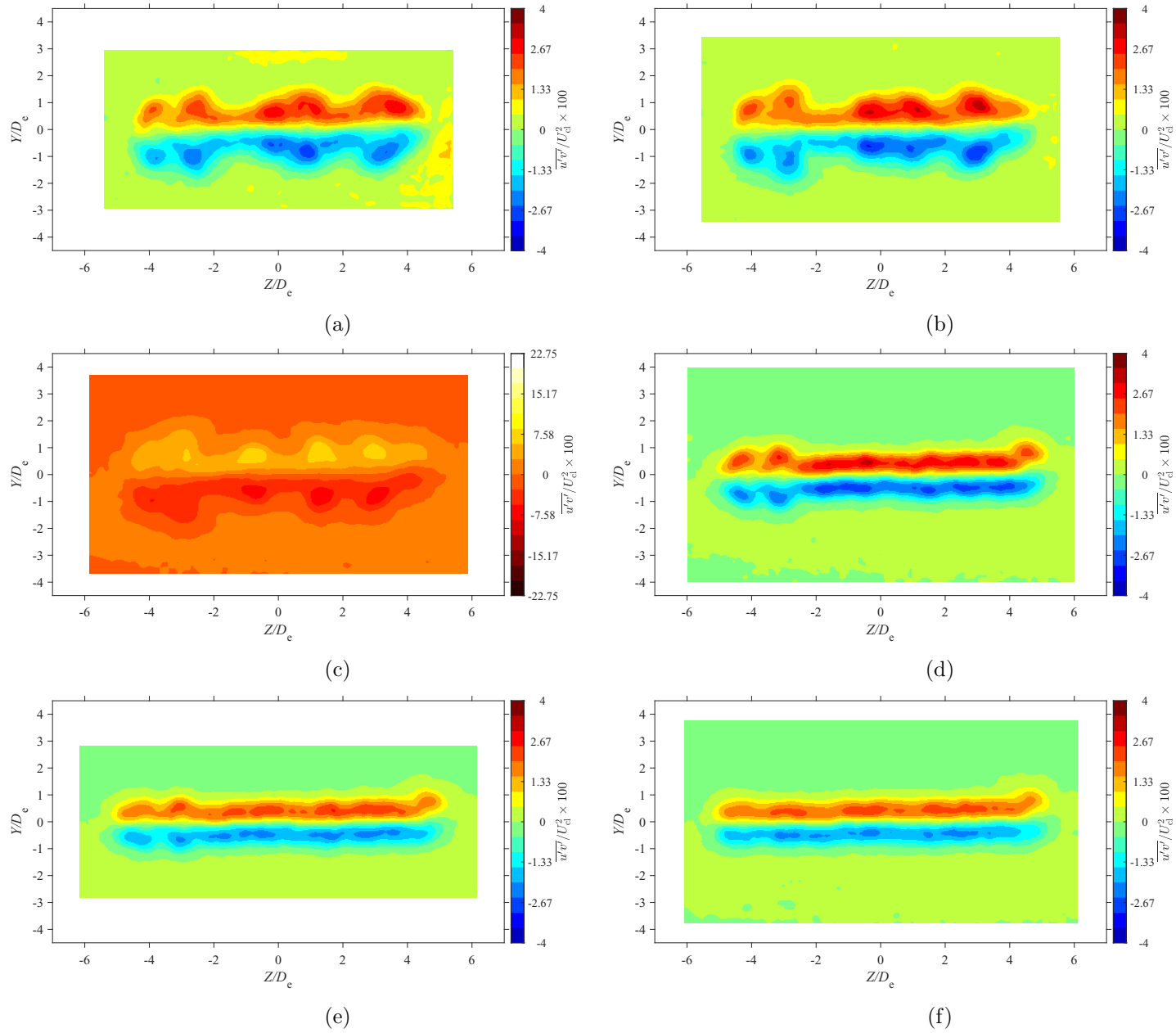
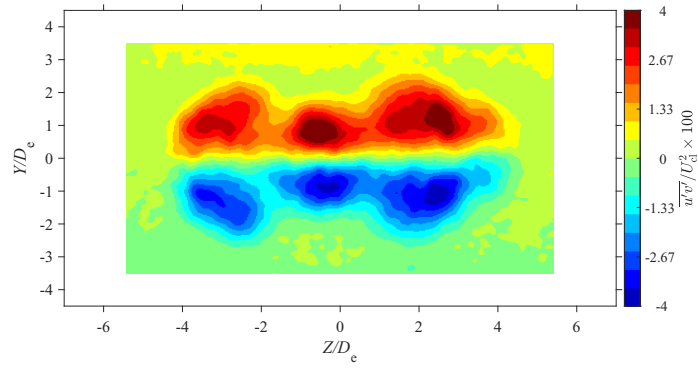
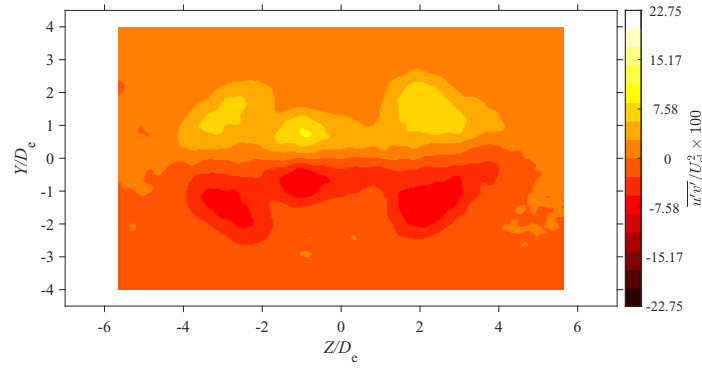


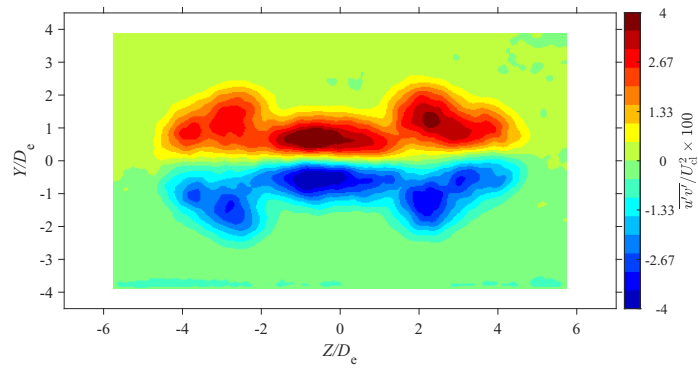
Figure E.77: Average lateral Reynolds shear stress map at location $X = 5De$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



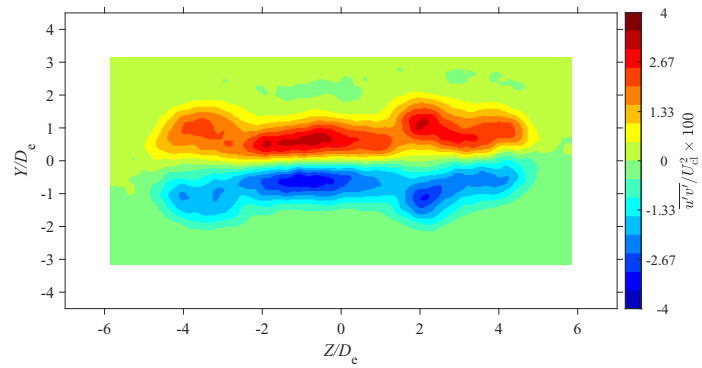
(a)



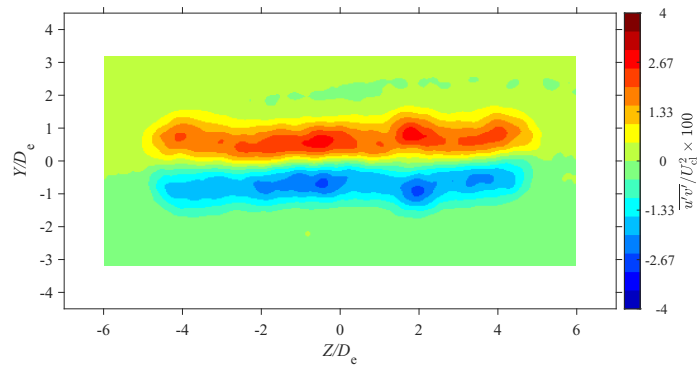
(b)



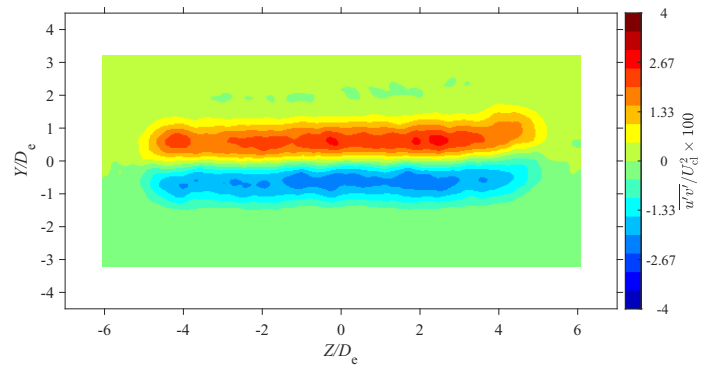
(c)



(d)



(e)



(f)

Figure E.78: Average lateral Reynolds shear stress map at location $X = 7De$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

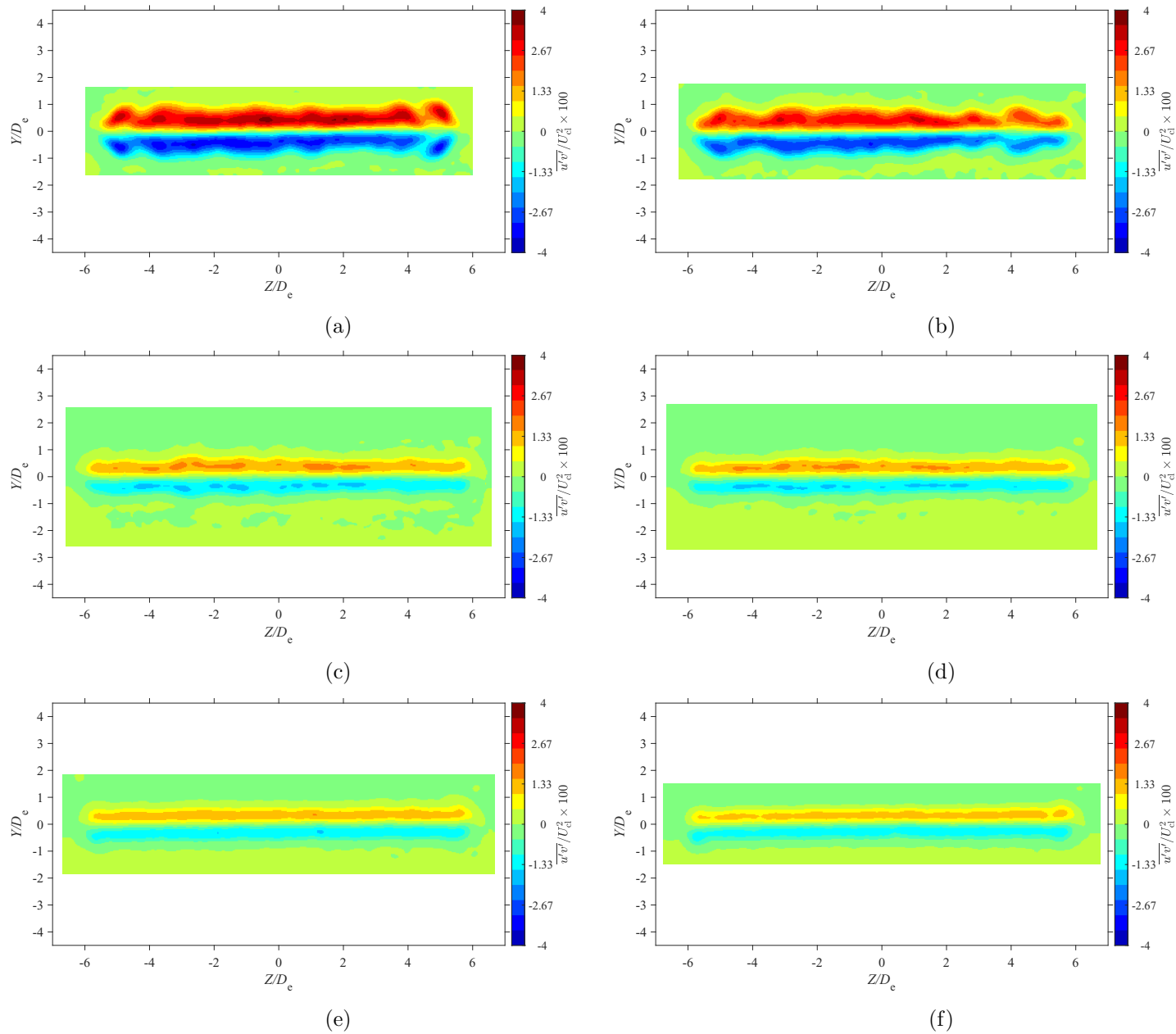
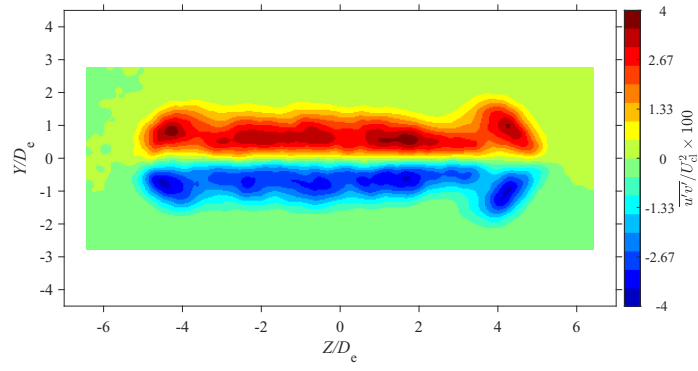
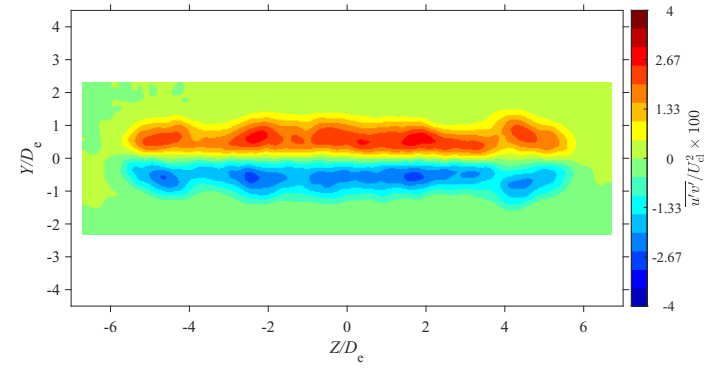


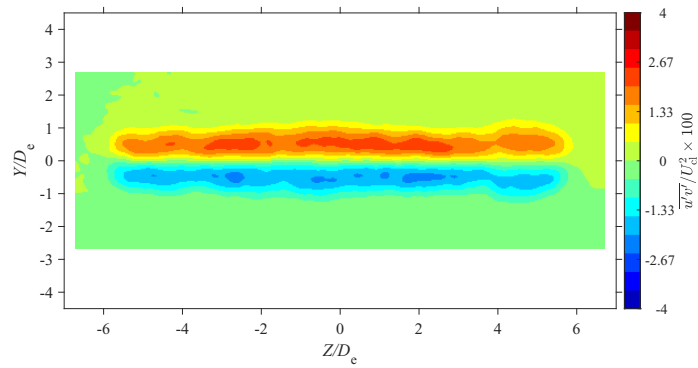
Figure E.79: Average lateral Reynolds shear stress map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



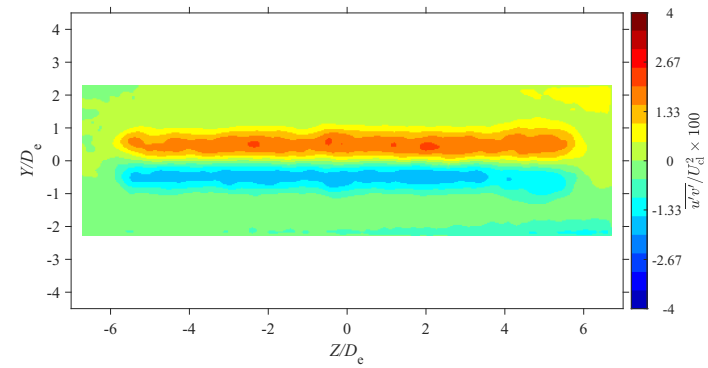
(a)



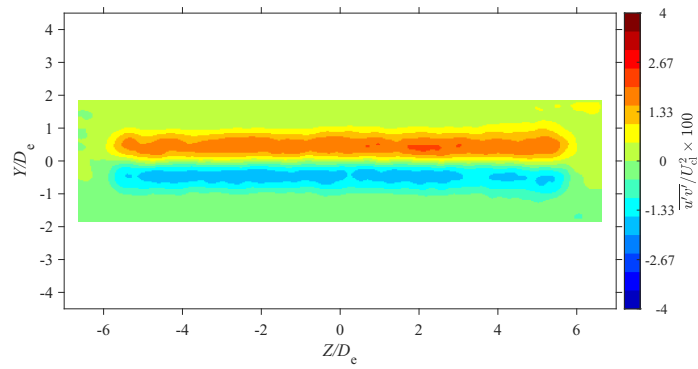
(b)



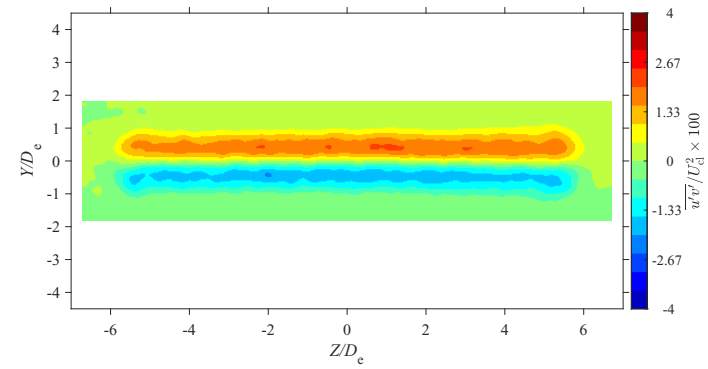
(c)



(d)



(e)



(f)

Figure E.80: Average lateral Reynolds shear stress map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

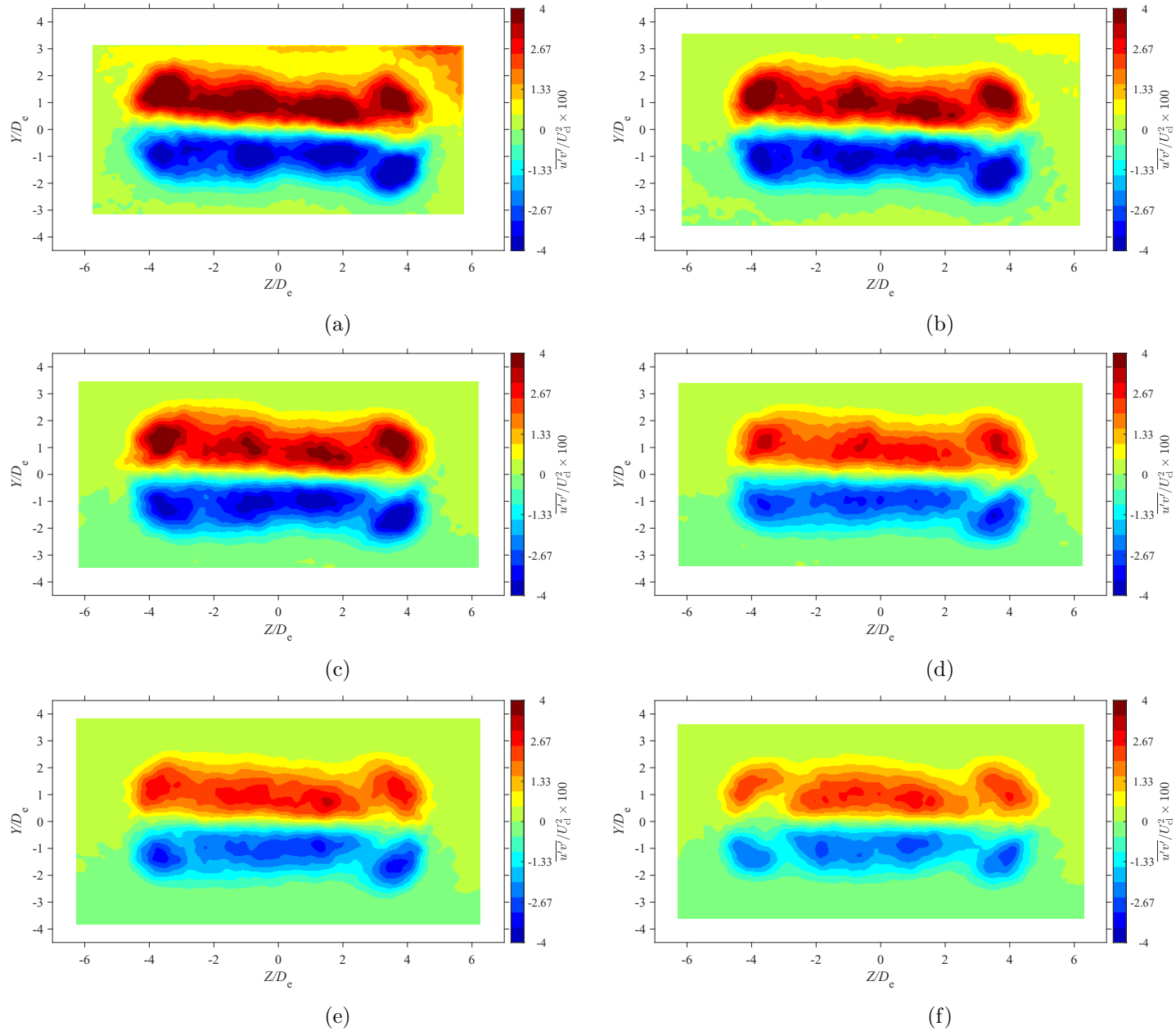


Figure E.81: Average lateral Reynolds shear stress map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

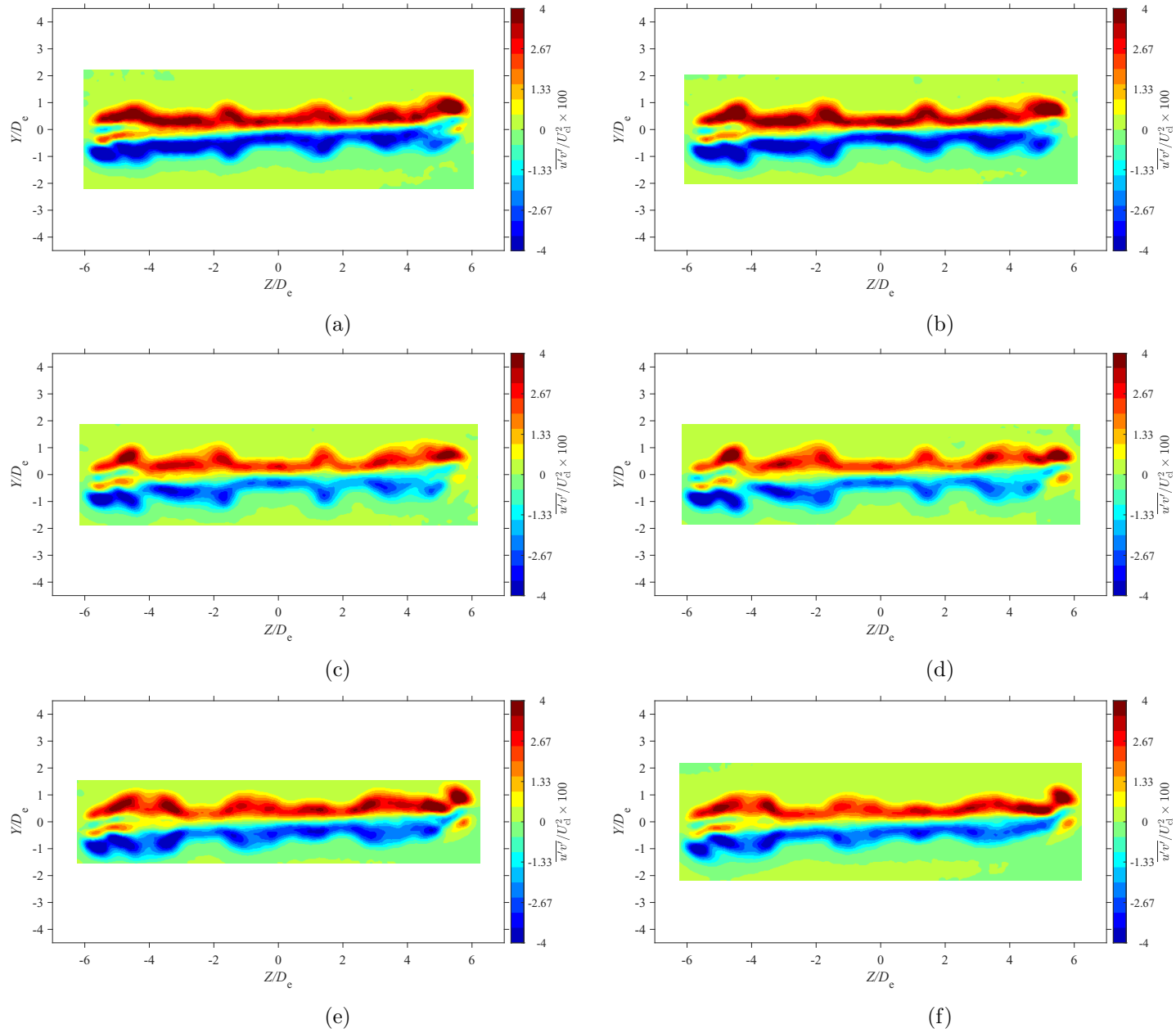


Figure E.82: Average lateral Reynolds shear stress map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

