

High-Order Moving Overlapping Grid Methodology in a Spectral Element Method

by

Brandon E. Merrill

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2015 by the
Graduate Supervisory Committee:

Yulia Peet, Chair
Huei-Ping Huang
Eric Kostelich
Marcus Herrmann
Ronald Calhoun

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

A moving overlapping mesh methodology that achieves spectral accuracy in space and up to second-order accuracy in time is developed for solution of unsteady incompressible flow equations in three-dimensional domains. The targeted applications are in aerospace and mechanical engineering domains and involve problems in turbomachinery, rotary aircrafts, wind turbines and others. The methodology is built within the dual-session communication framework initially developed for stationary overlapping meshes. The methodology employs semi-implicit spectral element discretization of equations in each subdomain and explicit treatment of subdomain interfaces with spectrally-accurate spatial interpolation and high-order accurate temporal extrapolation, and requires few, if any, iterations, yet maintains the global accuracy and stability of the underlying flow solver. Mesh movement is enabled through the Arbitrary Lagrangian-Eulerian formulation of the governing equations, which allows for prescription of arbitrary velocity values at discrete mesh points.

The stationary and moving overlapping mesh methodologies are thoroughly validated using two- and three-dimensional benchmark problems in laminar and turbulent flows. The spatial and temporal global convergence, for both methods, is documented and is in agreement with the nominal order of accuracy of the underlying solver. Stationary overlapping mesh methodology was validated to assess the influence of long integration times and inflow-outflow global boundary conditions on the performance. In a turbulent benchmark of fully-developed turbulent pipe flow, the turbulent statistics are validated against the available data. Moving overlapping mesh simulations are validated on the problems of two-dimensional oscillating cylinder and a three-dimensional rotating sphere. The aerodynamic forces acting on these moving rigid bodies are determined, and all results are compared with published data. Scaling

tests, with both methodologies, show near linear strong scaling, even for moderately large processor counts.

The moving overlapping mesh methodology is utilized to investigate the effect of an upstream turbulent wake on a three-dimensional oscillating NACA0012 extruded airfoil. A direct numerical simulation (DNS) at Reynolds Number 44,000 is performed for steady inflow incident upon the airfoil oscillating between angle of attack 5.6° and 25° with reduced frequency $k = 0.16$. Results are contrasted with subsequent DNS of the same oscillating airfoil in a turbulent wake generated by a stationary upstream cylinder.

DEDICATION

To my beautiful, loving wife Katie whose support and encouragement motivates me to be better than I am. To my children, whose sweet hugs can get me through anything. To my parents for providing me with the foundation to accomplish great things. To all of my family for their unconditional love and wholehearted support.

ACKNOWLEDGMENTS

I would like express sincere gratitude to my advisor Dr. Yulia Peet, for her constant guidance and support throughout this journey. The methods, skills, and practices I learned under her mentorship will be invaluable in all of my future endeavors. This work would not have been possible without her expert direction.

I express appreciation to Dr. Tirupalavanam Ganesh, under whose direction I assisted in Engineering Education research during the first year of my PhD studies. His enthusiasm for S.T.E.M. education is contagious and has kindled within me that same excitement.

I am grateful to my committee for the direction they have given me in this research. I am extremely fortunate to have been taught by most of them through courses and office visits. The knowledge I have gained through their instruction is unparalleled and will be utilized again and again.

I acknowledge support from NSF grant CMMI #1250124, and give thanks to the NSF XSEDE program for providing computational resource allocations on the SDSC Gordon, SDSC Comet, and TACC Stampede clusters that have been used for scaling analyses and large dynamic stall simulations.