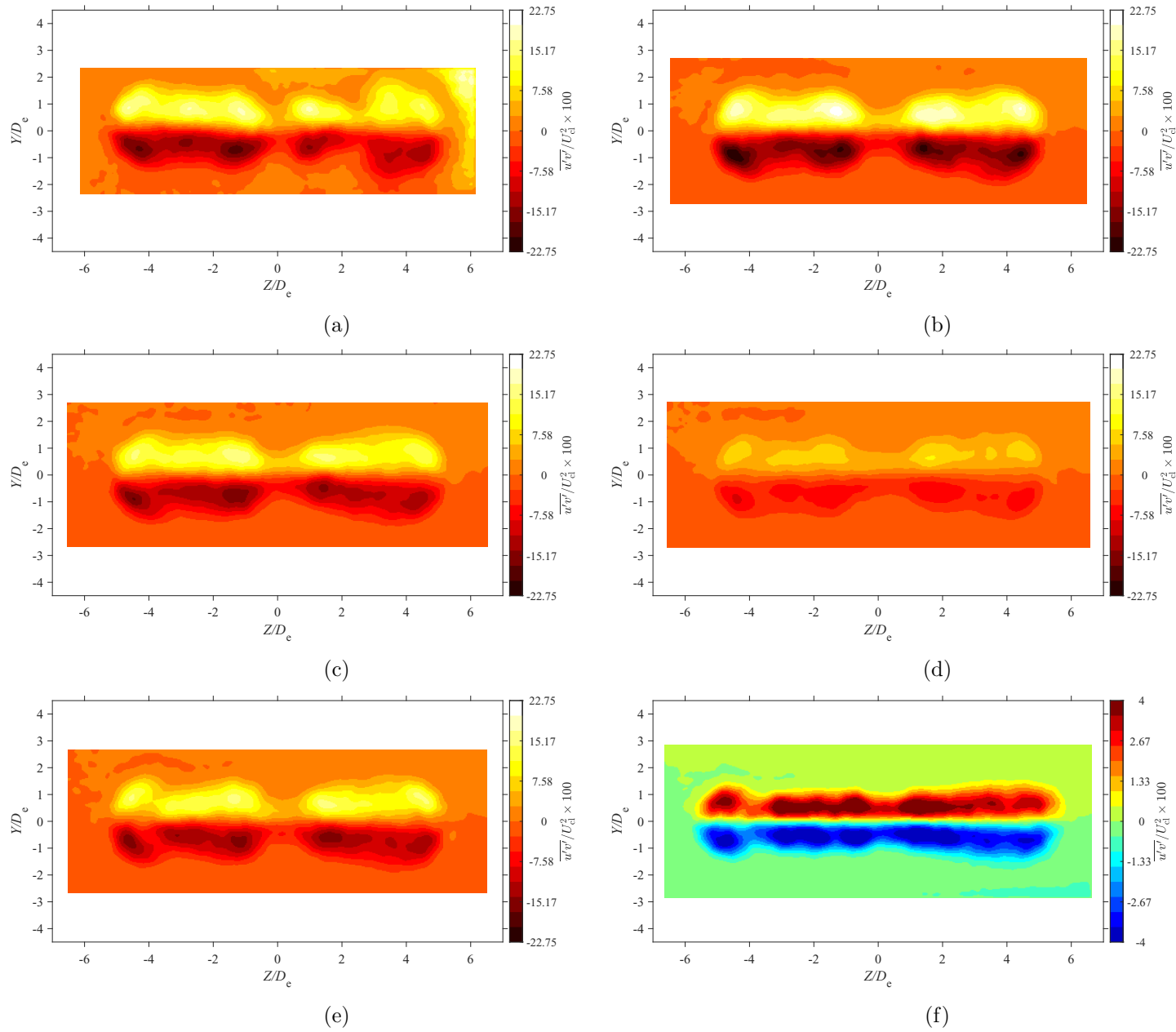


Figure E.83: Average lateral Reynolds shear stress map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

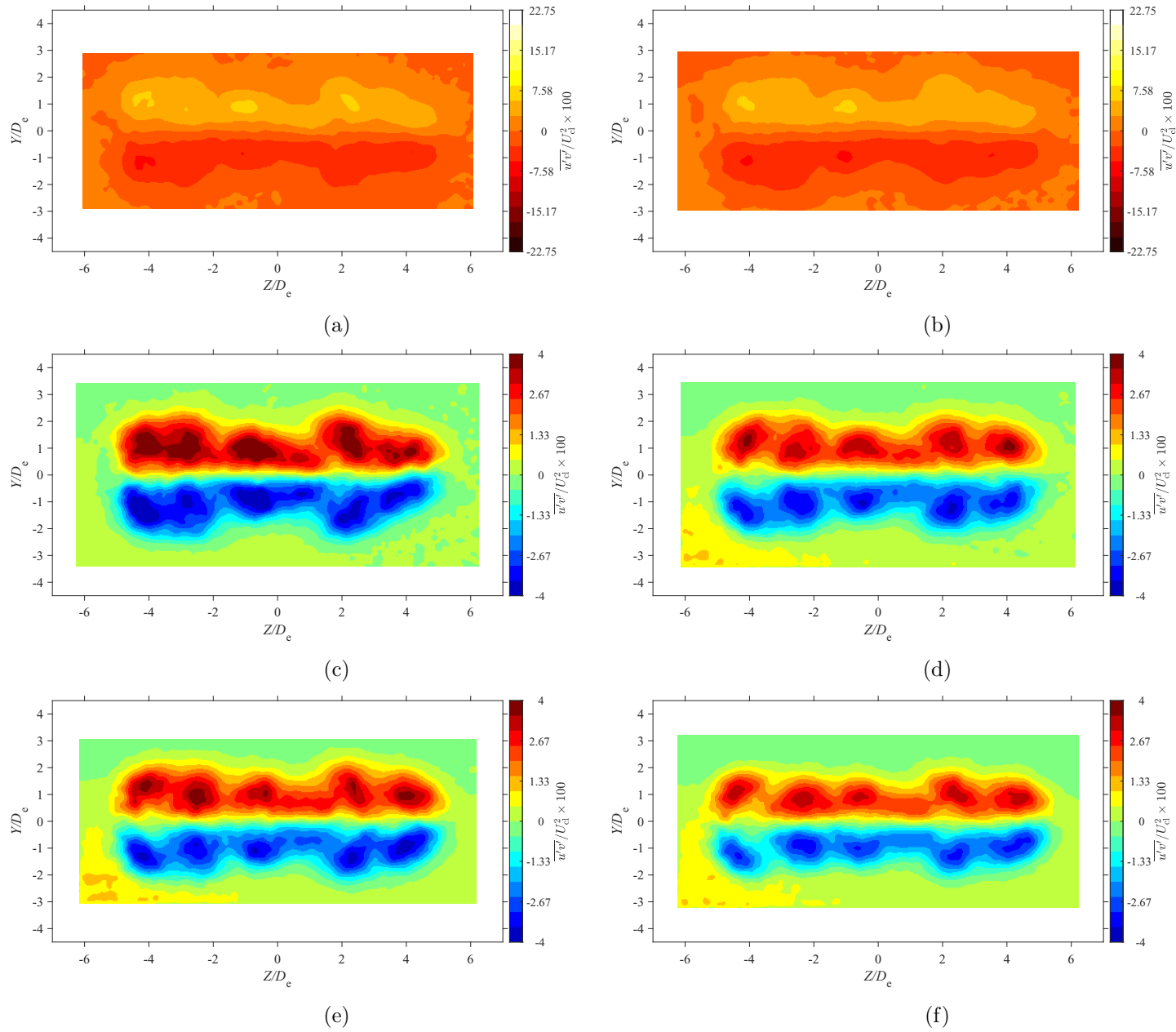
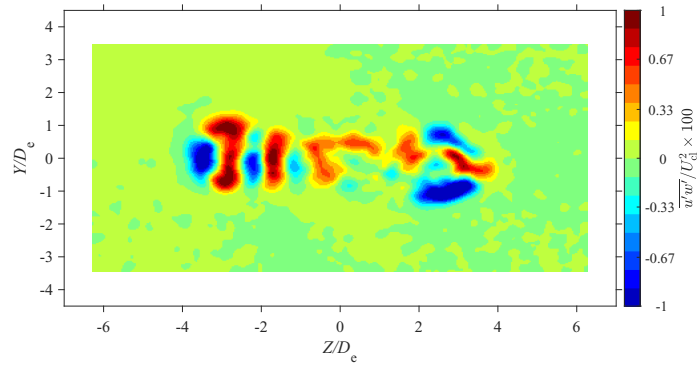
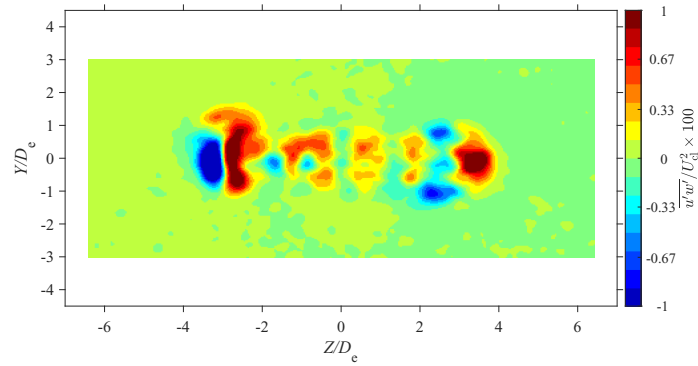


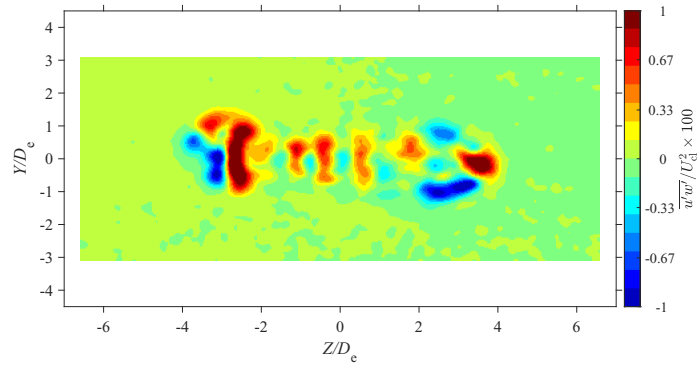
Figure E.84: Average lateral Reynolds shear stress map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



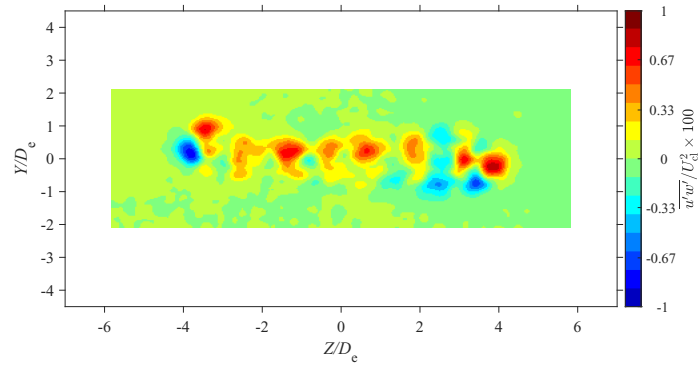
(a)



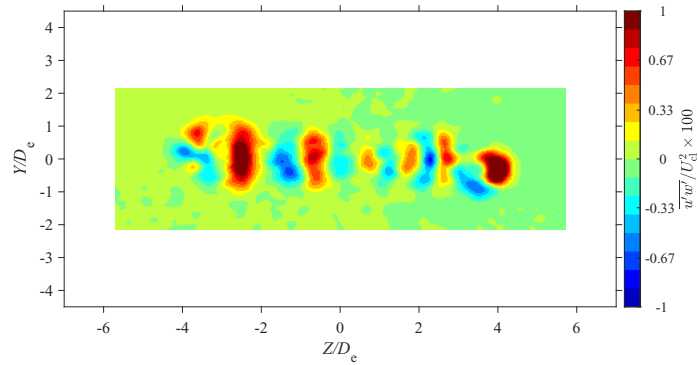
(b)



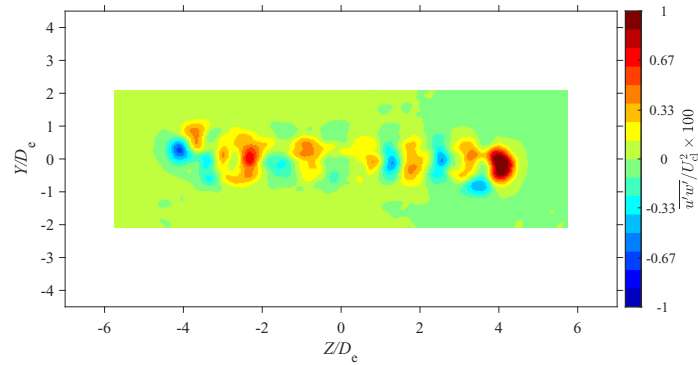
(c)



(d)



(e)



(f)

Figure E.85: Average spanwise Reynolds shear stress map at location $X = 3De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

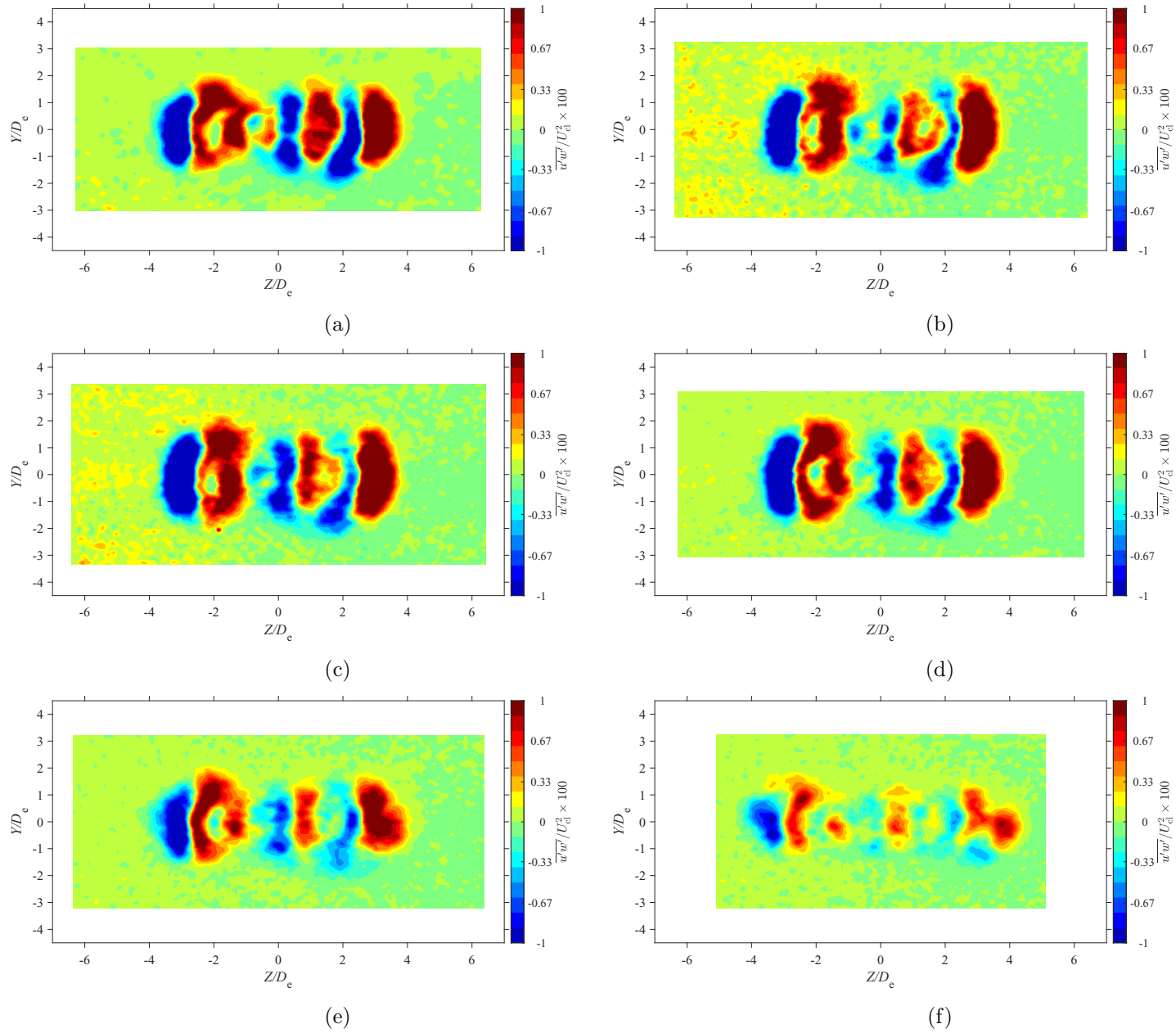


Figure E.86: Average spanwise Reynolds shear stress map at location $X = 5De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

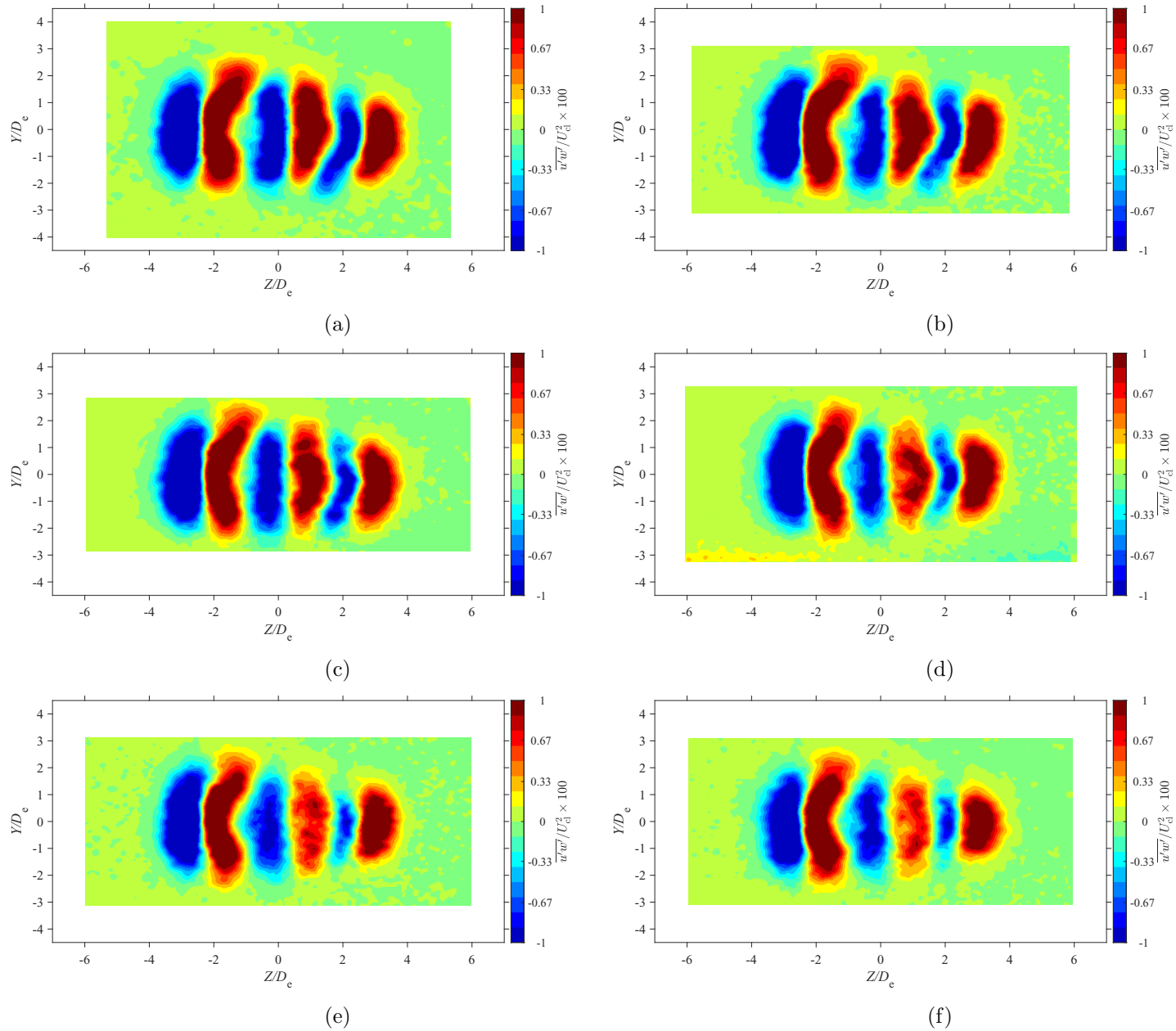


Figure E.87: Average spanwise Reynolds shear stress map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

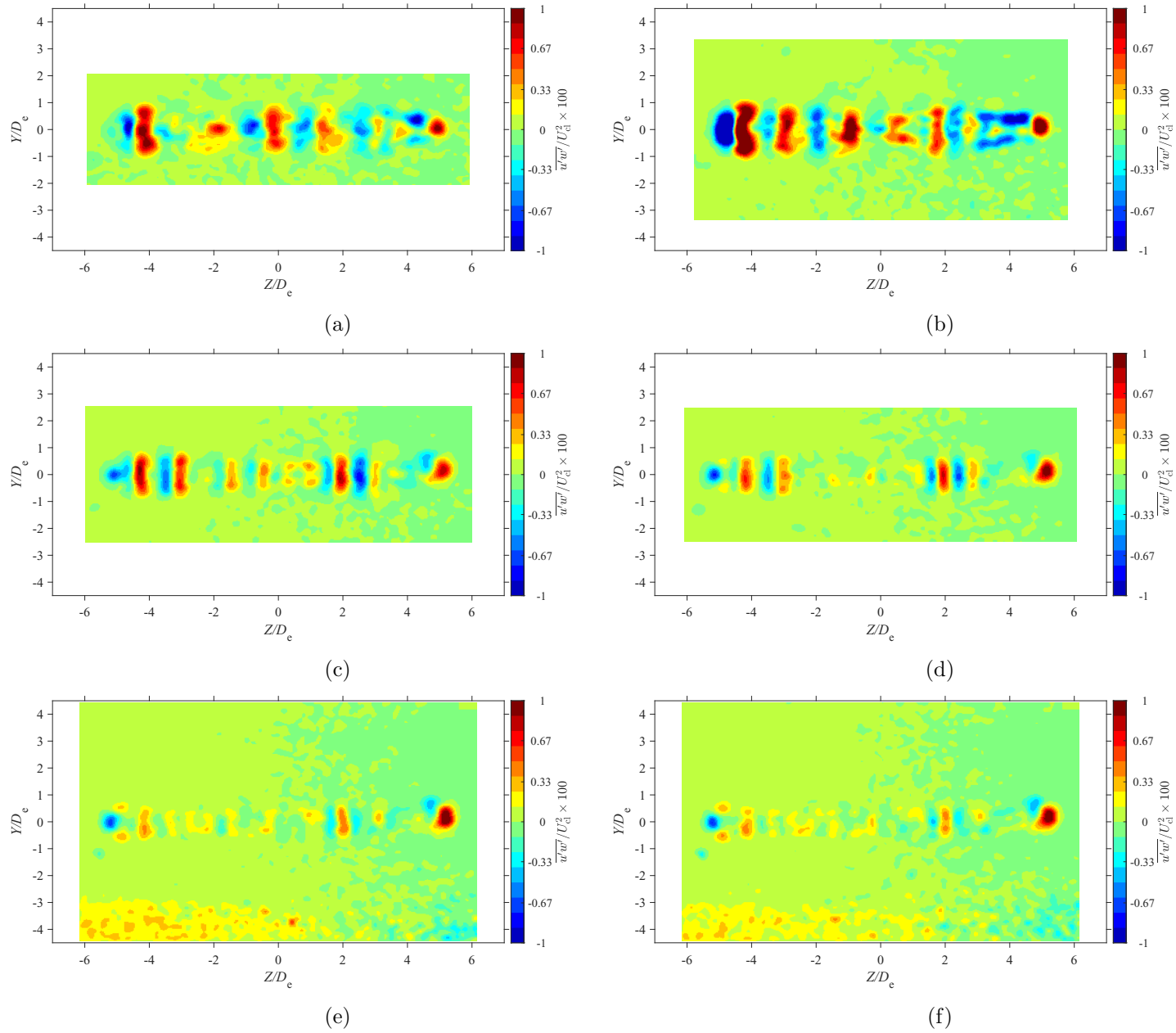


Figure E.88: Average spanwise Reynolds shear stress map at location $X = 3De$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