While I could not possibly list every individual who has helped me get where I am today, I acknowledge, and am genuinely grateful for, the never-ending support I have received from family, friends, colleagues, teachers, church leaders, and many, many more.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Literature Review	2
1.3 Overview and Accomplishments	10
2 NUMERICAL METHODOLOGY	14
2.1 Mathematical Formulation	14
2.1.1 Governing Equations	14
2.1.2 Arbitrary Lagrangian-Eulerian Formulation	16
2.2 Numerical Formulation	20
2.2.1 Global Variational Form	21
2.2.2 Spatial Discretization	23
2.2.3 One-Dimensional Operators	26
2.2.4 Extension to Multiple Dimensions	30
2.2.5 Extension to Deformed Geometries	31
2.2.6 Coupling Local Elements	32
2.2.7 Extension to Vector Space	34
2.2.8 Spatially Discretized Equations	34
2.2.9 Temporal Discretization	35
2.2.10 Neumann Boundaries	38
2.2.11 Dirichlet and Moving Wall Boundaries	38
2.2.12 Interface Conditions	39

CHAPTER	Page
2.2.13 Mesh Velocity	45
2.2.14 Iterative Solver	46
2.2.15 Fully-Discretized Equations	46
2.3 Parallel Communication	48
3 STATIONARY OVERLAPPING MESH VALIDATION	51
3.1 Convecting Walsh's Eddies	51
3.1.1 Mesh Configuration	52
3.1.2 Accuracy	53
3.2 Convecting Taylor Vortex	62
3.2.1 Accuracy	65
3.3 Turbulent Pipe Flow	68
3.3.1 Mesh Configuration	69
3.3.2 Forcing.....	71
3.3.3 Accuracy	73
4 MOVING OVERLAPPING MESH VALIDATION	79
4.1 Convecting Two-Dimensional Eddies	79
4.1.1 Simulation Configuration	80
4.1.2 Rotating Mesh Accuracy	81
4.1.3 Sliding Mesh Accuracy	83
4.1.4 Influence of Mesh Velocity	88
4.2 Two-Dimensional Oscillating Cylinder	93
4.2.1 Stationary Cylinder	95
4.2.2 Oscillating Cylinder	97
4.3 Rotating Sphere	101
5 COMPUTATIONAL TIMING AND SCALING	110

CHAPTER	Page
5.1 Stationary Overlapping Meshes	110
5.2 Moving Overlapping Meshes	115
6 DYNAMIC STALL	118
6.1 Problem Introduction	118
6.2 Validation and Comparison of 2D and 3D Simulations	121
6.2.1 Pitching-up Airfoil, $Re=80,000$	122
6.2.2 2D Oscillating Airfoil at $Re = 135,000$	125
6.2.3 2D and 3D Oscillating Airfoil at $Re = 44,000$	127
6.3 Influence of Upstream Disturbances	134
6.3.1 Problem Description and Setup	134
6.3.2 Oscillating Airfoil with Steady Inflow	137
6.3.3 Cylinder Wake Validation	144
6.3.4 Pitching Airfoil in a Turbulent Wake	150
6.3.5 Power Spectral Densities of Aerodynamic Coefficients	161
7 CONCLUSION	168
7.1 Summary	168
7.2 Future Directions	170
REFERENCES	172
APPENDIX	
A PRESSURE SOLUTION FOR WALSH'S EDDIES	186

LIST OF TABLES

Table		Page
2.1	Coefficients for the BDFk and EXTk Schemes, $k=1,2,3$	37
2.2	Coefficients for IEXTm Schemes, $m=1, 2, 3$	43
4.1	Comparison of Results for the Stationary Cylinder in Uniform Flow Simulation Presented in Section 4.2.1 with Data Presented in [1]	96
4.2	Comparison of Results for the Oscillating Cylinder in Uniform Flow Simulation Presented in Section 4.2.2 with Data Presented in [1]	98
4.3	A tabular listing of regimes for each rotating sphere simulation per- formed in the present study.	103
6.1	Airfoil Grid Parameters for Moving Airfoil Simulations in Sections 6.2.1 and 6.2.2	123
6.2	Airfoil Grid Parameters for Oscillating Airfoil Simulations Presented in Section 6.2.3	129
6.3	Pitching Airfoil Cases in the Presence of a Turbulent Wake	137
6.4	Airfoil Grid Parameters for Oscillating Airfoil Simulations Presented in Section 6.3	137
6.5	Properties of the DSV During the Third and Fourth Pitching Cycles of Case I	143
6.6	Comparison of Lift Coefficient Maxima During the Third Cycle Among All Three Cases	153
6.7	Time Averaged Values of the Aerodynamic Force and Moment Coeffi- cients	158
6.8	Properties of the Dynamic Stall Vortex During the 3 rd and 4 th Pitching Cycles	159

LIST OF FIGURES

Figure		Page
2.1	Schematic of the Global Domain and Moving Subdomains	14
2.2	Illustrative Representation of Points on Interface Boundaries Being Mapped to Reference Coordinates at Time t^n and t^{n+1}	41
2.3	Basic Procedures for Computation of Moving Overlapping Mesh Method- ology	48
2.4	Dual-Session Communication Framework for Parallel Communication Among Two Independent Sessions	49
3.1	Two-Mesh Domains for Convecting Eddies Simulations, with Element Boundaries Shown	53
3.2	Pressure Contours of the Convecting Eddies Simulation	53
3.3	Spatial Accuracy of Convecting Eddies Simulations	55
3.4	Temporal Accuracy of the Velocity for the Square Interior Mesh Con- vecting Eddies Simulation	56
3.5	Temporal Accuracy of the Pressure for the Square Interior Mesh Con- vecting Eddies Simulation	56
3.6	Spatial Accuracy for Square Interior Mesh Convecting Eddies Tests with Global Dirichlet Boundary Conditions	57
3.7	Temporal Accuracy for Square Interior Mesh Convecting Eddies Sim- ulations with Global Dirichlet Boundary Conditions	57
3.8	Contour Plots of the Absolute Error in Velocity Magnitude for Con- vecting Eddies Simulation with Global Dirichlet Boundary Conditions .	58
3.9	L^∞ Error with Respect to Element Size ($\Delta x \times \Delta x$) for Square Interior Mesh Convecting Eddies Simulations	60

Figure	Page
3.10 L^∞ Error with Respect to Average Element Size ($\Delta x \times \Delta x$) for Circular Interior Mesh Convecting Eddies Simulations	61
3.11 Cross-Section Velocity and Pressure Profiles of the Square Interior Mesh Convecting Eddies Simulation Through the Center of the Domain	62
3.12 Mesh Configuration for Convecting Taylor Vortex Simulations	63
3.13 u -Velocity Contours when the Taylor Vortex is in the Overlapping Region	65
3.14 L^2 -Errors as the Taylor Vortex Travels through the Domain	65
3.15 Pressure Errors Throughout the Domain as the Taylor Vortex Enters and Leaves the Overlapping Region	67
3.16 Pressure Error Contours when the Taylor Vortex is Leaving the Global Domain	67
3.17 L^∞ Errors with Respect to Polynomial Order and Δt Refinement in Taylor Vortex Simulations	68
3.18 Profile Plots Through the Center of the Taylor Vortex when it Resides in the Overlapping Region	69
3.19 Cross Section Views of the Meshes for Turbulent Pipe Flow	70
3.20 An Axisymmetric Cross Section Illustration of the Subdomains in Turbulent Pipe Flow	72
3.21 Modified Flow Rates of the Individual Domains, and the Global Bulk Flow Rate in a Turbulent Pipe Flow Simulation	74
3.22 Average Velocity Profile of Turbulent Pipe Flow Normalized by the Centerline Velocity	75
3.23 u_{rms}^+ Profile of Turbulent Pipe Flow	76