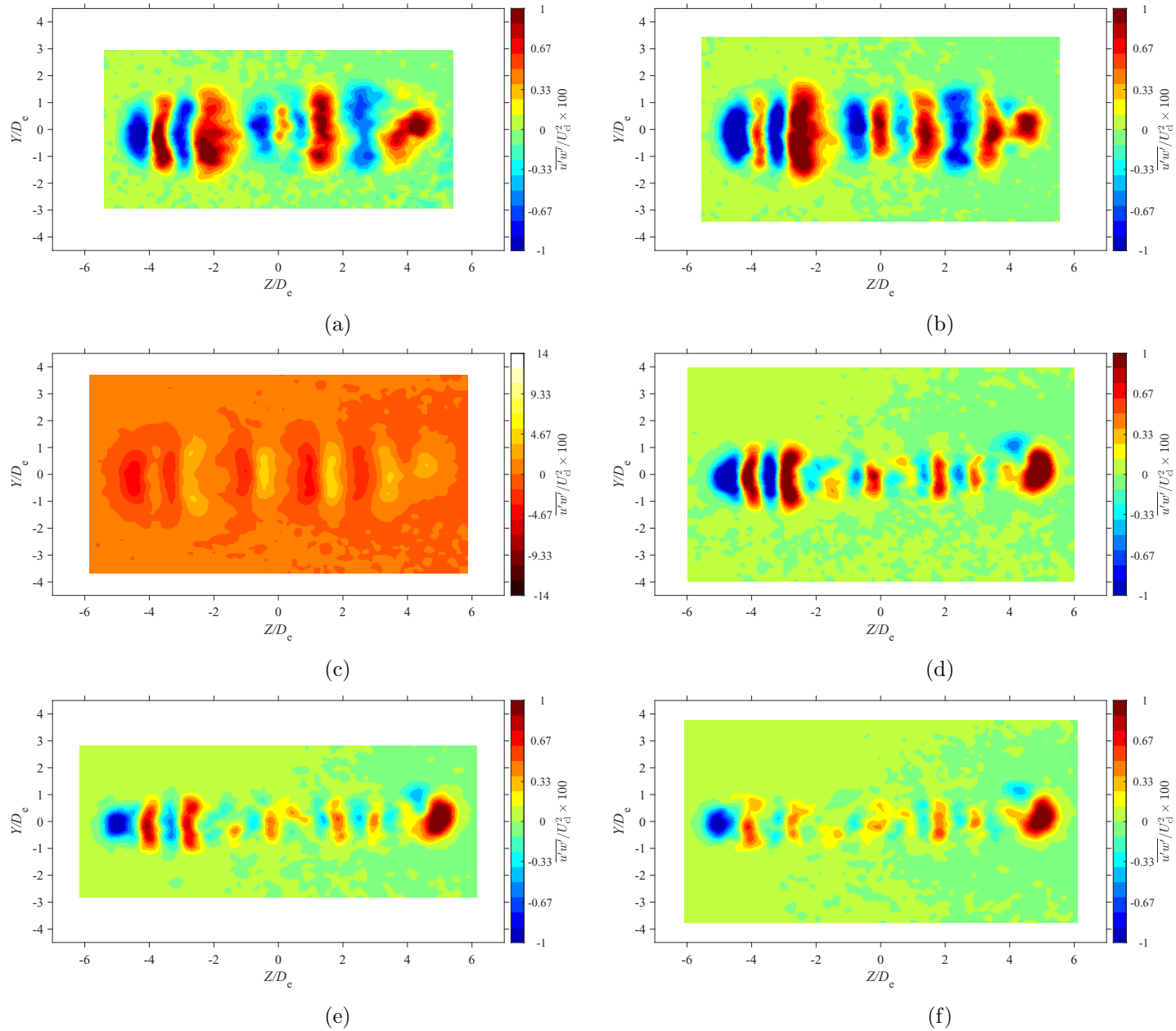


Figure E.89: Average spanwise Reynolds shear stress map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

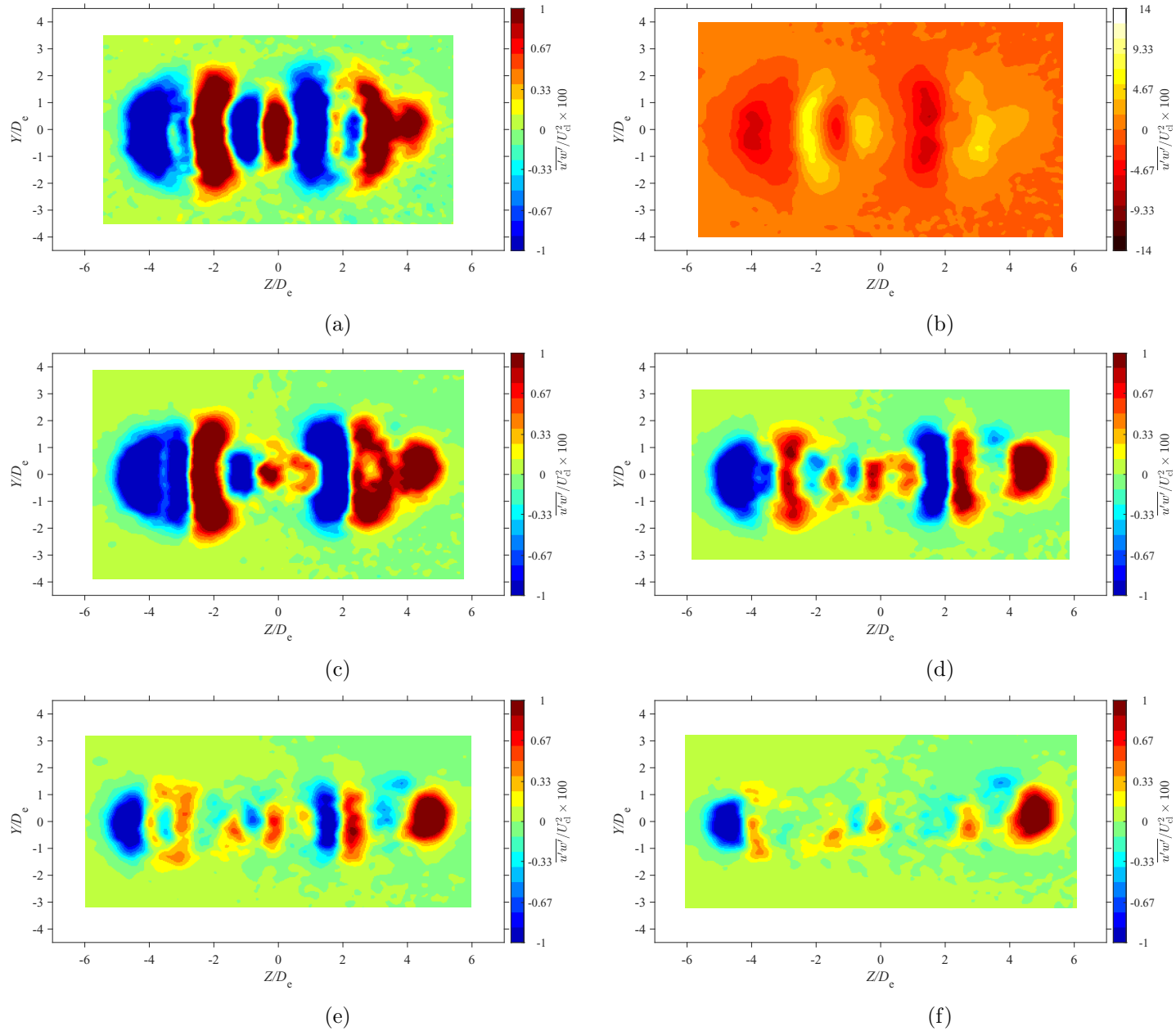


Figure E.90: Average spanwise Reynolds shear stress map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

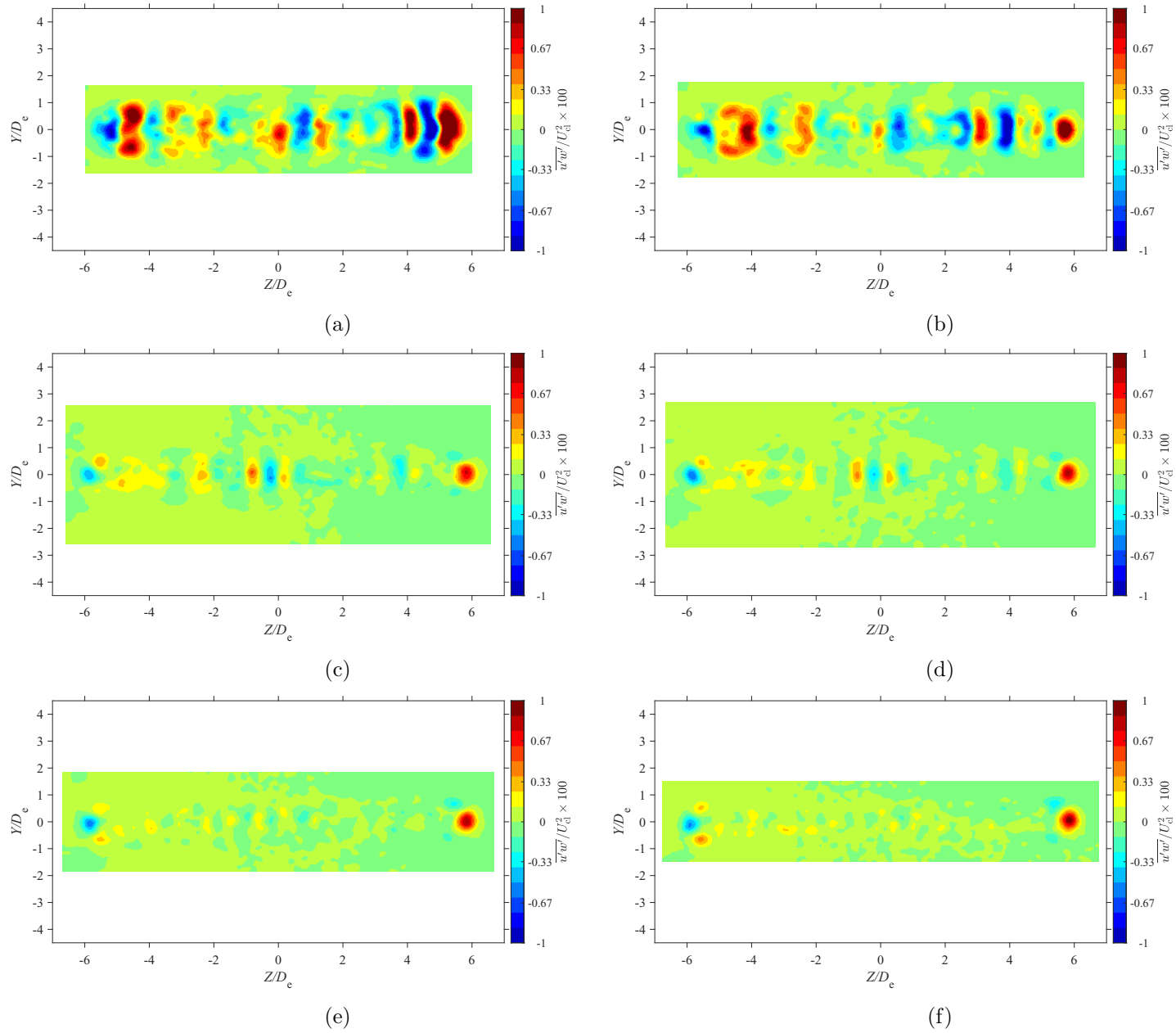


Figure E.91: Average spanwise Reynolds shear stress map at location $X = 3D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

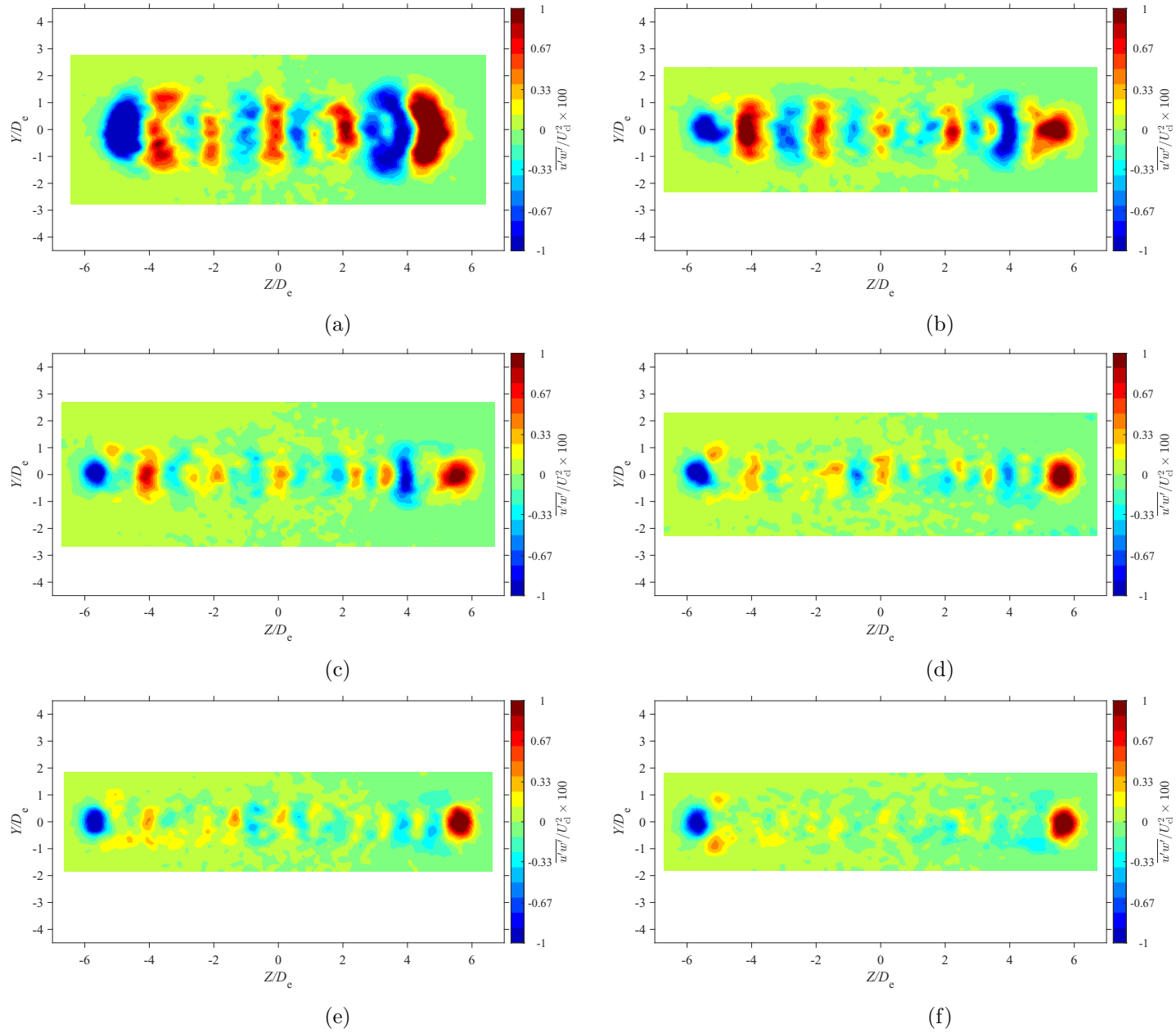


Figure E.92: Average spanwise Reynolds shear stress map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

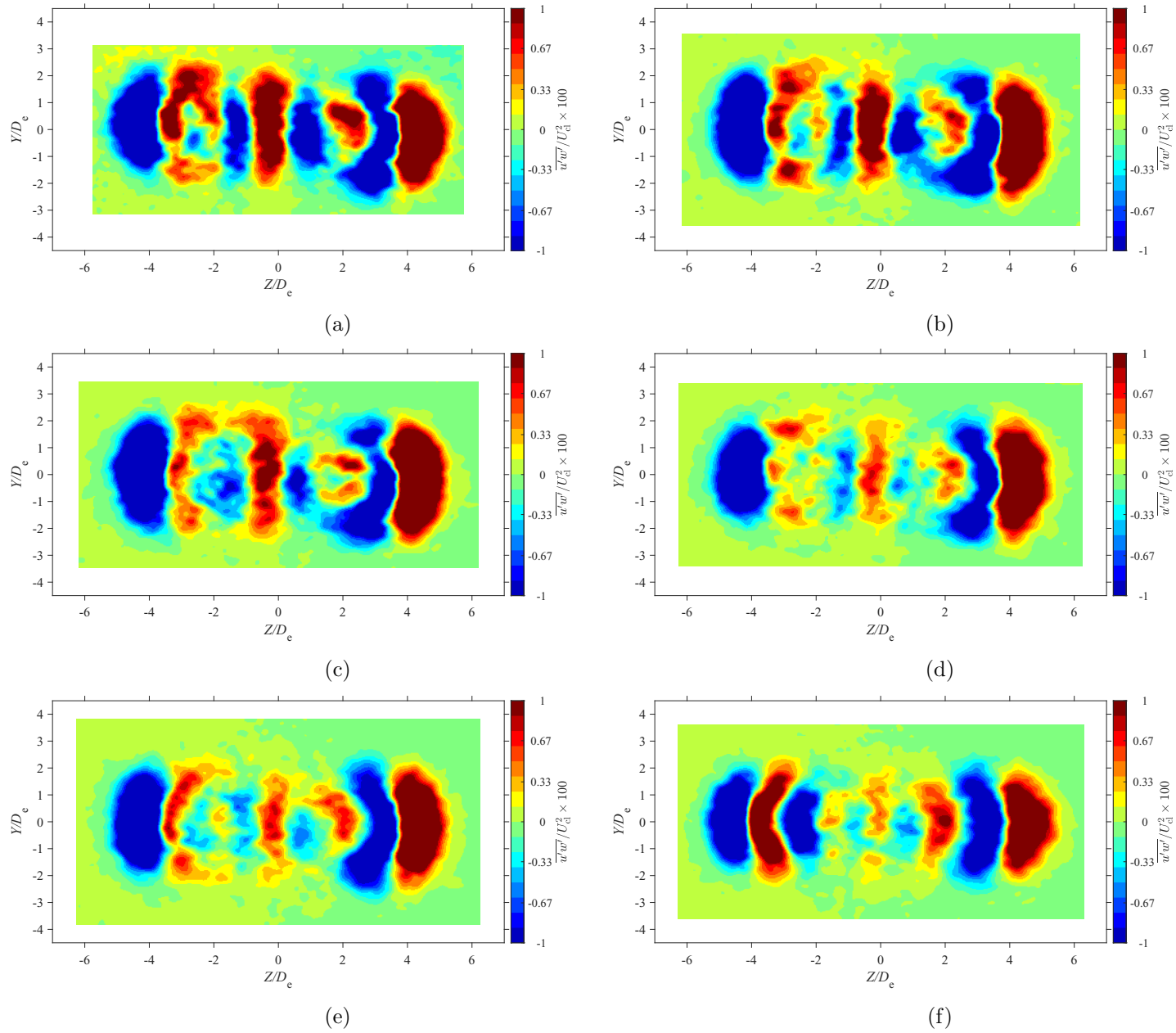


Figure E.93: Average spanwise Reynolds shear stress map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

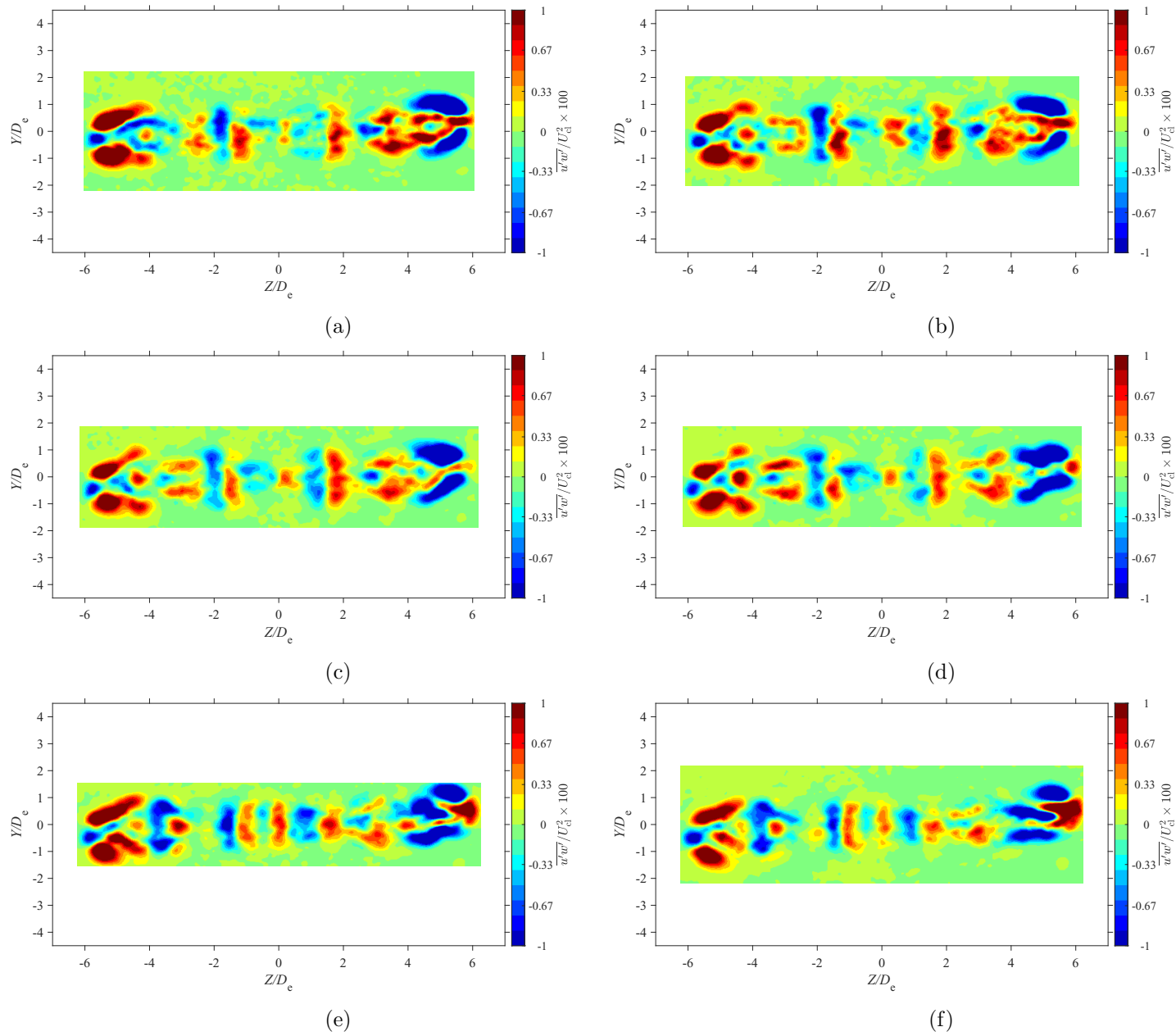


Figure E.94: Average spanwise Reynolds shear stress map at location $X = 3D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

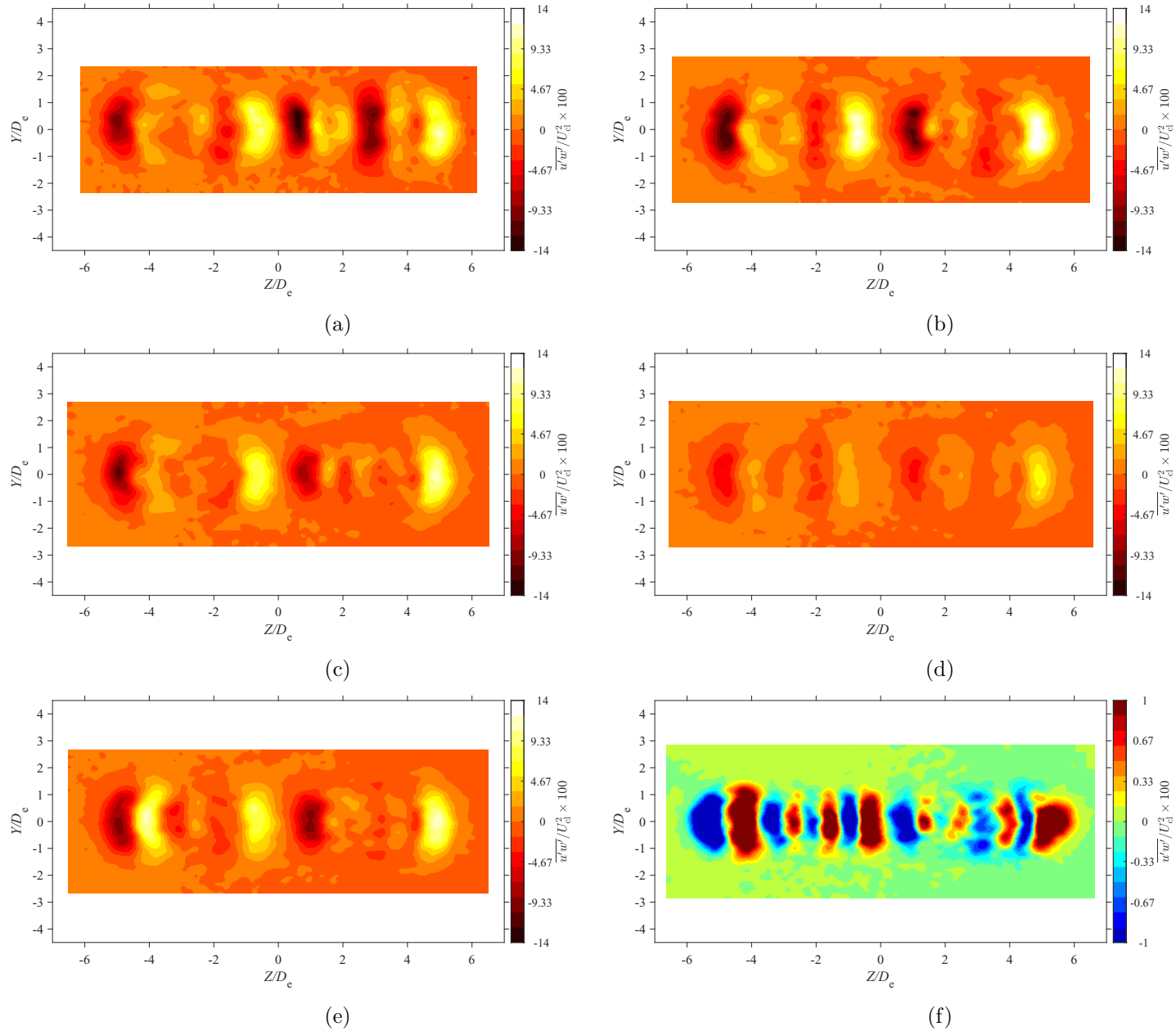
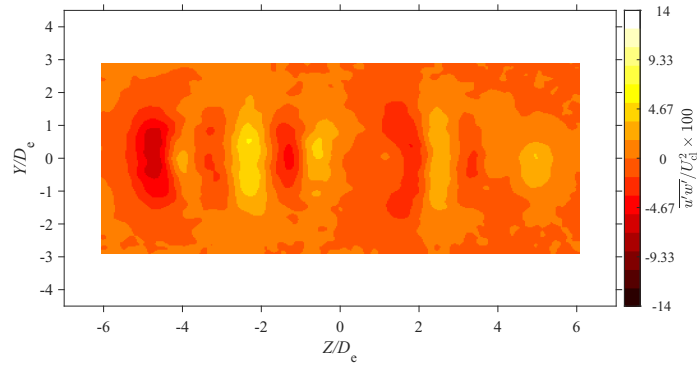
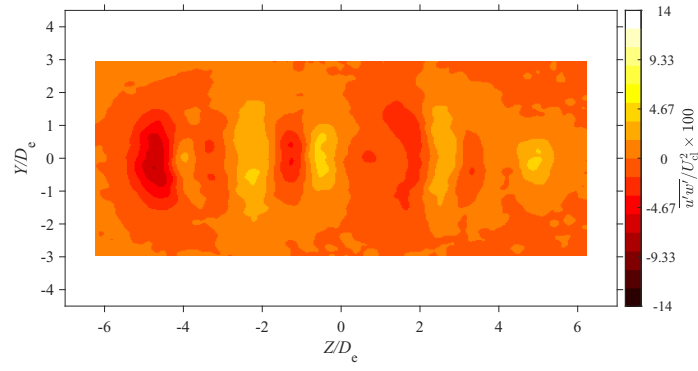


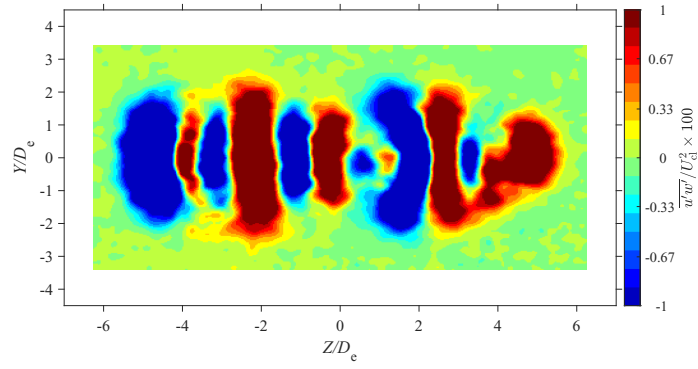
Figure E.95: Average spanwise Reynolds shear stress map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



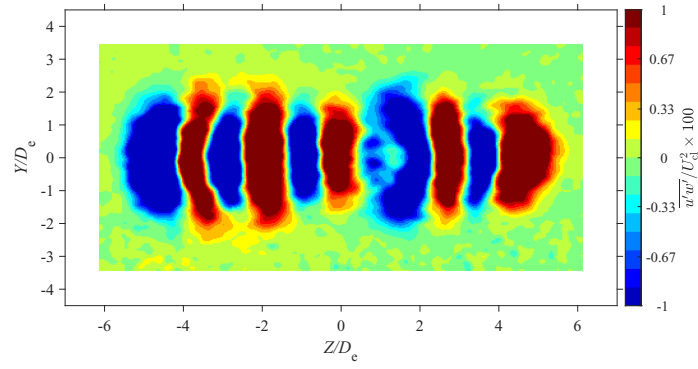
(a)



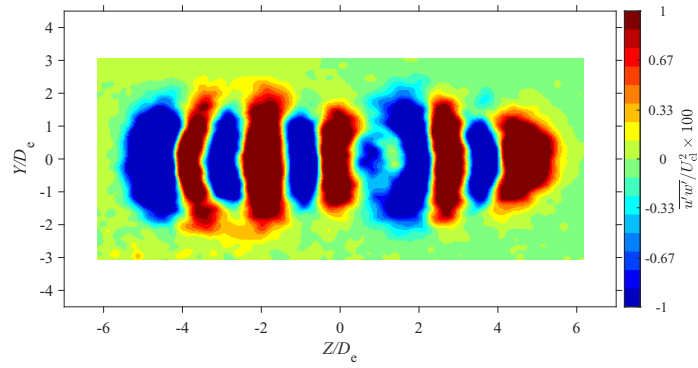
(b)



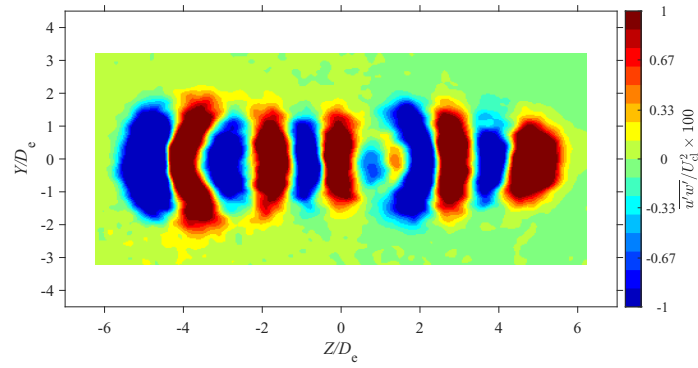
(c)



(d)



(e)



(f)

Figure E.96: Average spanwise Reynolds shear stress map at location $X = 7De$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

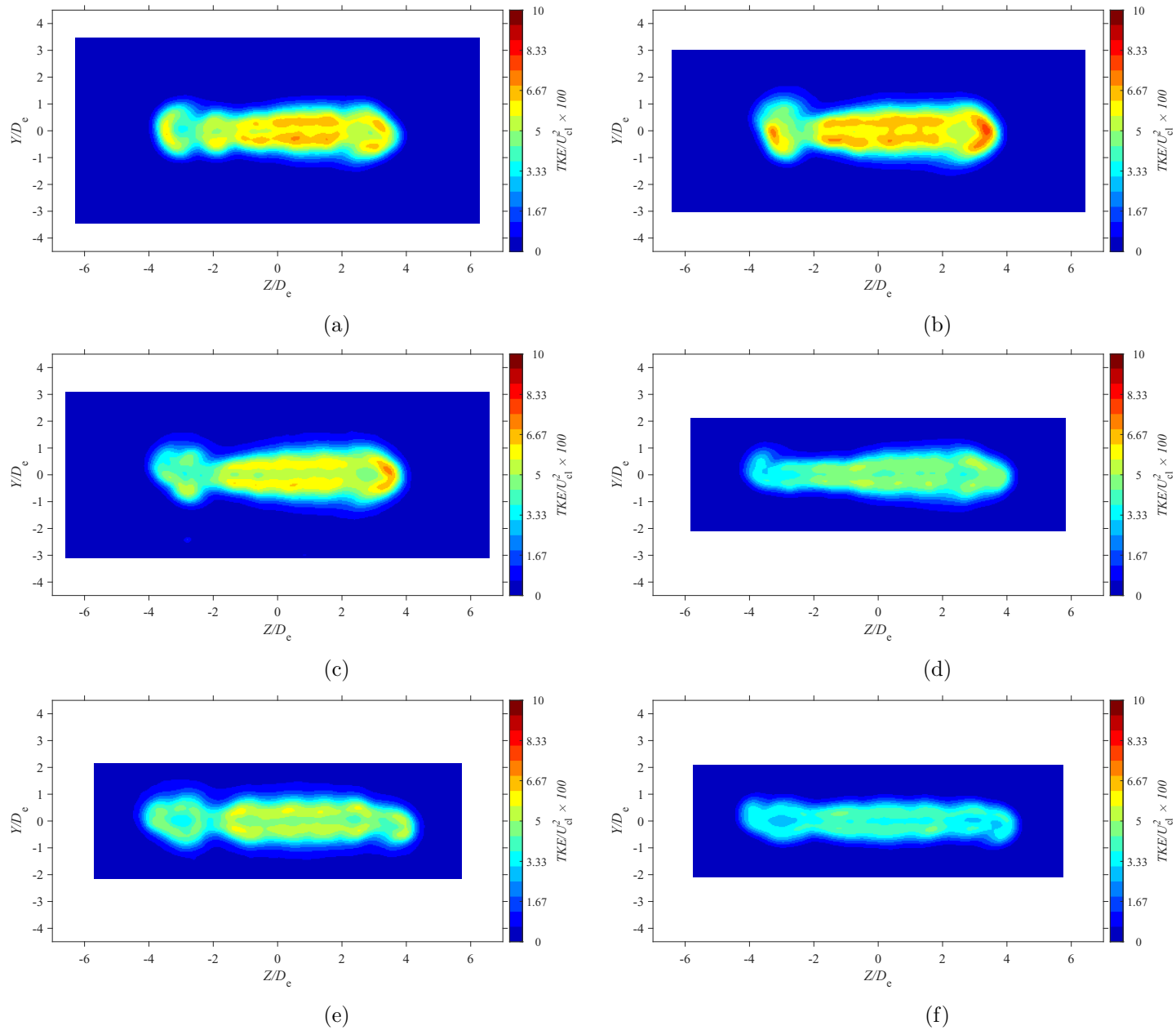


Figure E.97: Average turbulent kinetic energy map at location $X = 3De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

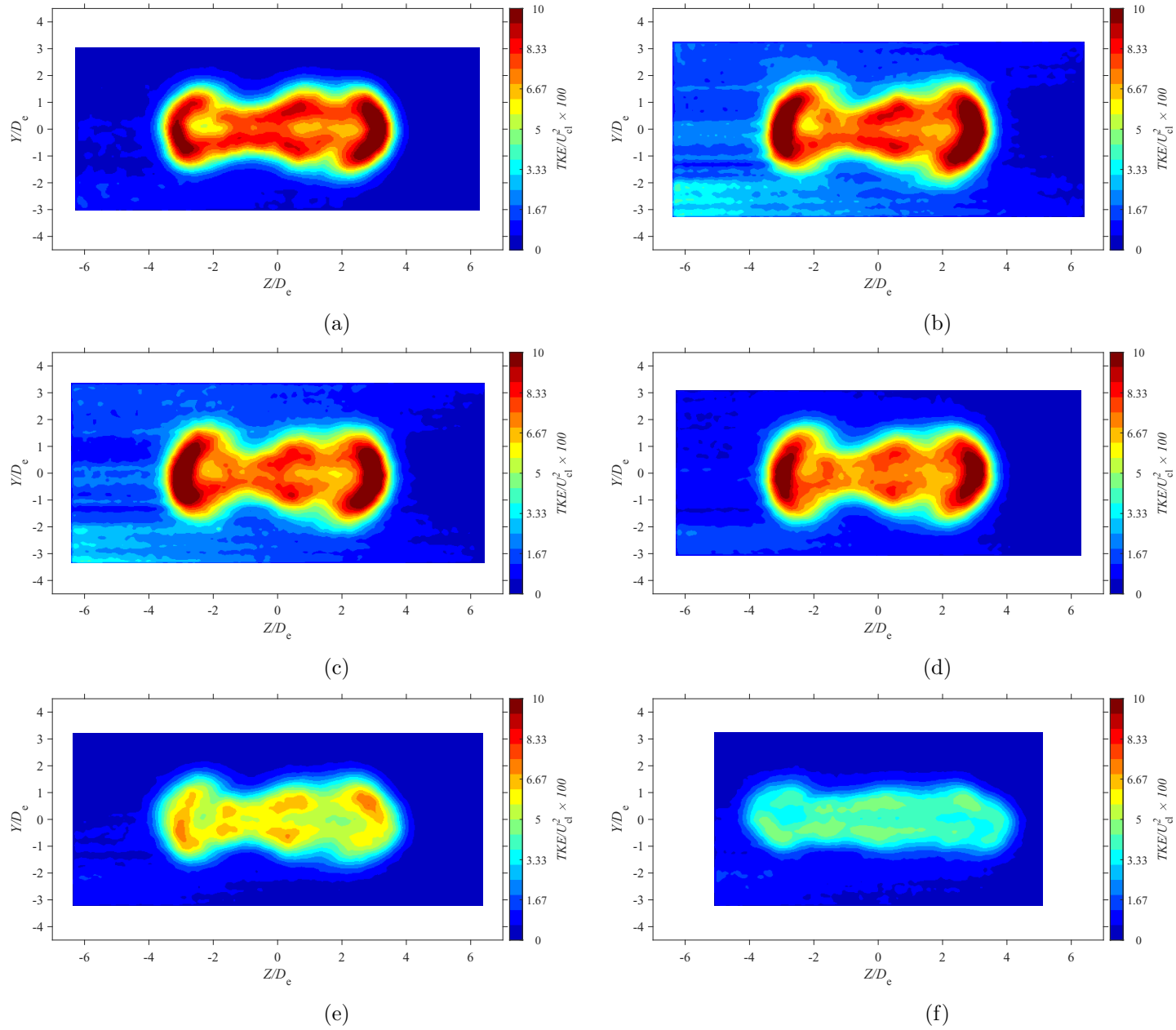
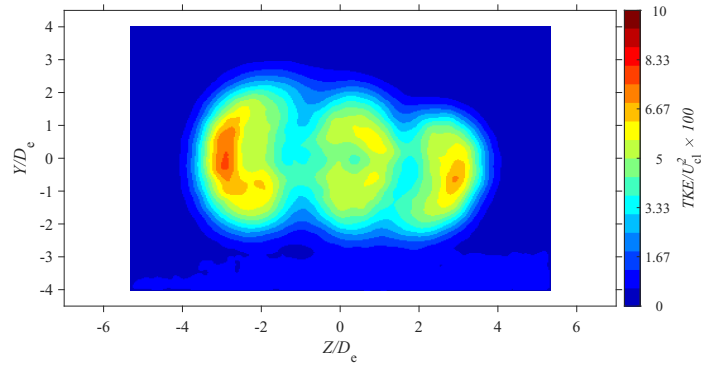
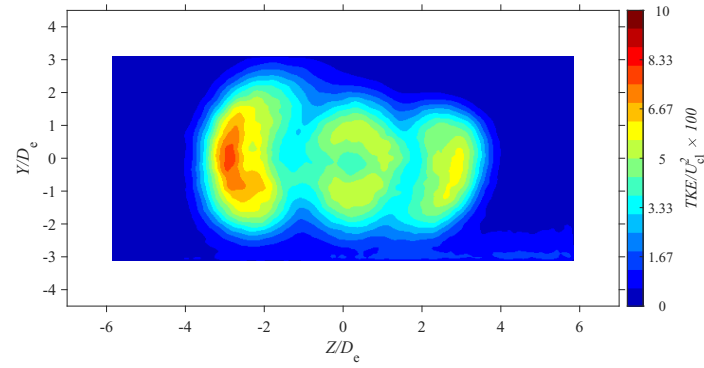


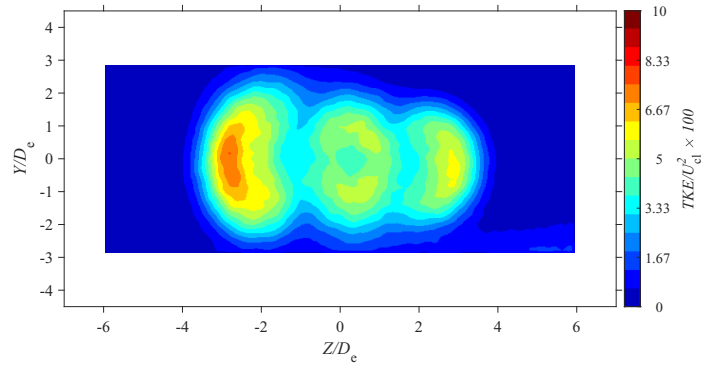
Figure E.98: Average turbulent kinetic energy map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



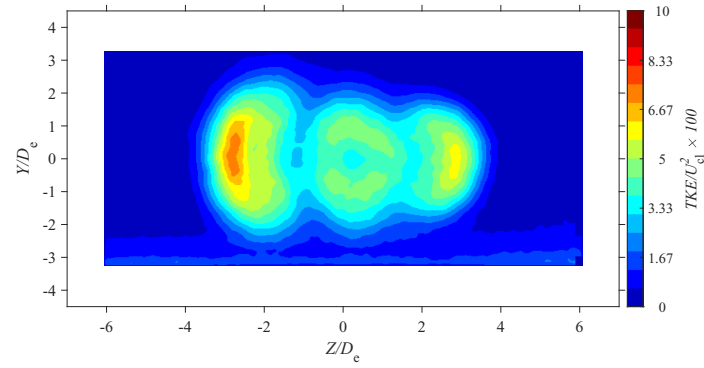
(a)



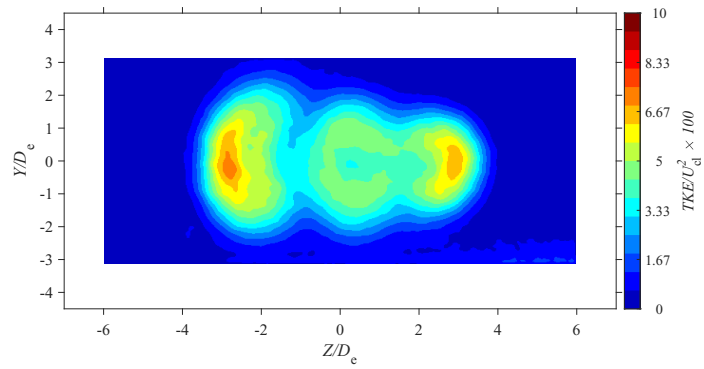
(b)



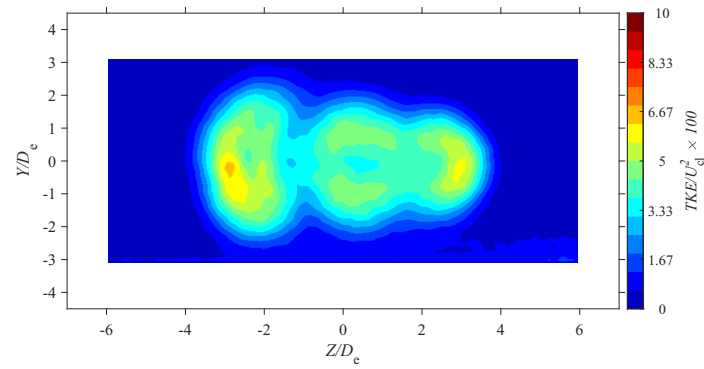
(c)



(d)



(e)



(f)

Figure E.99: Average turbulent kinetic energy map at location $X = 7De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

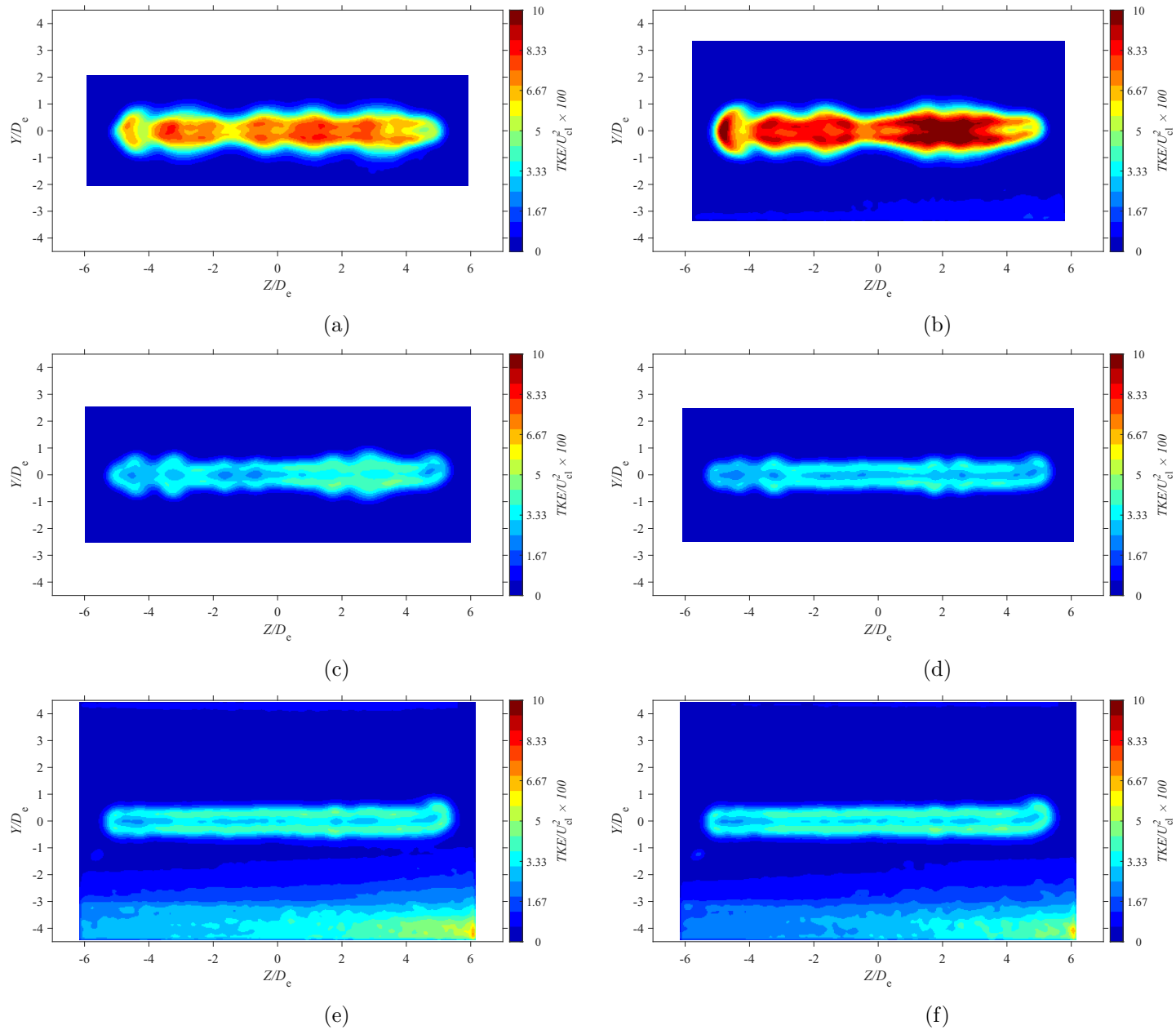
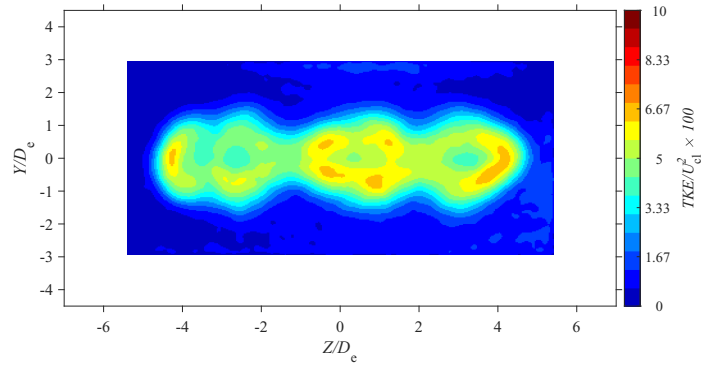
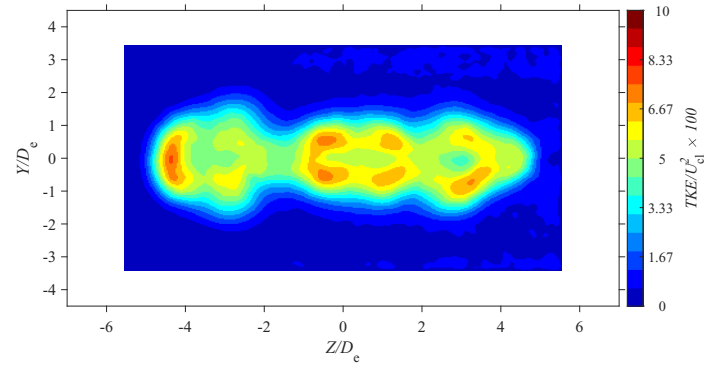