Figure	Page
3.24 Contours of Fluid Velocity Fluctuations in a Length-Wise Cross Section of Turbulent Pipe Flow	77
3.25 Contours of Axial Fluid Velocity Fluctuations in a Cylindrical Cross Section of Turbulent Pipe Flow	78
4.1 Velocity Magnitude Contours of Rotating Interior Mesh Convecting Eddies Simulations	82
4.2 Velocity Errors with Respect to Changing Polynomial-Order for Rotating Interior Mesh Convecting Eddies Simulations	83
4.3 Velocity Errors with Respect to Changing Timestep for Rotating Interior Mesh Convecting Eddies Simulations	84
4.4 Pressure Convergence Plots for Rotating Square Interior Mesh Convecting Eddies Simulations	85
4.5 Velocity Errors for Square Sliding Interior Mesh Convecting Eddies Simulations	86
4.6 Pressure Errors for Square Sliding Interior Mesh Convecting Eddies Tests	87
4.7 Velocity Errors with Respect to Sliding Mesh Velocity for Convecting Eddies Simulations	89
4.8 Velocity Errors with Respect to Rotating Mesh Angular Velocity for Convecting Eddies Simulations	90
4.9 Spatial and Temporal Convergence for Square Sliding Interior Mesh Convecting Eddies Simulations with Prescribed Mesh Velocity $w = 6.5$.	91
4.10 Velocity Errors in the Interior Mesh with Respect to Relative Velocity for Convecting Eddies Simulations	91
4.11 Velocity Errors in the Interior Sliding Mesh with Respect to Relative Velocity	92

Figure	Page
4.12 Mesh Geometry for Oscillating Cylinder Simulations.....	94
4.13 Enlarged Contour Plot of x -Direction Velocity Contours of an Oscil- lating Cylinder Simulation	97
4.14 Present 'Lock-In' Data Compared with Experimental Data in [2]	99
4.15 Plot of y -Direction Velocity in the Global Domain for a Cylinder Os- cillating With $A = 2.5$, $F = 1$, and $Re = 200$	100
4.16 Mesh Geometry for Transversely Rotating Sphere Simulations	102
4.17 Comparison of Force Coefficients at $Re=100$, 250 , and 300 in Rotating Sphere Simulations	106
4.18 Comparison of Strouhal Numbers Given by the Present Rotating Sphere Simulations and Data Presented in [3, 4, 5]	107
4.19 Visual Comparison of Vortices Shed from Rotating Spheres as Re- ported in [5]	109
5.1 Cross-Section of Meshes for Pipe Flow Computational Scaling Simula- tions	111
5.2 Computational Time Vs Number of CPUs for Stationary Overlapping Mesh Simulations Performed on the SDSC Gordon Cluster	112
5.3 Computational Scaling Data for Stationary Overlapping Mesh Simu- lations Performed on the TACC Stampede Cluster	113
5.4 Stationary Overlapping Mesh Timing Data for Simulations Performed on the TACC Stampede Cluster	114
5.5 Scaling of Stationary and Moving Overlapping Mesh Simulations	116
5.6 Wall-Time Spent in Different Components of the Moving Overlapping Mesh Methodology	117

Figure	Page
5.7	Percent Change in Computation Time, Comparing the Present Moving Overlapping Mesh Methodology with the Traditional Single Mesh Implementation and the Stationary Overlapping Mesh Methodology ... 117
6.1	A Cross Section of the Global Computational Domain for Steady Inflow Pitching Airfoil Simulations Described in Sections 6.2.1 and 6.2.2 122
6.2	Plot of Lift Coefficient with Respect to Time of a NACA0012 Airfoil Pitching at Reduced Pitch Rate $k=0.01$ 124
6.3	3D Spanwise Vorticity Plot of a NACA0012 Airfoil Pitching at Reduced Pitch Rate $k=0.01$ 125
6.4	Lift Coefficients for 2D Oscillating Airfoil Tests in Section 6.2.2 127
6.5	Lift Coefficient Values, as Published in [6] 128
6.6	Mean Lift Coefficient for 2D Tests in Section 6.2.2, Phase Averaged Over Three Cycles 128
6.7	A Cross-Section of the Global Computational Domain for Steady Inflow Oscillating Airfoil Simulations Described in Section 6.2.3 129
6.8	Circulatory and Non-Circulatory Lift Components as Reported in [7] ... 130
6.9	Lift Coefficients Plots from 2D Oscillating Airfoil Tests in Section 6.2.3 132
6.10	Lift Coefficient Plots from 3D Oscillating Airfoil Tests in Section 6.2.3 . 133
6.11	Comparison of Lift Coefficient Values Reported by 2D and 3D Oscillating Airfoil Simulations in Section 6.2.3 133
6.12	Oscillating Airfoil Mesh Configurations 138
6.13	Lift Coefficient Plots from Oscillating Airfoil Tests in Section 6.3.2 139
6.14	Zoomed-in Plot of Streamwise Velocity for a Cross-Section of the 3D Oscillating Airfoil Simulation in Section 6.3.2 140

Figure	Page
6.15 Plot of Spanwise Vorticity for a Cross-Section of the 3D Oscillating Airfoil Simulation in Section 6.3.2	141
6.16 Plot of Pressure Coefficient Values Along the Suction Surface of the Oscillating Airfoil at Posted Angles of Attack	142
6.17 Plot of Spanwise Vorticity for a Cross-Section of the 3D Oscillating Airfoil Simulation in Section 6.3.2	142
6.18 Cross Section of 3D Cylinder Mesh	144
6.19 Time History of Force Coefficeints on a Cylinder at $Re_D = 3900$	145
6.20 Velocity Magnitude Plot of the Cylinder Wake	145
6.21 Turbulent Wake Statistics of the Flow Behind a Cylinder in the Near Wake	147
6.22 Turbulent Wake Statistics of the Flow Behind a Cylinder in the Far Wake	148
6.23 Turbulent Wake Statistics of the Flow Behind a Cylinder in the Very Far Wake.....	149
6.24 Lift Coefficient Results Compared Among All Three Cases for the 3 rd and 4 th Cycles	150
6.25 A Comparison of Time Dependent Aerodynamic Forces and Moments Among Cases I, II, and III.....	152
6.26 A Comparison of Aerodynamic Forces and Moments Phase Averaged Over the Last Three Cycles.....	154
6.27 Plot of Streamwise Velocity for a Cross-Section of Case II.....	155
6.28 Plot of Streamwise Velocity for a Cross-Section of Case III.....	156
6.29 Plot of Spanwise Vorticity for a Cross-Section of Case II	157
6.30 Plot of Spanwise Vorticity for a Cross-Section of Case III	158

Figure	Page
6.31 Spanwise Averaged Pressure Coefficients Along the Suction Surface of the Airfoil for Cases I, II and III	159
6.32 Spanwise Averaged Spanwise Vorticity Plots When the Center of the DSV Reaches the $x/c = 1.95$ Position	160
6.33 PSD Functions of Lift Coefficients.....	162
6.34 PSD Functions of Drag Coefficients	163
6.35 PSD Functions of Pitching Moment Coefficients	164
6.36 Normalized PSD Functions of Aerodynamic Coefficients	165
6.37 Energy Partitions Derived from the Aerodynamic Coefficient PSD Func- tions	166

Chapter 1

INTRODUCTION

1.1 Motivation

Within the engineering and physical communities, there are many important problems that involve fluid flow around moving bodies, including propellers and blades on rotor- and watercraft, wind turbines, stirred reactors, maneuvering aircraft, and biological flows such as blood flow through a pumping heart. Many of these problems include complex physics, often involving turbulence interactions which require high-accuracy computational methods for reliable flow field predictions. However, accurate modeling of the fluid flow around complex moving geometries has traditionally been a challenging task. Many flow solvers that can handle such problems have limitations in the types and ranges of motion that an object may travel. Other methods must introduce additional computational schemes to maintain sufficient resolution near moving wall boundaries. In all cases, greater computational accuracy and efficiency is desirable.