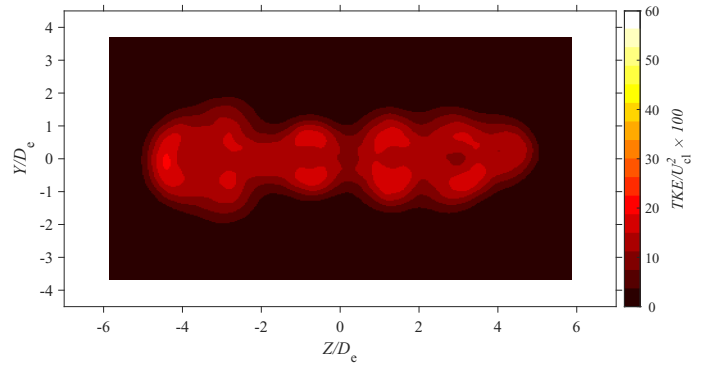
Figure E.100: Average turbulent kinetic energy map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



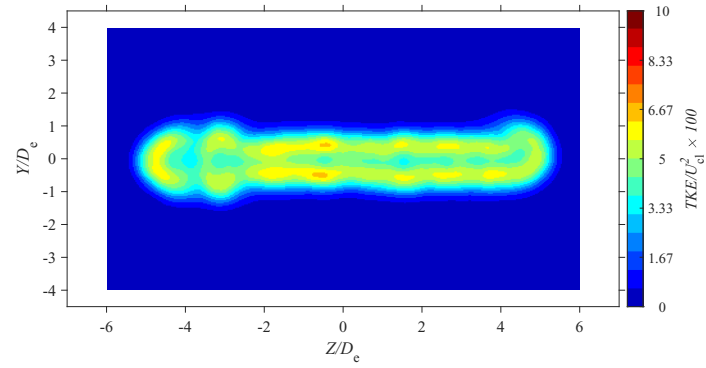
(a)



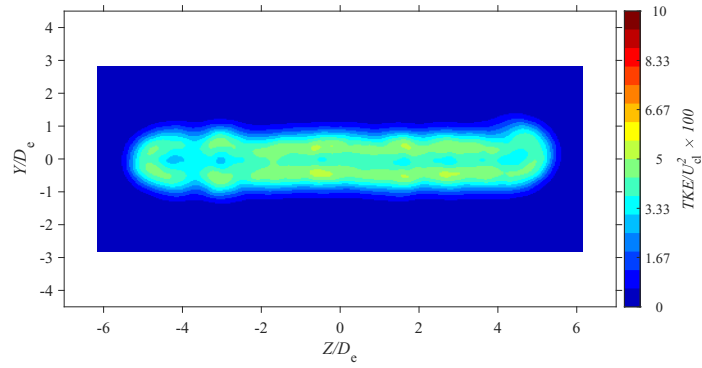
(b)



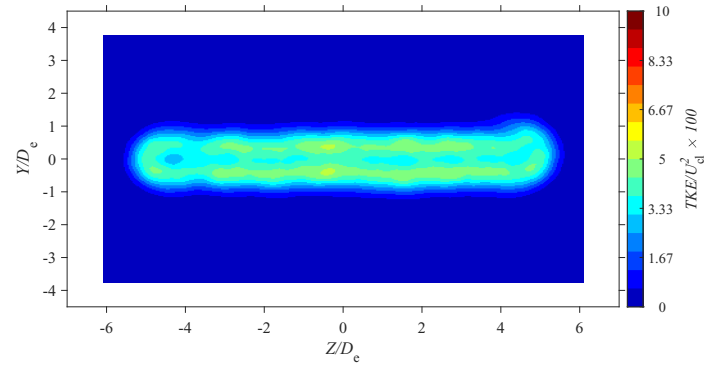
(c)



(d)



(e)



(f)

Figure E.101: Average turbulent kinetic energy map at location $X = 5De$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

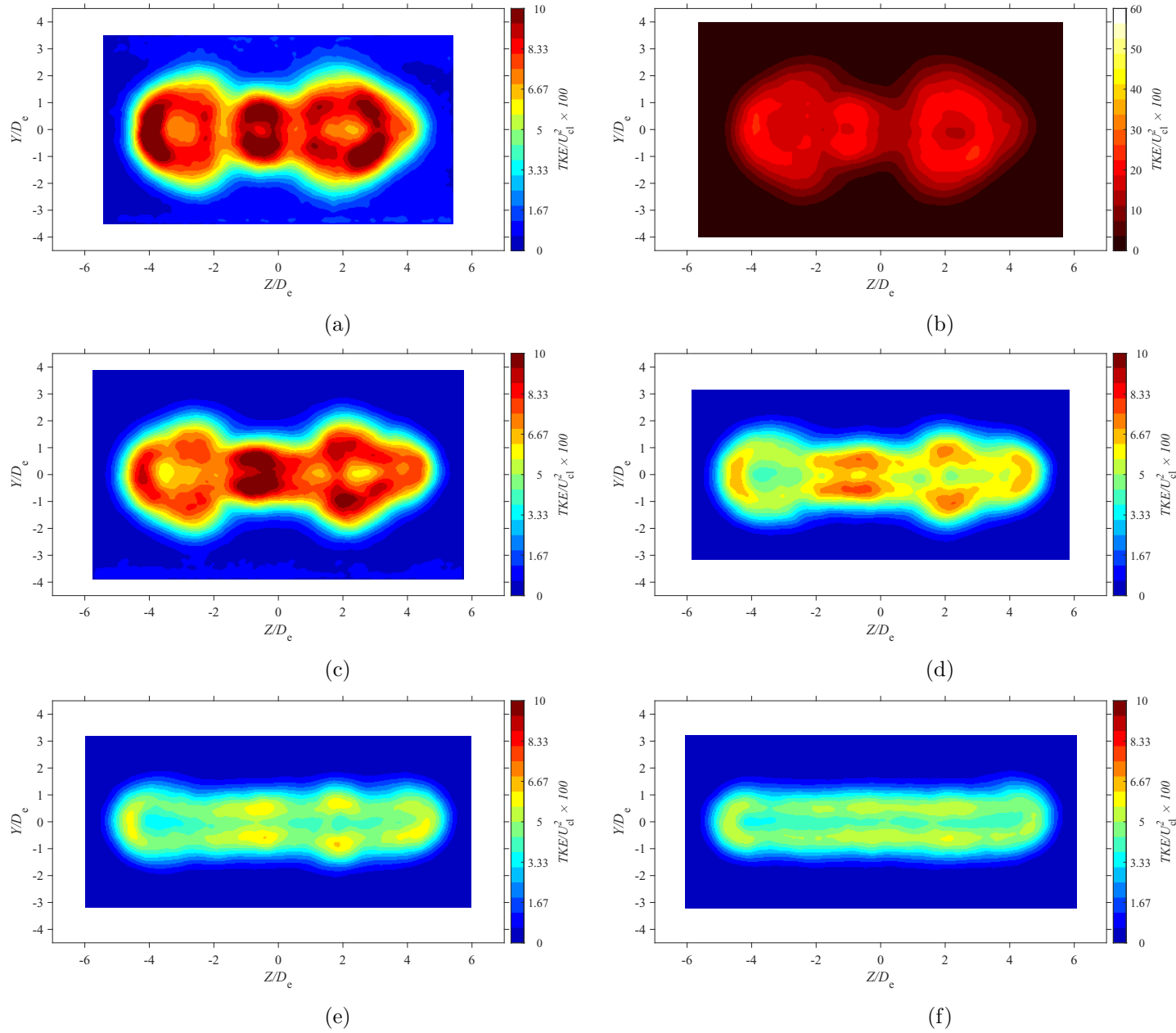
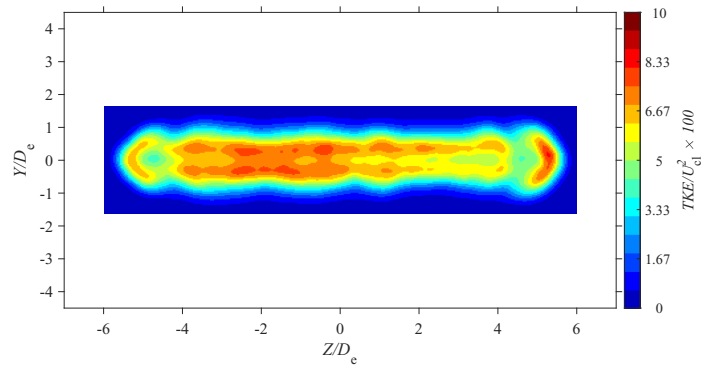
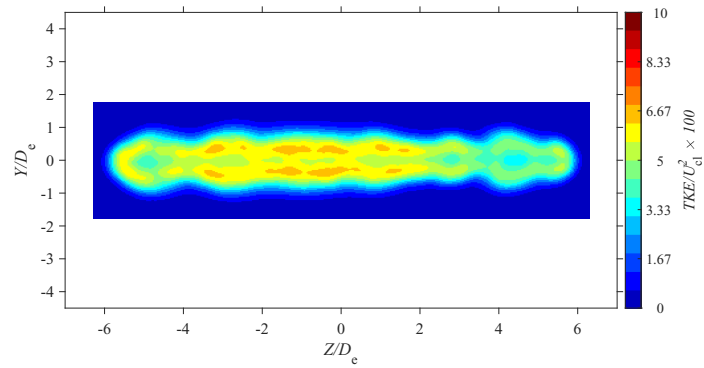


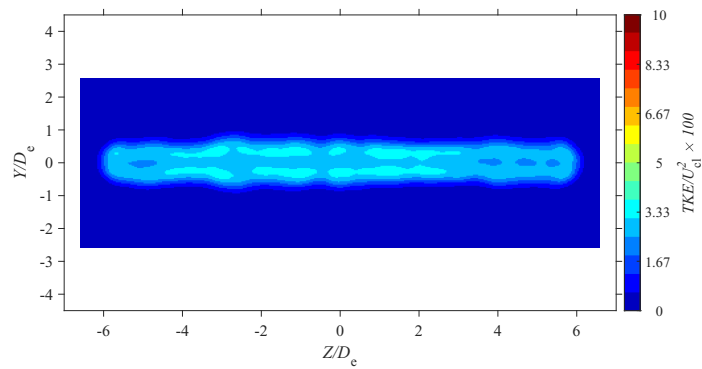
Figure E.102: Average turbulent kinetic energy map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



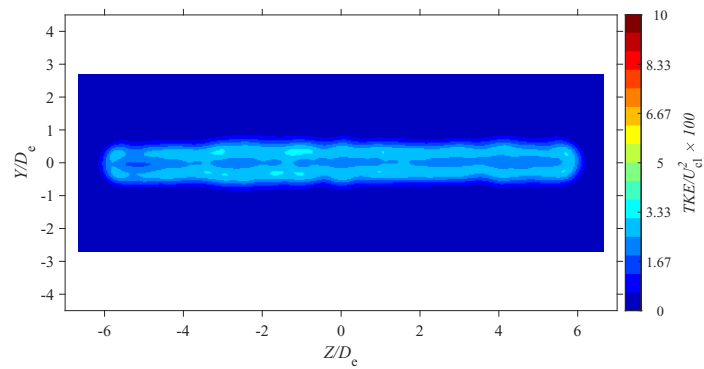
(a)



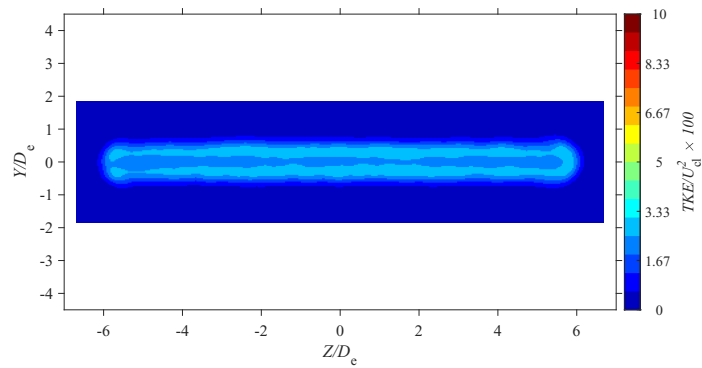
(b)



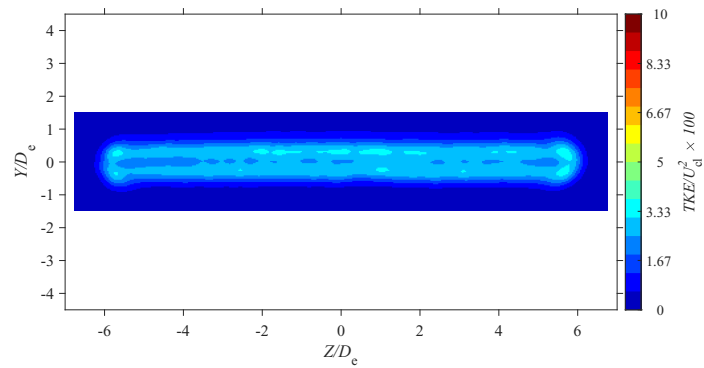
(c)



(d)

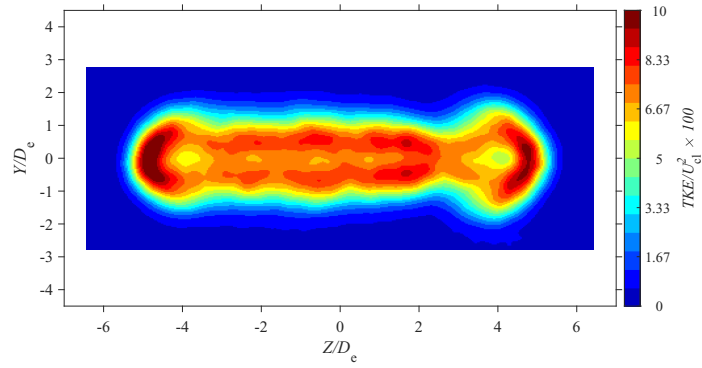


(e)

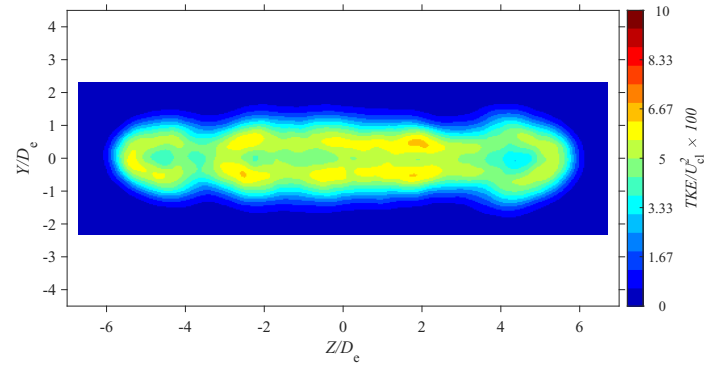


(f)

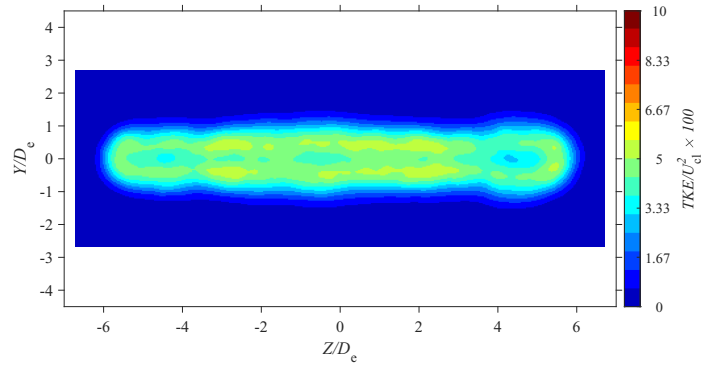
Figure E.103: Average turbulent kinetic energy map at location $X = 3De$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



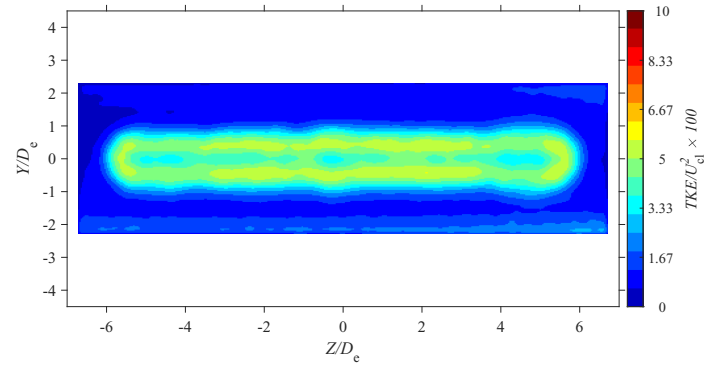
(a)



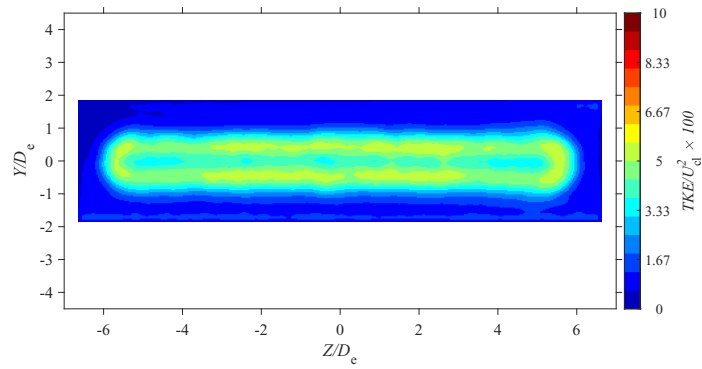
(b)



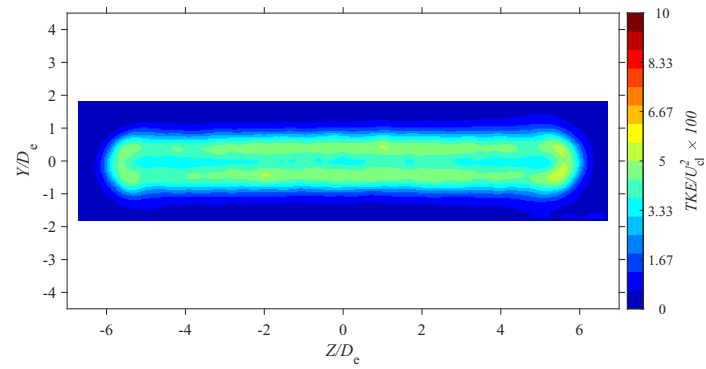
(c)



(d)



(e)



(f)

Figure E.104: Average turbulent kinetic energy map at location $X = 5De$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

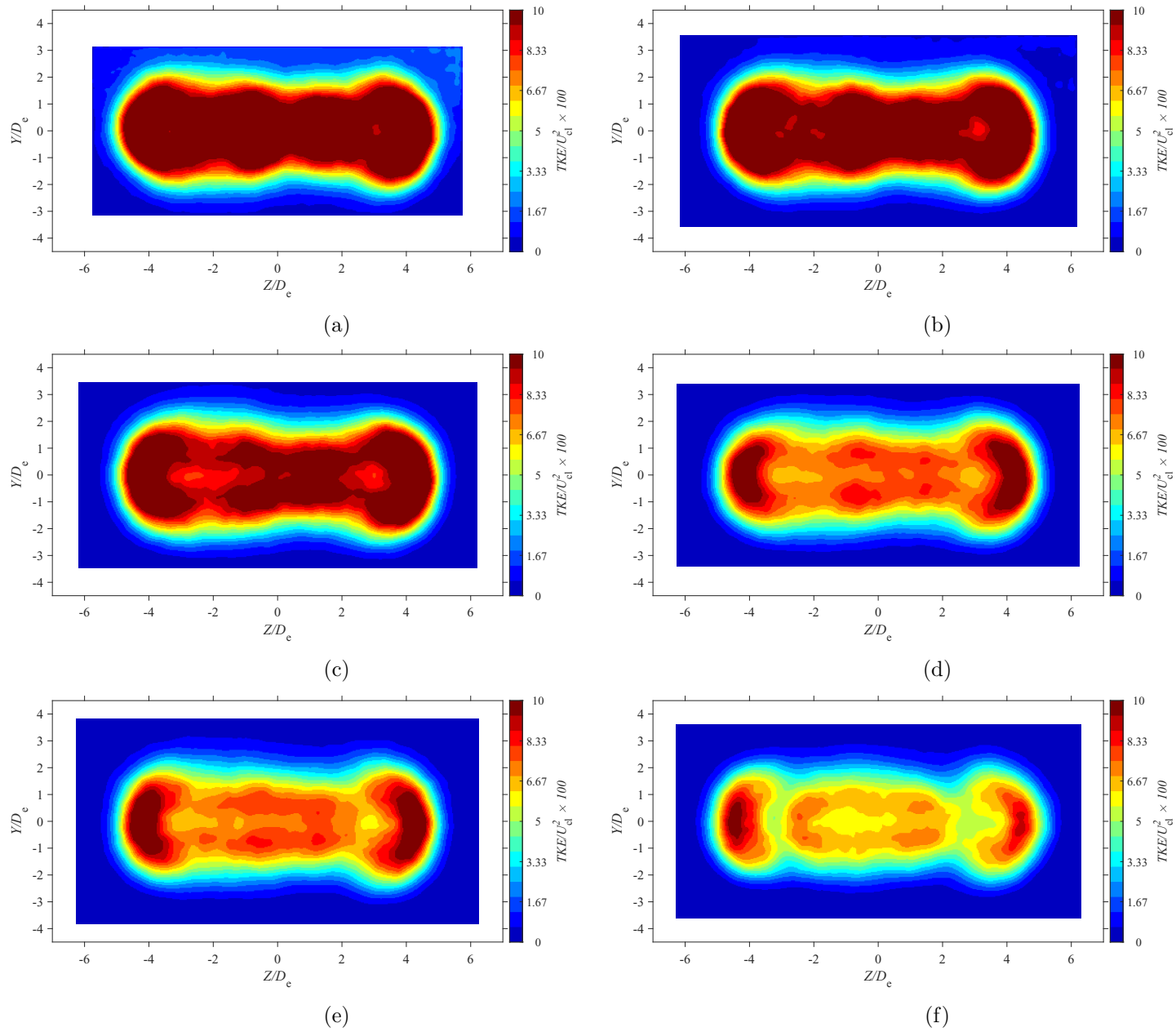
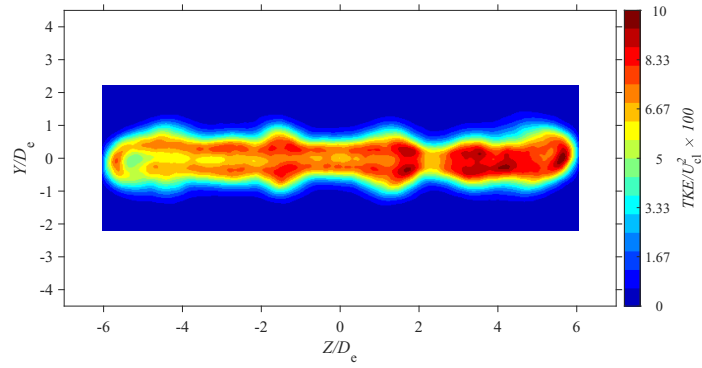
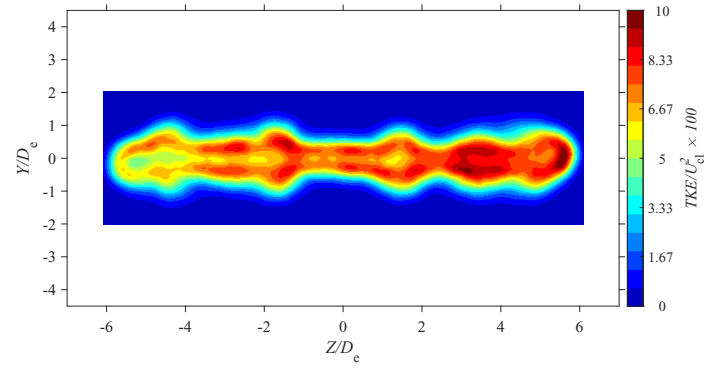