Many problems require irregular inflow conditions to model realistic situations, such as upstream disturbances or incoming turbulence incident upon moving rigid bodies. While finding approximations for such problems using a single domain can be very difficult, decomposing the domain into overlapping subdomains that are allowed to move independent of one another greatly simplifies the process, and also allows for consistent grid resolution near moving wall boundaries. However, the majority of existing methods that can accommodate these problems are employed in low-order solvers and use low-order interface interpolation methods. The following literature

review sheds additional light on the many short-comings inherent in current schemes developed to handle moving rigid body problems. The development of a domain decomposition method within a high-order global method, allows for more accurate flow approximations at a lower computational cost, giving users greater flexibility over grid resolution, rigid-body motion, and global solution accuracy.

1.2 Literature Review

The present discussion will introduce previous computational developments in domain decomposition for use with stationary grids, and will then give an overview of previous methods developed for the approximation of fluid flow around moving rigid bodies, including advantages and shortcomings.

Domain Decomposition. Finding numerical solutions to partial differential equations (PDEs) by decomposing the computational domain into smaller subdomains is an idea that has been around for well over a century. Domain decomposition methods have been utilized for several different purposes, including straightforward parallelization [8, 9, 10, 11], simplified mesh generation for complex geometries [12, 13, 14, 15], and the ability to use different parameters or methods in different subdomains [16, 17, 18, 19]. These techniques exist in many forms, and each has its strengths. Some decompose the global domain into overlapping subdomains [20, 21, 22, 23, 24, 25], while others employ non-overlapping subdomains [26, 16, 14, 27, 28, 29]. Some use explicit interpolation techniques for values at interface boundaries [16, 30, 27, 31], and others carry out implicit interpolation [32, 33, 29, 34, 20, 21, 35, 36, 37]. Domain decomposition techniques have been developed for use with several numerical methods including finite difference [21, 12, 16, 30], finite element [32], finite volume [24, 38], and spectral methods [28, 29].

The earliest known research in domain decomposition methods was performed by H.A. Schwarz whose work was published in 1870 [20]. The original Schwarz Alternating Method, initially proposed for analytical calculations [39], was developed for the global solution of boundary value problems for harmonic functions [22] decomposed into overlapping subdomains, $\Omega = \Omega_1 \cup \Omega_2$. The solution in the first subdomain (Ω_1 with boundaries $\partial\Omega_1 \cap \partial\Omega$ and $\Gamma_1 = \partial\Omega_1 \setminus \partial\Omega$) is found using the global boundary conditions on $\partial\Omega_1 \cap \partial\Omega$ and corresponding values from Ω_2 at the previous iteration on Γ_1 . The solution of Ω_2 is then found by using values from the solution in Ω_1 on Γ_2 . These two steps are iterated until sufficient convergence is reached (see [40, 39, 41]).

In the 1960's, Volkov generalized the original Schwarz Alternating Method into a numerical domain decomposition technique, in a form of the *Composite Mesh Difference Method* (CMDM) [21]. CMDM used finite difference methods to solve the 2-dimensional Poisson equation numerically on overlapping grids. His research laid the foundation for subsequent techniques that extended the use of CMDM to other elliptical and hyperbolic PDEs, and boundary value and initial value problems, with the ability to use curvilinear meshes (see [42, 43, 44, 45, 13]). Overlapping domain decomposition methods have also been developed to model complex equations and handle various difficulties in solving practical problems. The Chimera Grid Scheme, introduced in [46], employs overset (overlapping) grids for modeling flows in complex geometries. Shortly after initial development it was enhanced for use with three dimensional flows modeled by the Euler equations [47] and later with the addition of the thin-layer Navier-Stokes equations. More recently, Chimera Grid techniques have been used to model various problems with complex geometries [48, 49]. Subsequently, Henshaw and Schwendeman [24, 11] developed a method for using overlapping mesh techniques in modeling high-speed reactive flows, in two and three dimensions.