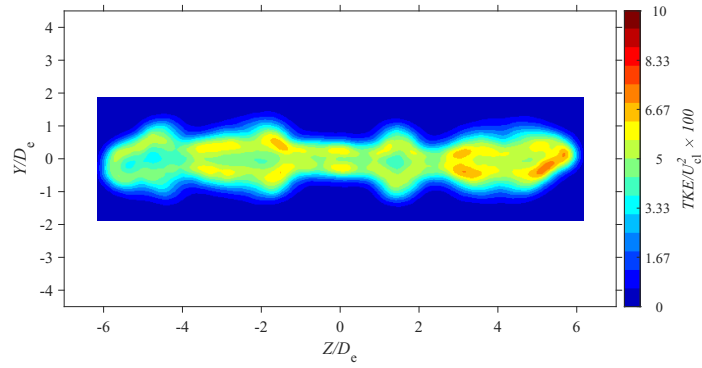
Figure E.105: Average turbulent kinetic energy map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$



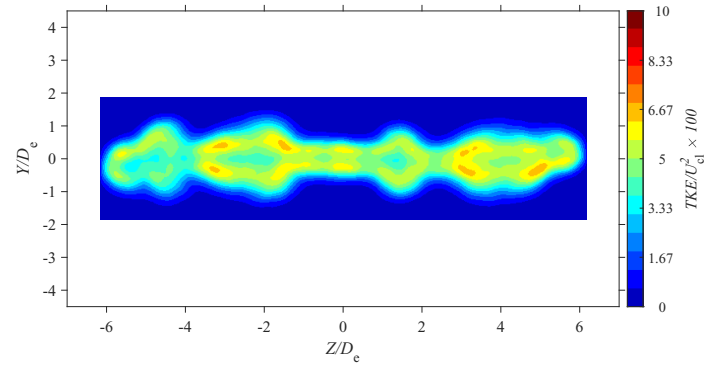
(a)



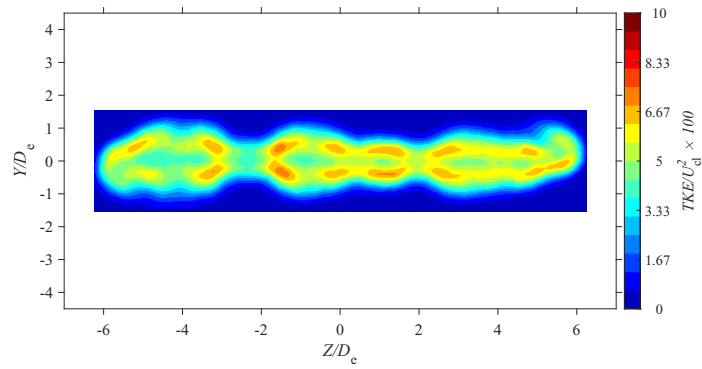
(b)



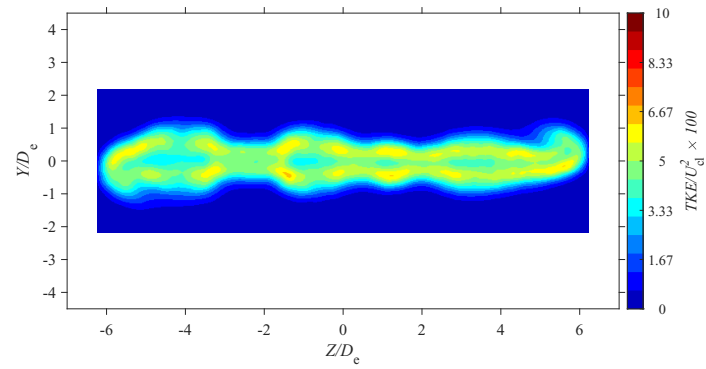
(c)



(d)



(e)



(f)

Figure E.106: Average turbulent kinetic energy map at location $X = 3De$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

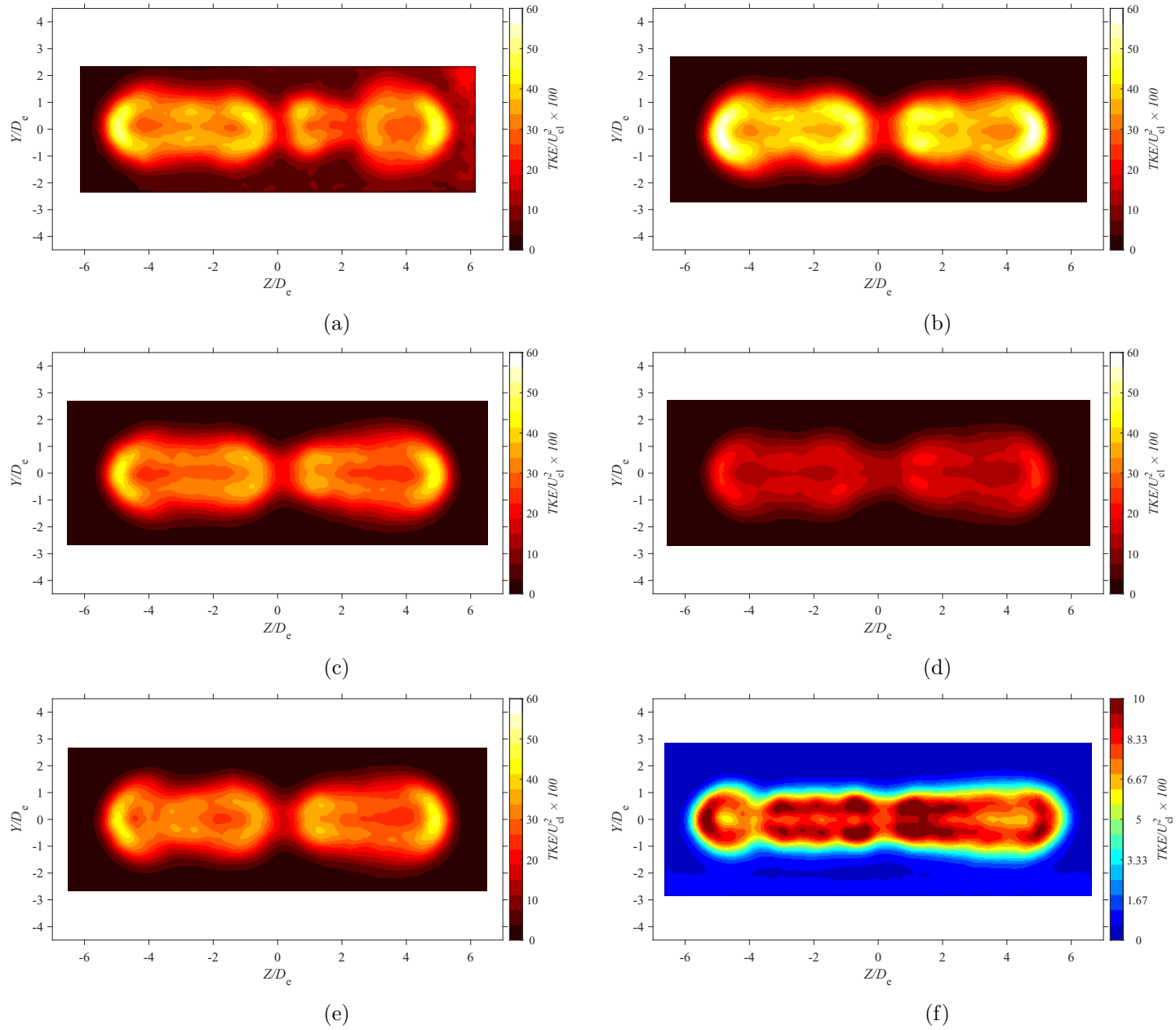


Figure E.107: Average turbulent kinetic energy map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

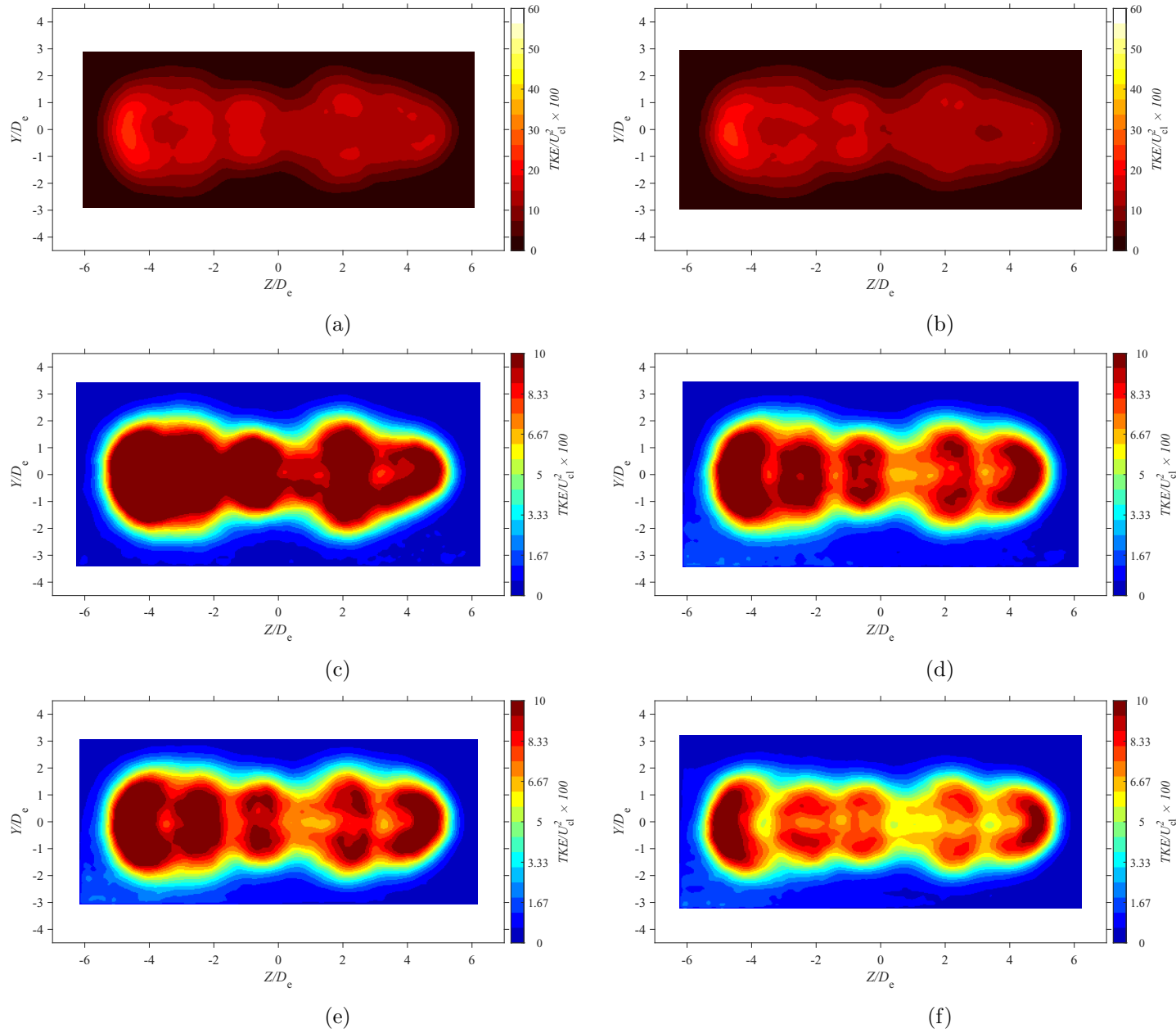


Figure E.108: Average turbulent kinetic energy map at location $X = 7De$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

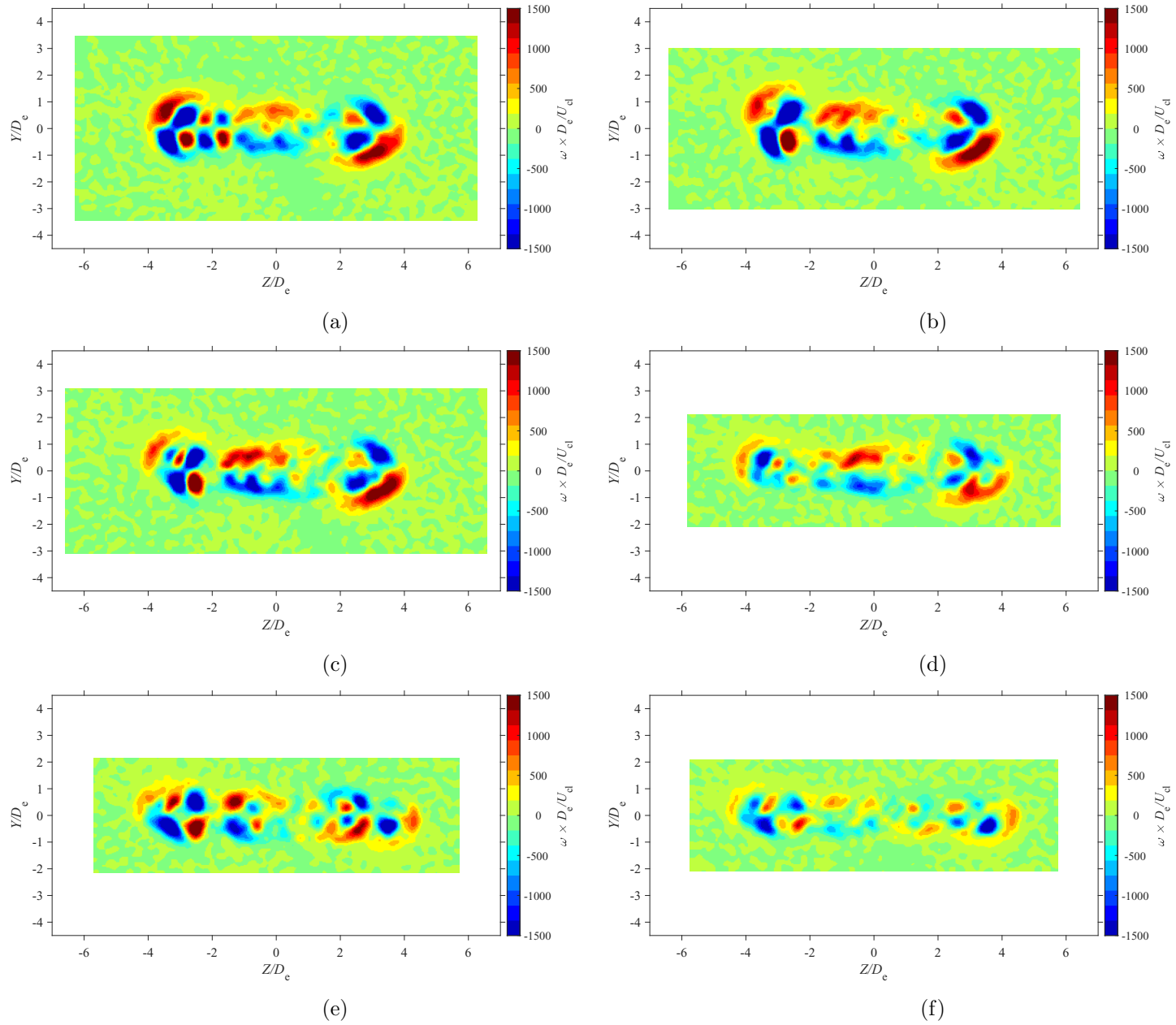


Figure E.109: Average vorticity map at location $X = 3De$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

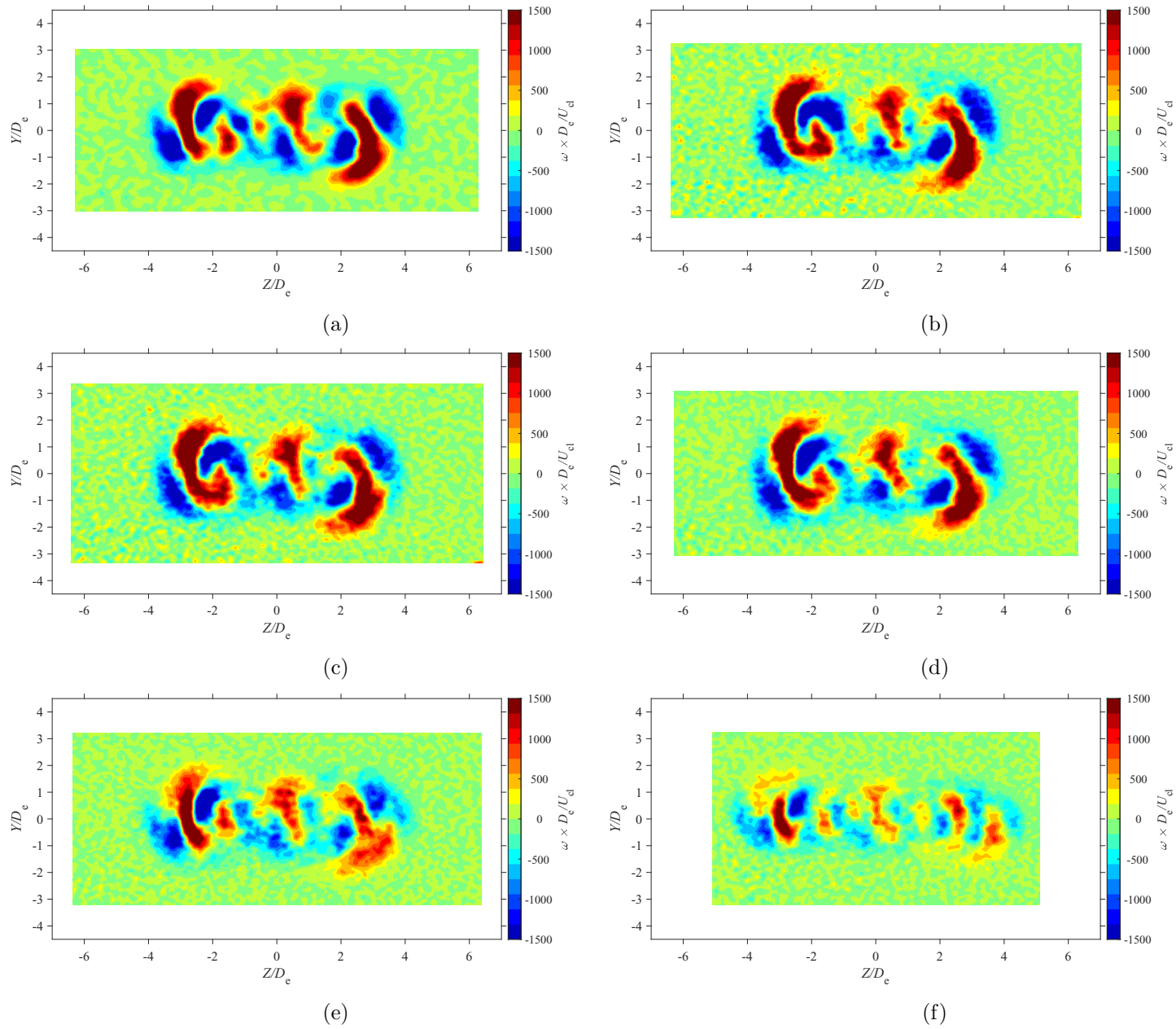


Figure E.110: Average vorticity map at location $X = 5D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

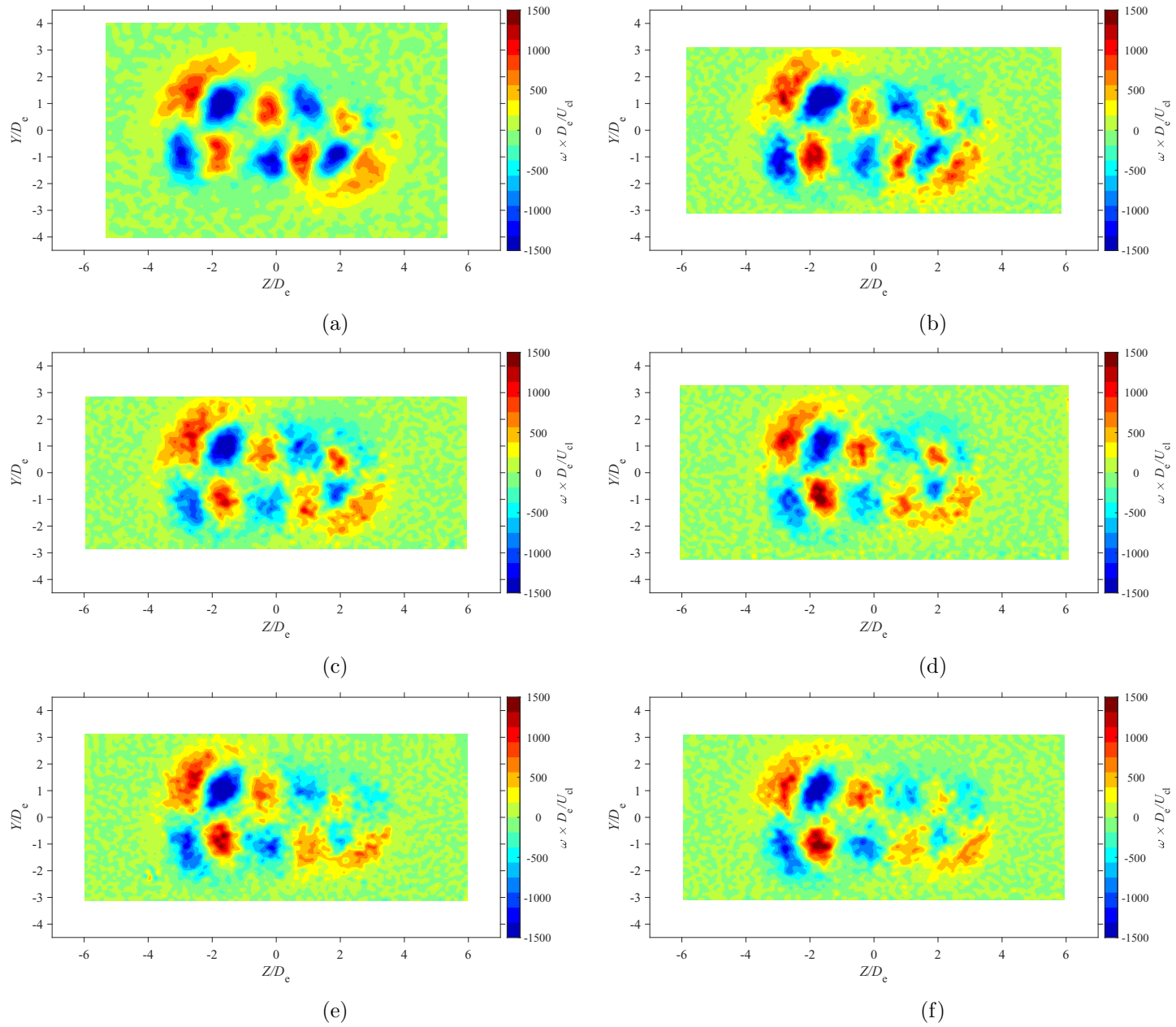


Figure E.111: Average vorticity map at location $X = 7D_e$ for $AR = 100$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

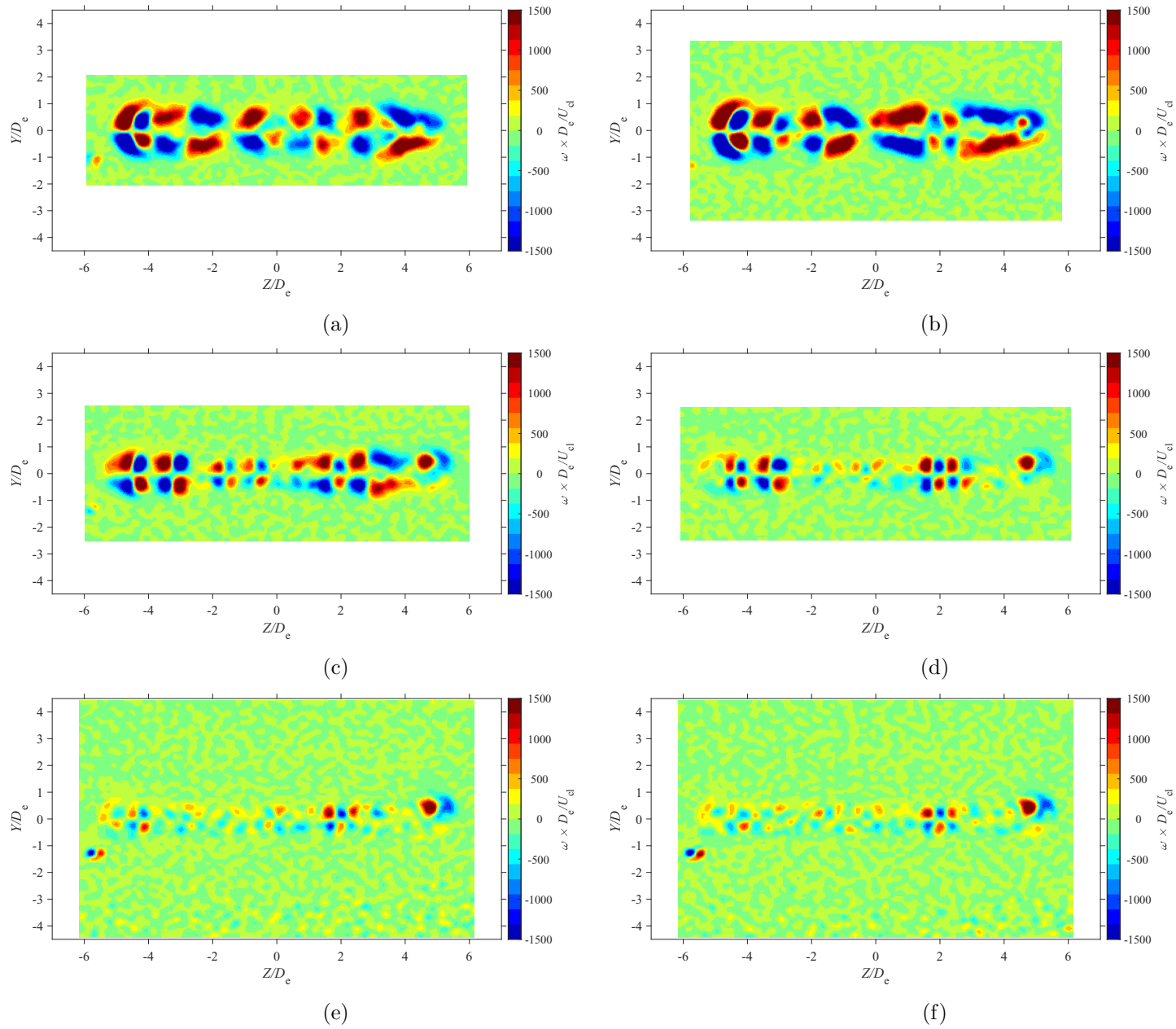


Figure E.112: Average vorticity map at location $X = 3D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

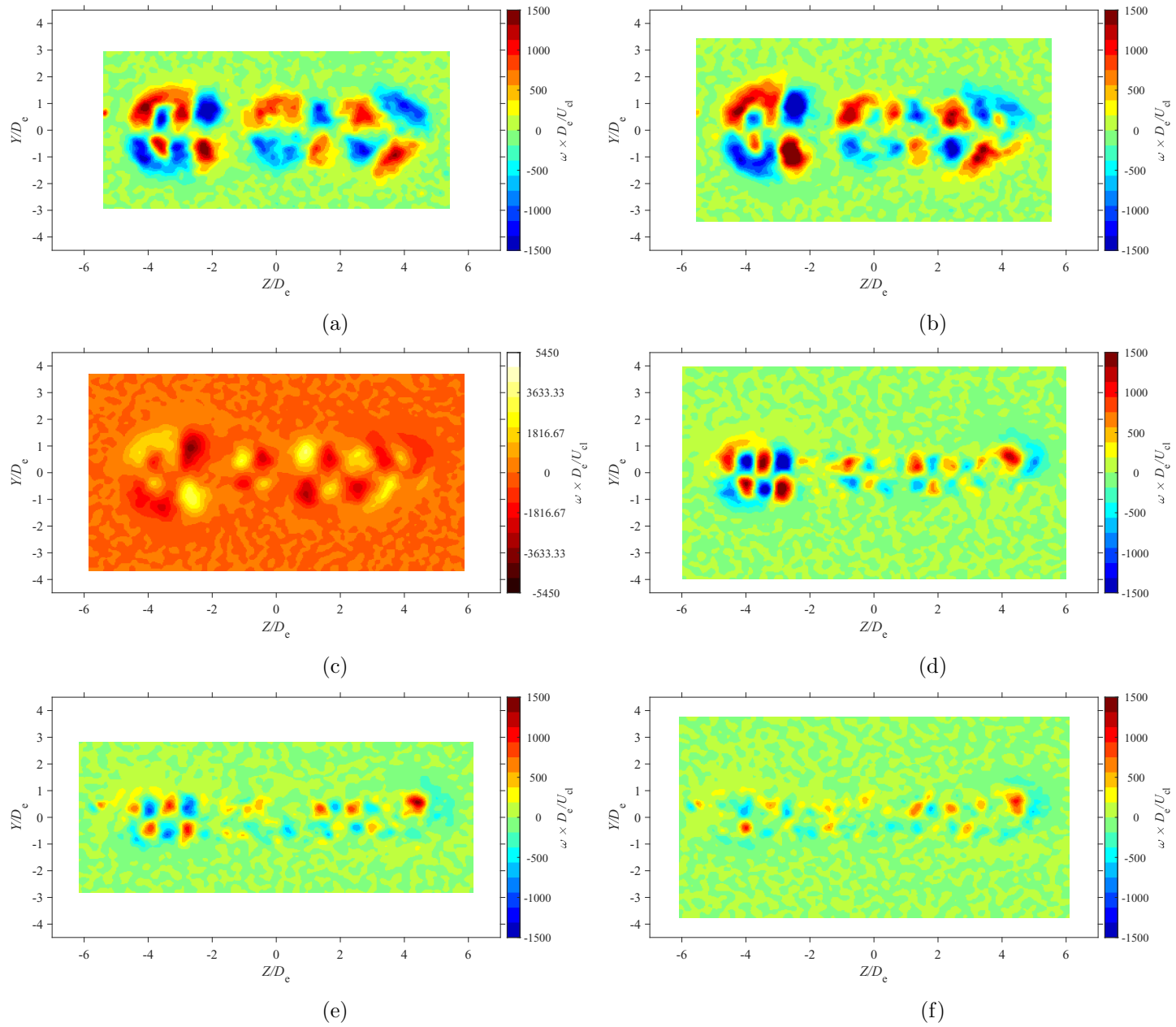


Figure E.113: Average vorticity map at location $X = 5D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

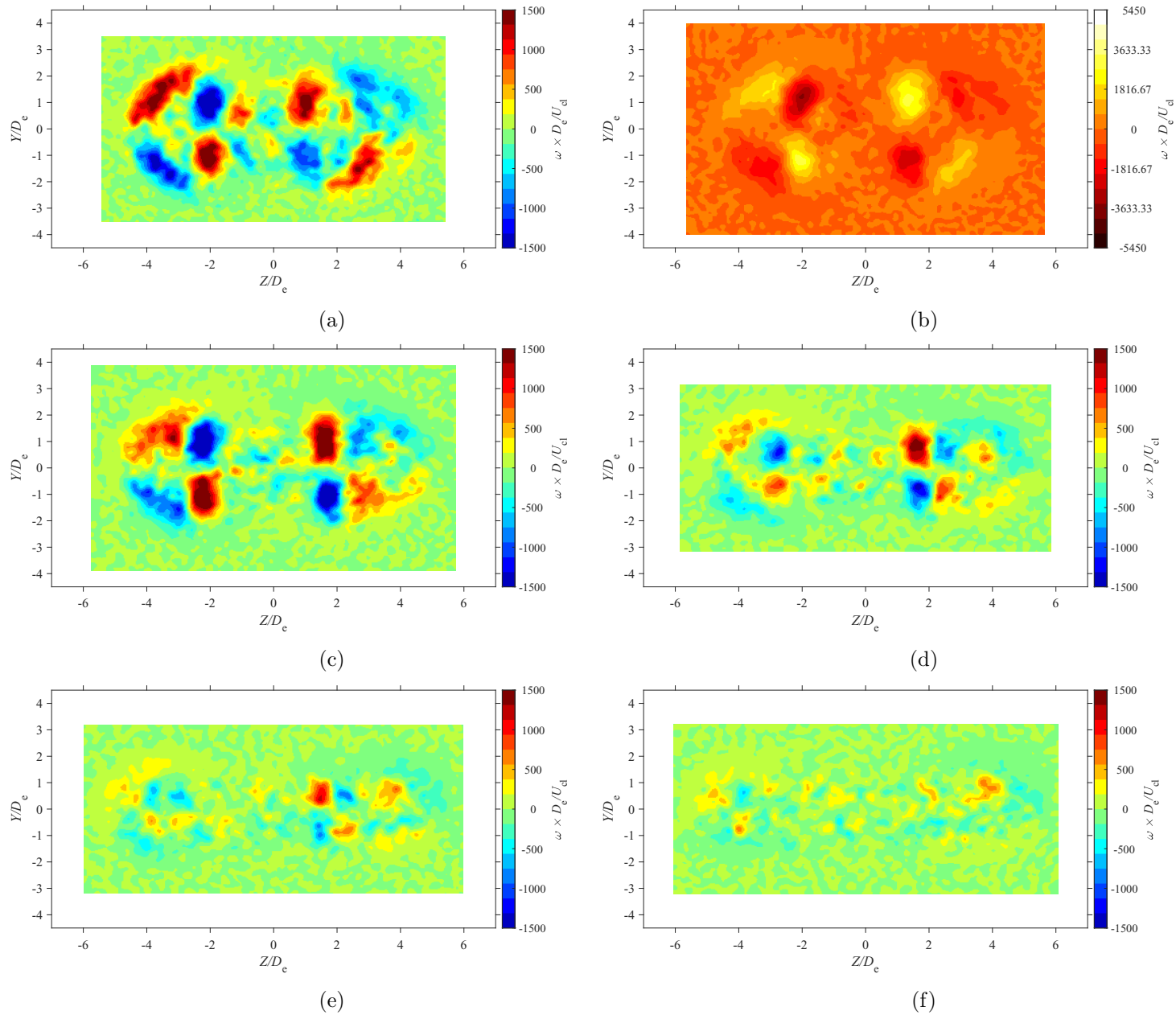


Figure E.114: Average vorticity map at location $X = 7D_e$ for $AR = 150$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

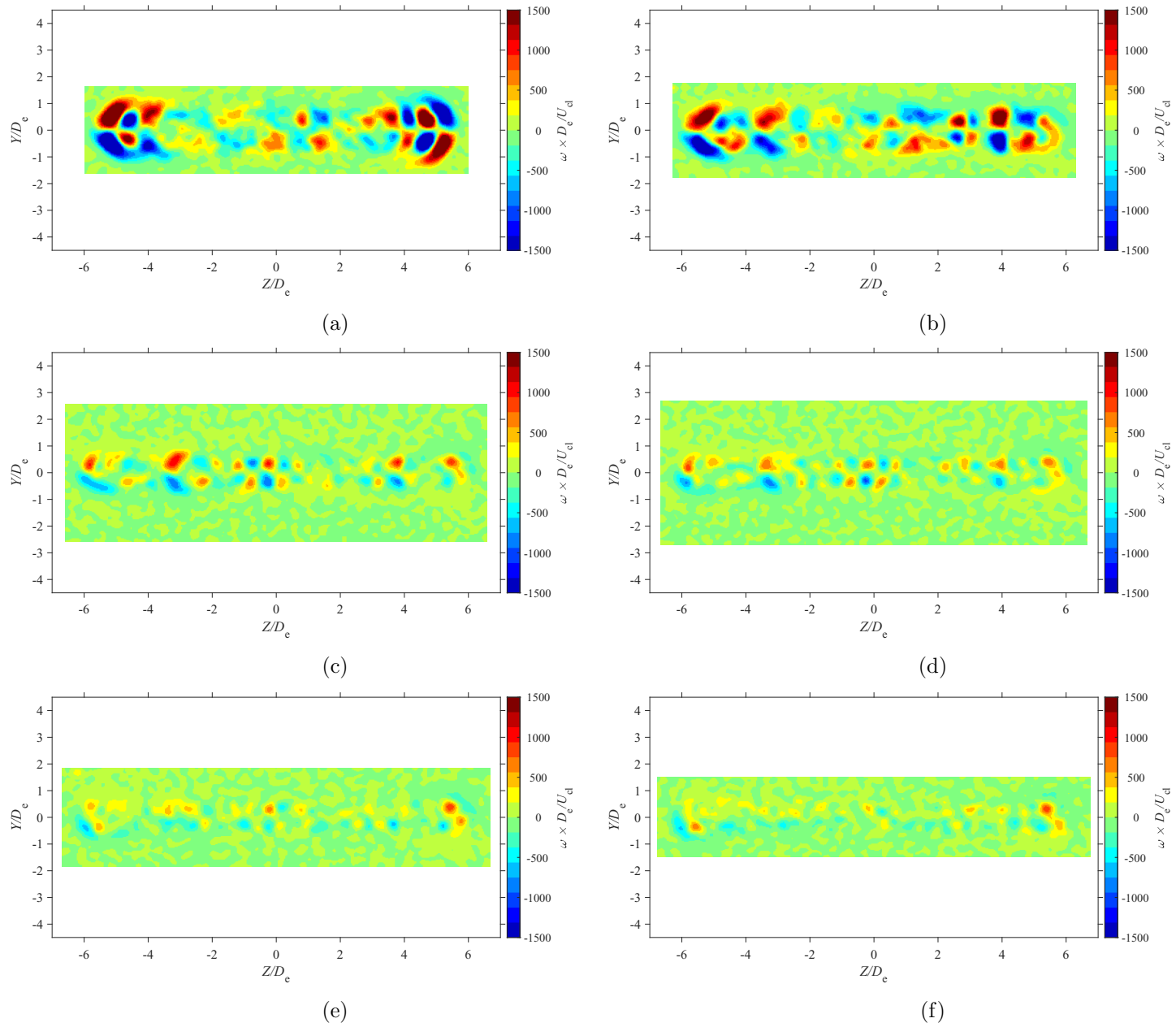


Figure E.115: Average vorticity map at location $X = 3De$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

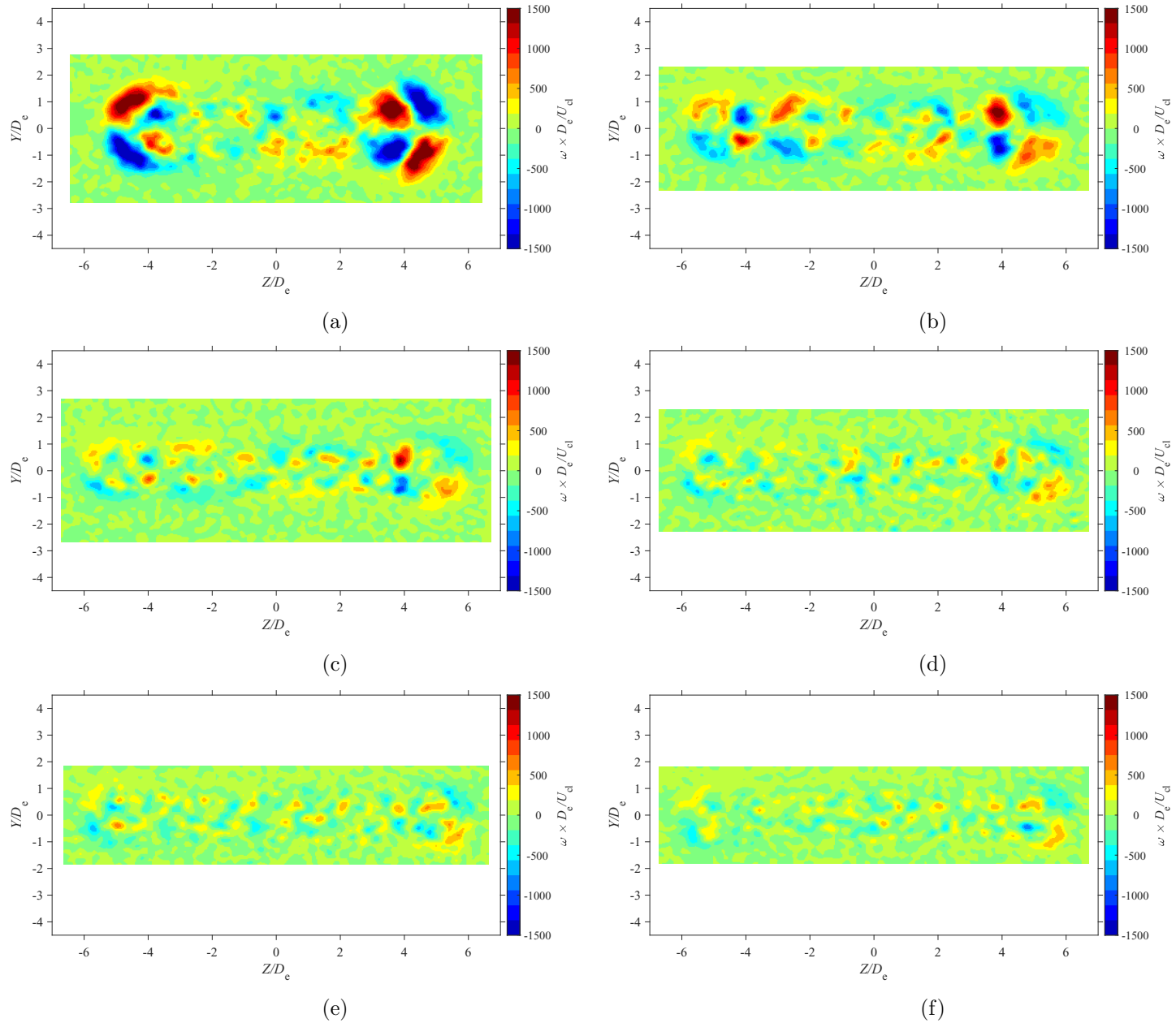


Figure E.116: Average vorticity map at location $X = 5D_e$ for $AR = 200$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

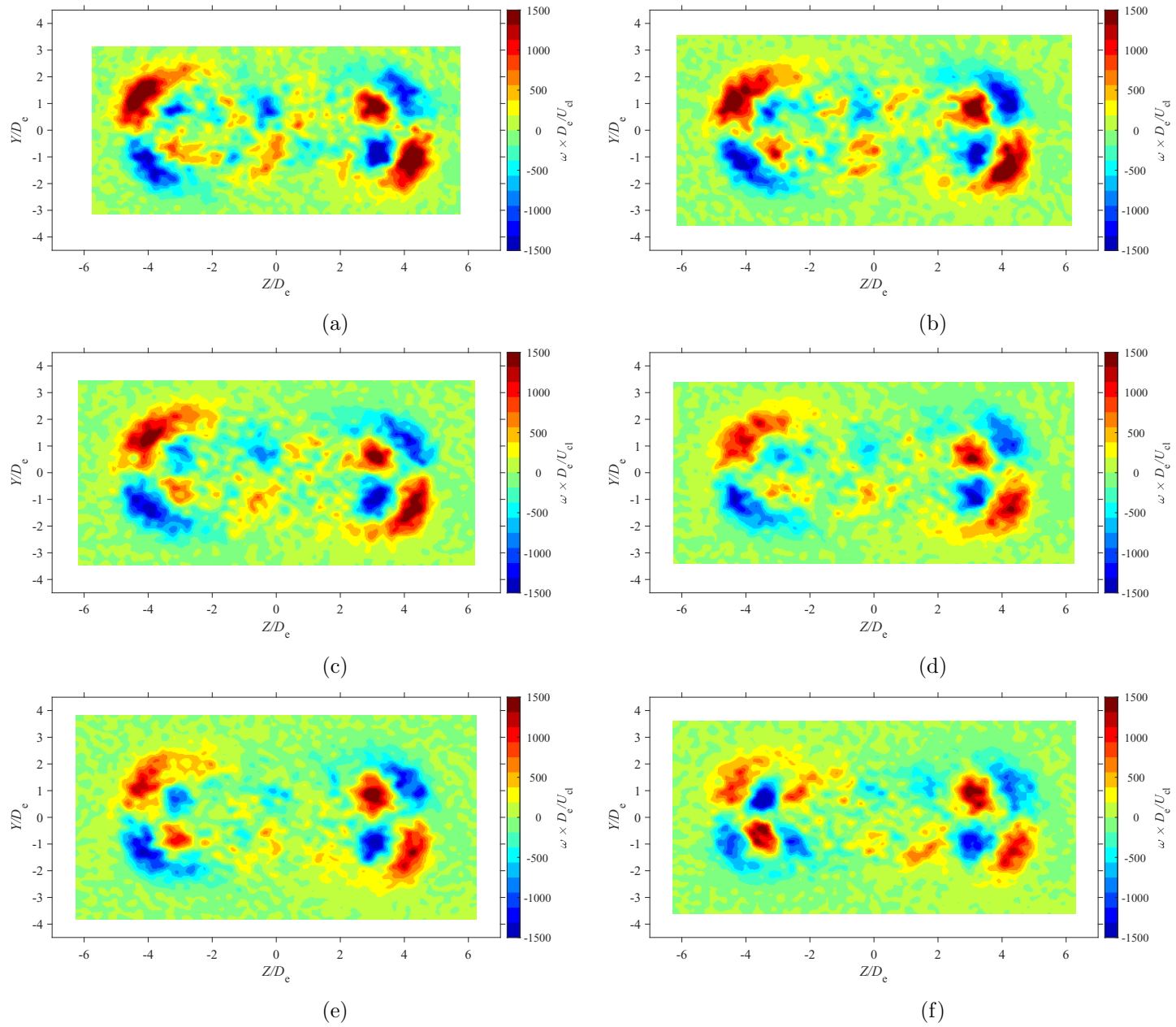


Figure E.117: Average vorticity map at location $X = 7D_e$ for $AR = 20$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

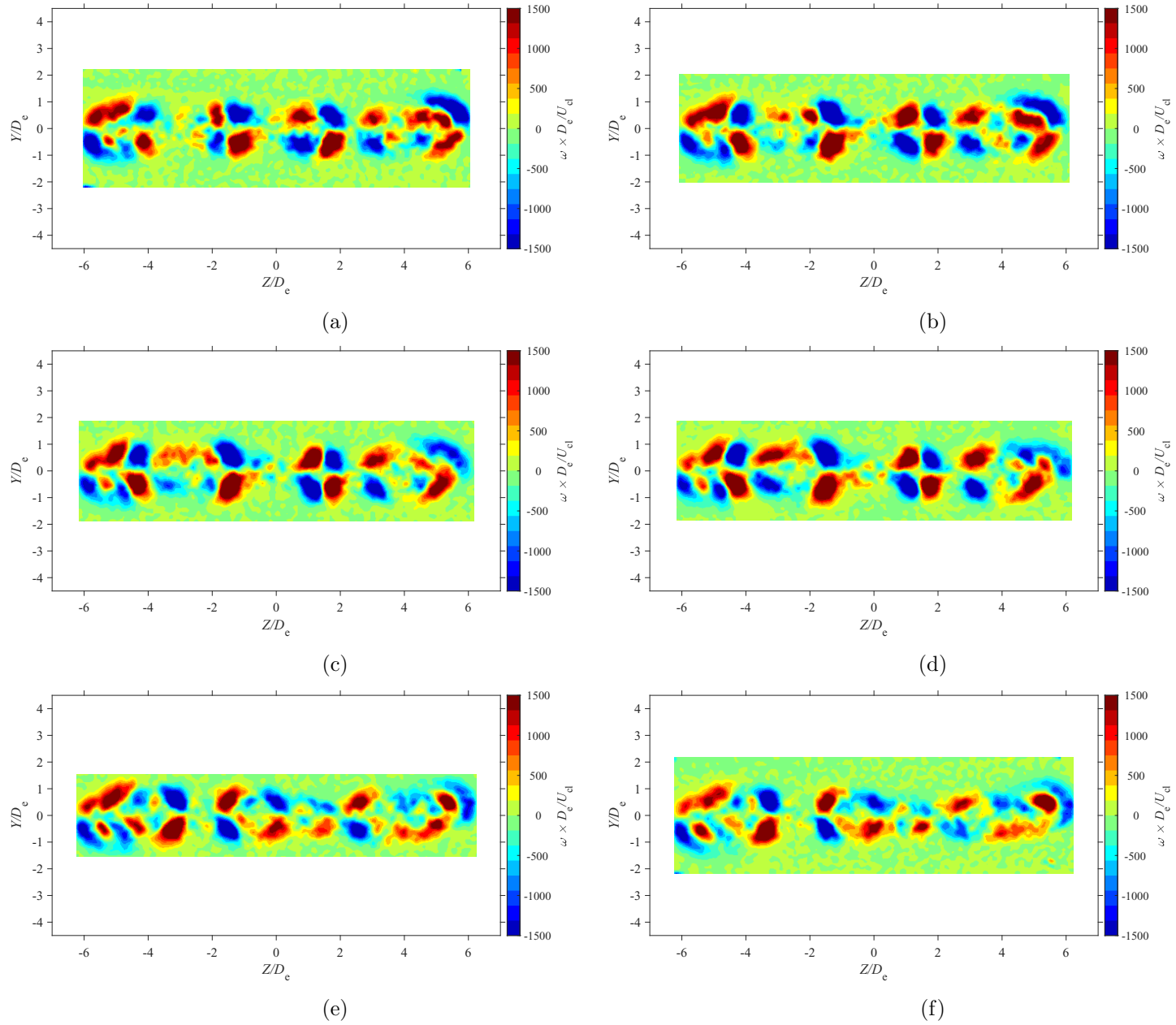


Figure E.118: Average vorticity map at location $X = 3De$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