In addition, techniques that employ non-overlapping grids (sometimes called patched grids) were developed. Examples include a zonal approach that uses a flux-vector splitting technique for the determination of interface values in Euler equations [50, 51, 52, 53], Lions method [26] that uses an iterative technique to arrive at the correct values to be passed between non-overlapping subdomains in solving Laplace’s equation and more general second-order elliptic problems, Dawson’s approach [16] that solves the heat equation using an explicit finite difference formula to determine the interface values and allows for different time stepping to be used in different subdomains. Non-overlapping grid techniques have also been extended and employed in solving the advection-diffusion equation [27] and the Navier-Stokes equations [52, 29]. Some of the more recently developed non-overlapping domain decomposition methods achieve very high finite global accuracy [54] and some even spectral accuracy [28, 14, 55, 29]. In [29], Manna, Vacca, and Deville describe a spectral non-overlapping domain decomposition method for solving the Navier-Stokes equations in three dimensions. The Mortar Element Method [28, 14], which was built upon the original Spectral Element Method (SEM) [56, 10, 57], adds additional flexibility to SEM by allowing the global domain to be composed of non-conforming, yet non-overlapping, elements.

While non-overlapping mesh methods allow some flexibility in mesh generation, the constraints in these techniques inhibit additional flexibility that is seen in overlapping mesh methods. By allowing variable overlap size, a broad range of potential mesh configurations are supported with overlapping methods, thus allowing for more simplified mesh generation. Additionally, overlapping methods provide a convenient framework for further extension towards moving domain methods, allowing for general and unconstrained motion of rigid body parts through the background stationary meshes [58, 59, 60, 61].

So far, existing overlapping grid methods for the time-dependent PDE coupling have been traditionally relying on low-order, finite-difference or finite-volume schemes. Although some of the methods have been extended to achieve higher-order spatial convergence, using extended stencil finite-difference or compact schemes, the upper bound of the global accuracy has been usually limited to four [30, 31, 62], and at most six [63, 64, 65, 34, 66]. Recently, Brazell, Sitaraman and Mavriplis developed a high-order overlapping Discontinuous Galerkin solver for compressible equations, that uses Lagrangian interpolation at interface boundaries, and documented a polynomial convergence up to fourth order [67].

One of the inherent challenges with overlapping grid methods is to minimize the errors that are introduced due to the coupling of the individual subdomain solutions into the global solution. The coupling errors consist of spatial errors and temporal errors, and have to be treated separately. Spatial errors are introduced by the spatial interpolation stencil employed to obtain a function value at the interface points of one domain from the gridpoint values in the adjacent domains at the same time level. Some overlapping mesh methods circumvent the spatial error by requiring that the gridpoints in overlapping domains exactly coincide [68, 69], thus fully conserving communicated information, with the drawback of decreased flexibility in mesh generation. Other methods that do not require the exact match of the gridpoints and thus are more flexible, use finite order interpolation schemes to determine values from adjacent subdomains. Although simple linear interpolation techniques have been popular [13, 16, 70, 36, 71], it was shown by Chesshire and Henshaw [13] that an interpolation scheme should be consistent with the accuracy of the underlying solver and higher-order interpolation is required to maintain the accuracy of the coupled solution with high-order methods. Thus, in fourth- and sixth- order methods [67, 30, 34, 31]

where the mesh overlap size remains constant, a generalized Lagrangian interpolation method of the matching order (fourth or sixth) was employed.

Temporal errors in domain decomposition methods employing implicit timestepping schemes occur due to the fact that the values in two separate subdomains cannot be determined simultaneously for the current timestep without the correct interface values, yet the correct interface values cannot be known without the current solution in the subdomains. Thus implicit subdomain coupling methods either write the interpolation dependencies into the global matrix [13, 33, 34], or couple independent subdomain solutions through the use of Schwarz-like iterations [20] to ensure interface value convergence [35, 36, 37, 38]. Global convergence of the variables in adjacent subdomains often requires many, sometimes hundreds [36], or even thousands [38] iterations, the number generally being dependent upon the PDE, the overlap size, and time step [72, 73]. Henshaw [74] shows that the convergence rates when solving elliptic PDE’s on overlapping subdomains can be reduced by using a multigrid solver with a smoothing algorithm near interface boundaries. However, if the coupling scheme does not introduce temporal errors larger than those of the global timestepping scheme, the global temporal accuracy will be preserved [75], particularly the interface values need not strictly match, but must merely be “consistent”. The notion of consistency has been previously used when constructing and analyzing interpolation and iteration schemes for the overlapping grid methods [37], where consistency was interpreted in a spatial sense. In [75], Peet and Fischer propose an explicit temporal interface extrapolation algorithm for the overlapping grid methods that preserves the overall temporal accuracy, and analyzed its stability. The proposed coupling scheme, which is utilized by the current methodology, is essentially “temporally consistent” (the differences between the interface values due to temporal coupling are guaranteed to be smaller than the temporal error of the underlying flow solver).