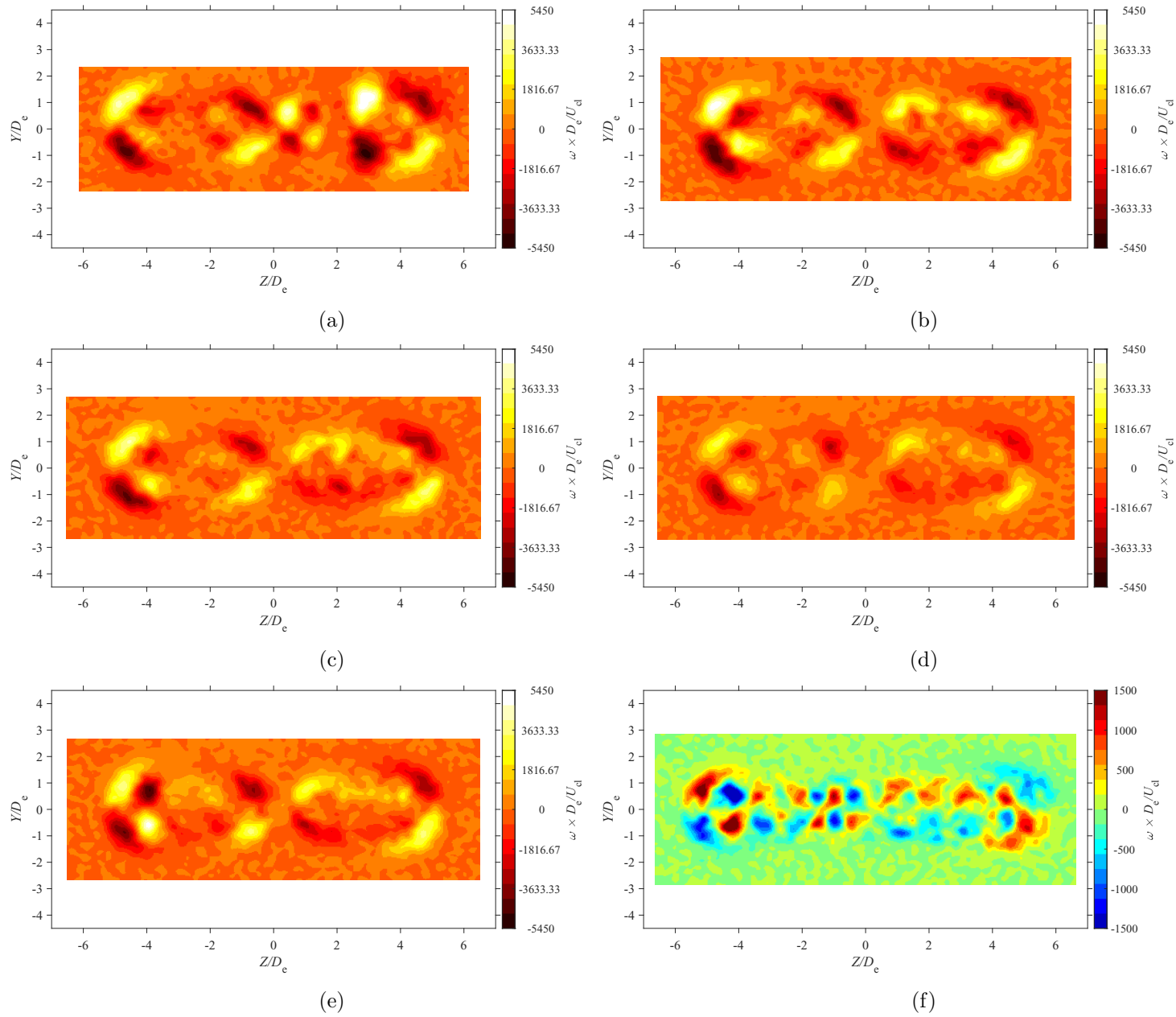


Figure E.119: Average vorticity map at location $X = 5D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

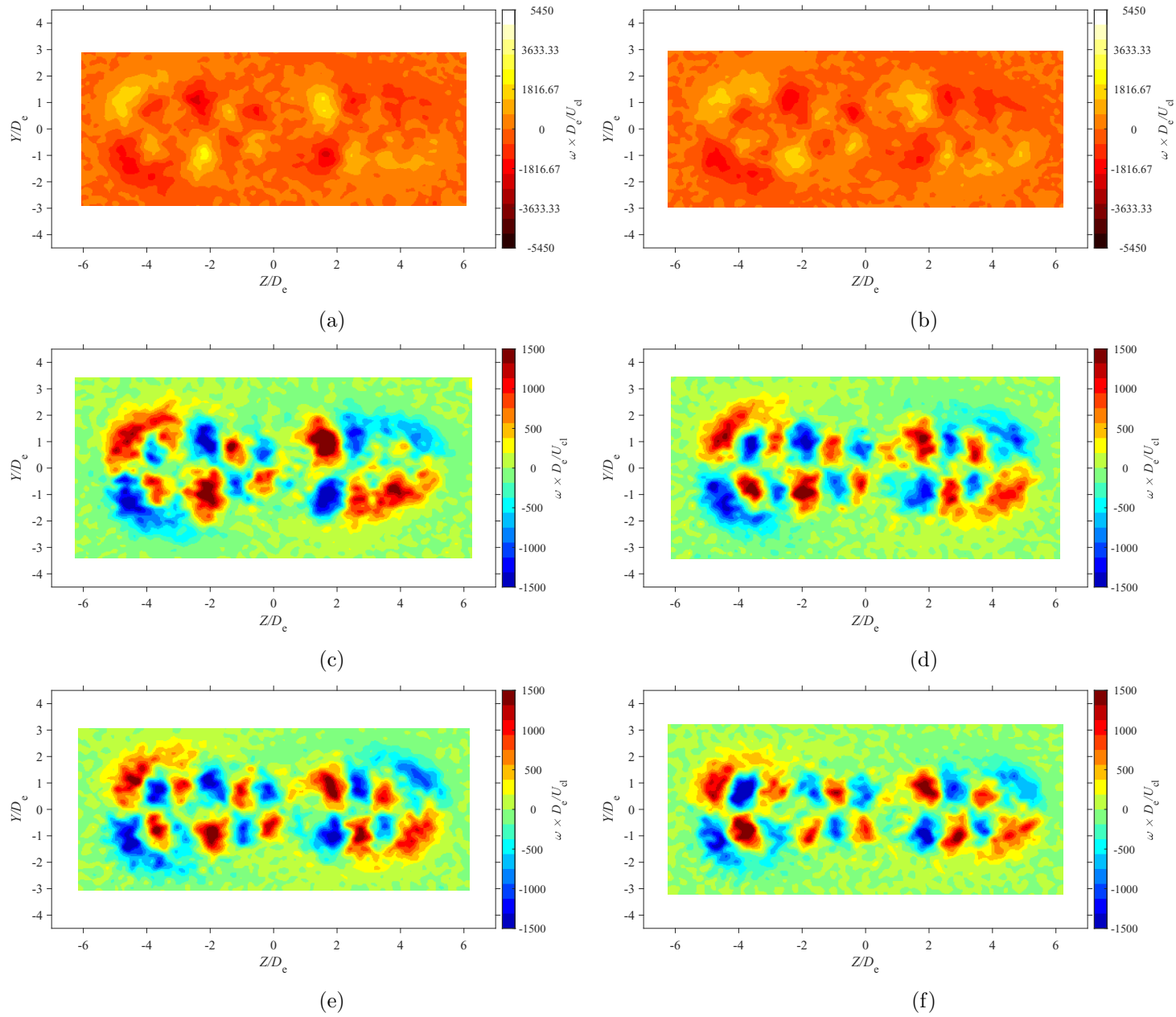


Figure E.120: Average vorticity map at location $X = 7D_e$ for $AR = 210$ at Reynolds numbers (a) $Re = 10800$, (b) $Re = 14400$, (c) $Re = 18000$, and (d) $Re = 21600$, (e) $Re = 25200$, and (f) $Re = 28800$

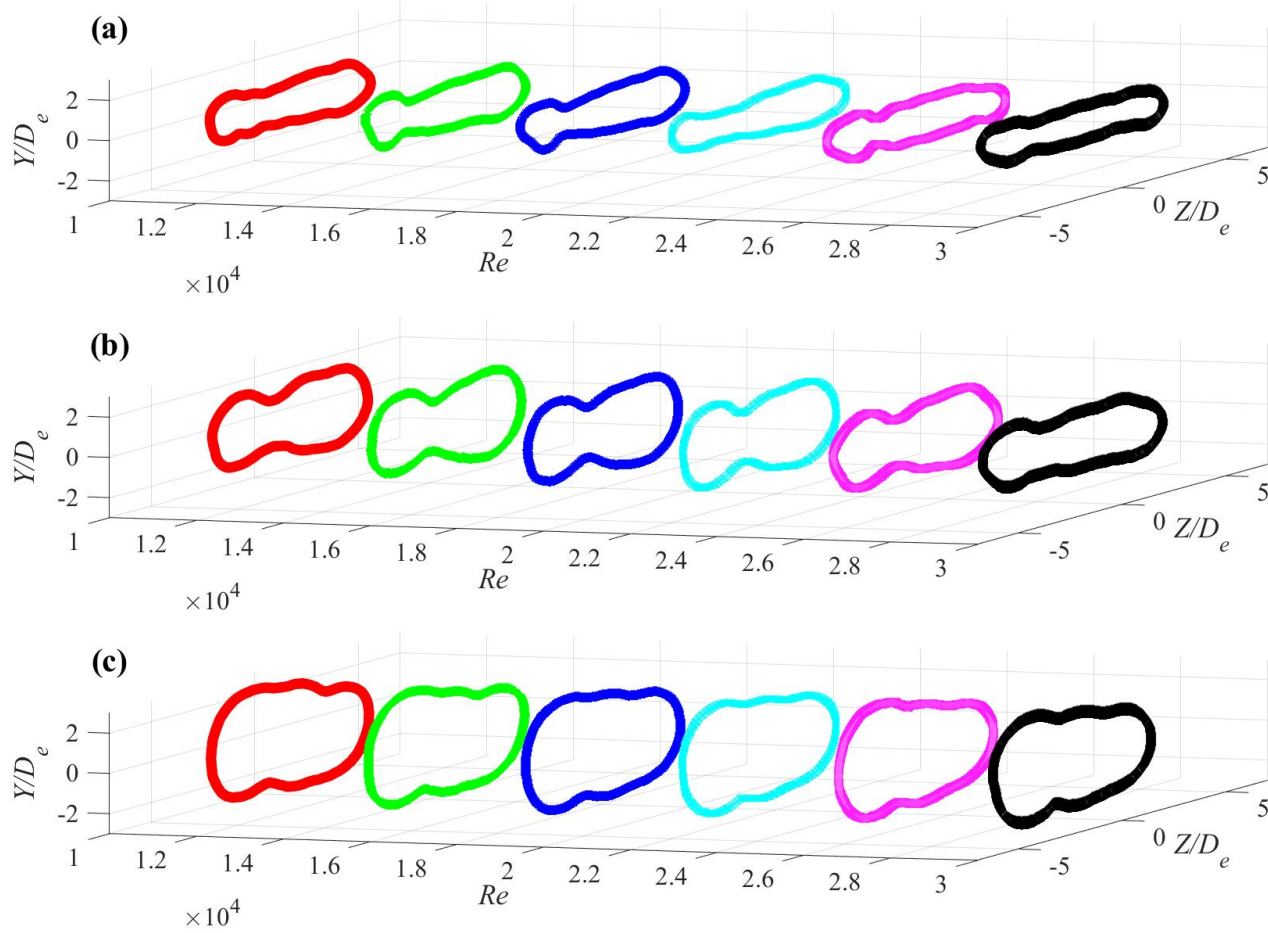


Figure E.121: Jet boundary evolution for $AR = 100$ at (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$

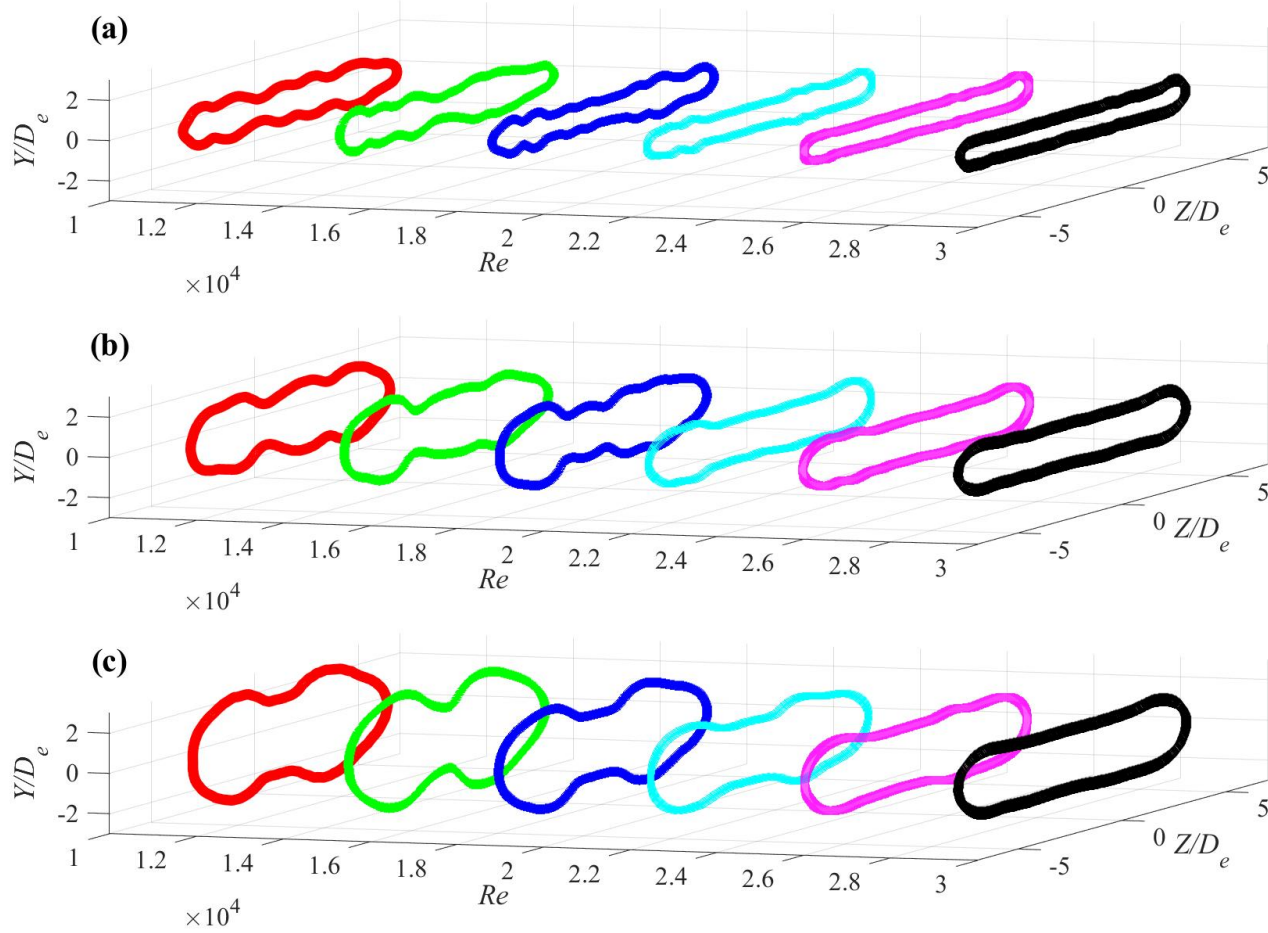


Figure E.122: Jet boundary evolution for $AR = 150$ at (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$

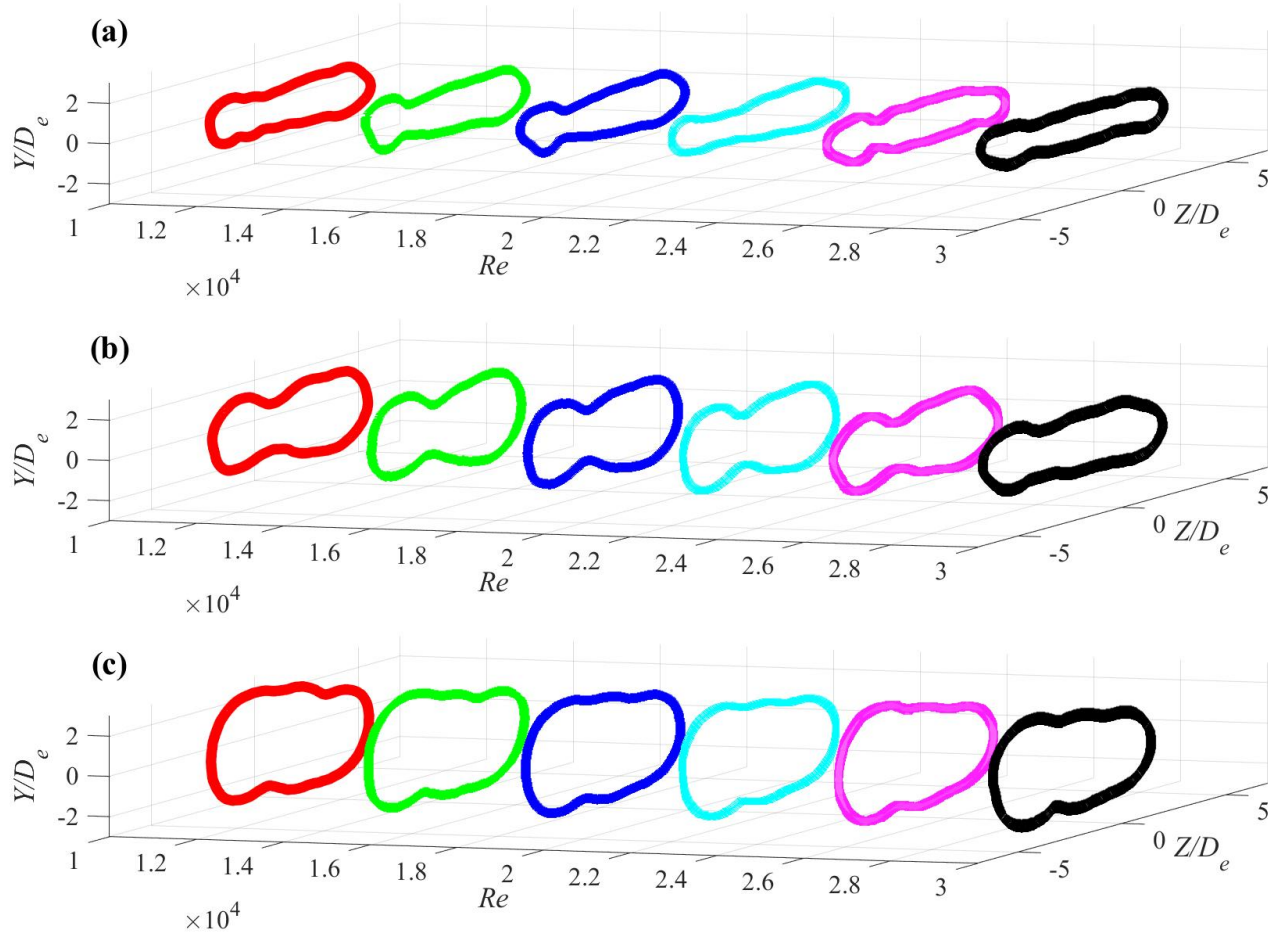


Figure E.123: Jet boundary evolution for $AR = 200$ at (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$

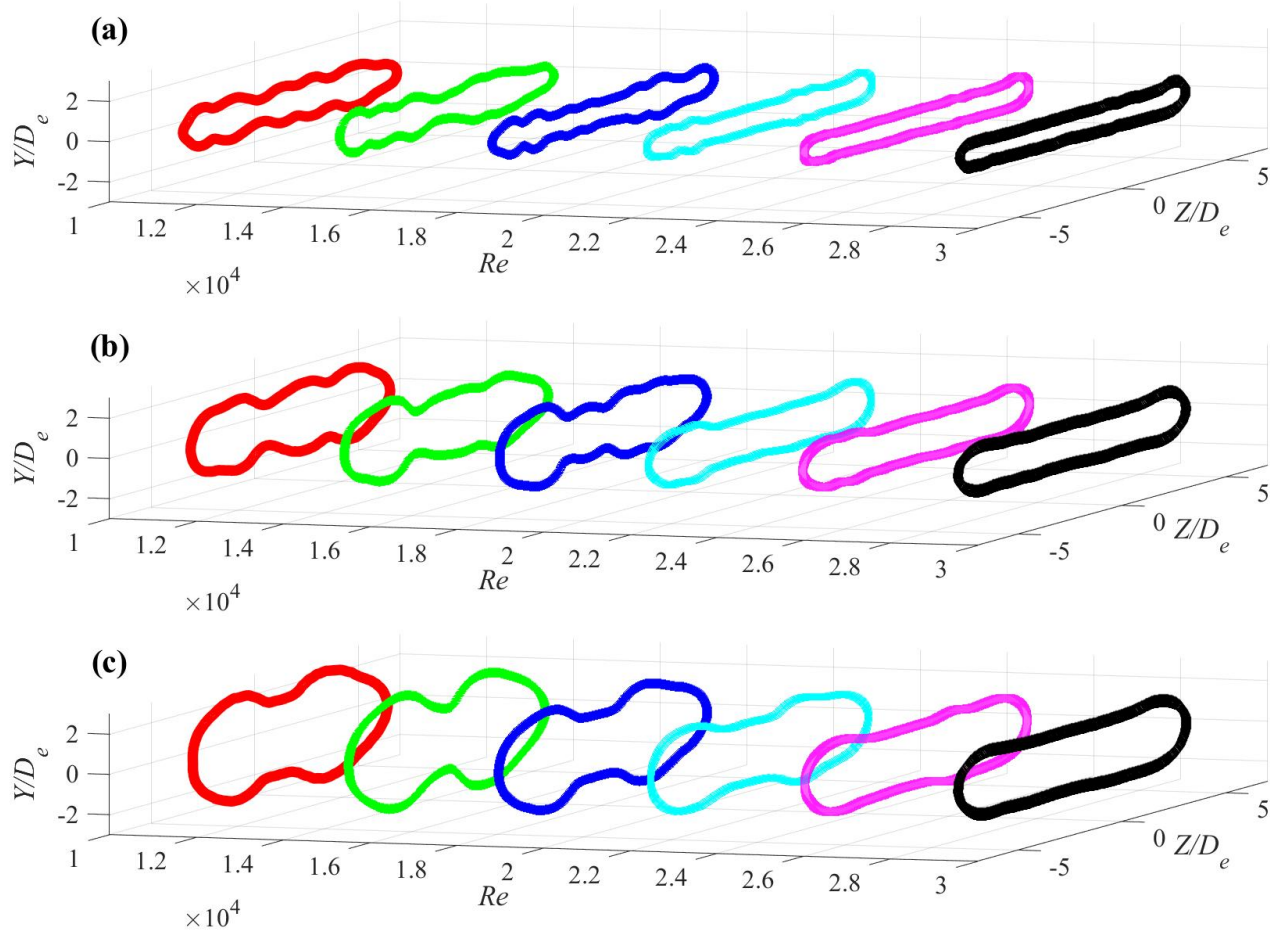


Figure E.124: Jet boundary evolution for $AR = 210$ at (a) $X = 3D_e$, (b) $X = 5D_e$, and (c) $X = 7D_e$