Accommodating Moving Bodies. In the ensuing review of previous techniques for approximating the flow around moving bodies, the methods can be categorized into two general classes: global-mesh methods and zonal-mesh methods (also referred to as embedded grid, Chimera, or domain decomposition methods). In global-mesh methods, a single grid is used to model the fluid flow around moving bodies, using either fixed-mesh or body-conforming mesh schemes.

Fixed-mesh methods, such as immersed boundary and fictitious domain, perform calculations on a global (non-decomposed) Eulerian mesh that does not conform to the fluid-solid interface. (Note that while many of these methods are commonly applied to fluid-fluid interface problems, the focus of this work will be on their applications to fluid-solid interface problems). The immersed boundary method, which was originally introduced by Peskin in 1972 to model the blood flow through a pumping heart [76, 77], allows for flow solutions to be computed around irregular shaped objects by tracking Lagrangian points at the interface boundary against the Eulerian (fixed) mesh where the fluid flow is calculated. The mesh within an immersed boundary solver need not conform to the fluid-solid interface, and the no-slip boundary conditions at fluid-solid interfaces are enforced by adding a body forcing term to the governing equations using either continuous or discrete forcing [78, 79]. In the similar fictitious domain method, first introduced by Glowinski et al. [80, 81, 82], the boundary conditions on the surface of rigid bodies are enforced by including distributed Lagrange multipliers in the governing equations. While several advances have been made to improve efficiency of both immersed boundary and fictitious domain methods [79], the accuracy of such methods is lacking, with few fixed-mesh methods even reaching second-order spatial accuracy [83, 84, 85, 86, 87]. In addition, resolution near interface boundaries is inconsistent as the object moves, requiring dynamic remeshing techniques such as adaptive mesh refinement (AMR) to maintain sufficient

resolution, especially when attempting to capture characteristics of boundary layer flows over moving bodies [88, 89, 90].

In contrast, body-conforming methods ensure more consistent resolution near moving boundaries, though mesh generation is usually a more complex task. As solid interfaces move, the mesh must also move and deform, thus inhibiting large displacements which would cause detrimental mesh distortions. Body-conforming methods include Deforming-Spatial-Domain/Stabilized Space-Time (DSD/SST) and Arbitrary Lagrangian-Eulerian (ALE) methods. In the DSD/SST method, as originally presented by Tezduyar et al. [91, 92], the governing equations for incompressible fluid flow, the Navier-Stokes equations, are defined using the space-time formulation, where the global time interval of the simulation is divided into subintervals called space-time slabs, and discretization is performed using interpolation functions in the four-dimensional space-time domain. The shape and orientation of the spatial domain is given by the shape of the individual space-time slabs, where solution on each slab is solved using a finite element formulation, thus treating moving boundaries throughout the global time interval [93]. The ALE method combines the Lagrangian and Eulerian formulations of the incompressible Navier-Stokes equations by changing the material derivative to account for additional convection introduced by the velocity of the moving grid points. Original developments into the combination of Eulerian and Lagrangian formulations began in the sixties [94, 95], though the standardized ALE method was developed by a group at Los Alamos National Lab [96, 97] where it was applied to finite difference methods. This ALE formulation was extended to finite element methods in the seventies and eighties [98, 99, 100, 101, 102], with later extension to the spectral element method by Ho and Patera [103, 104]. In both of these techniques large rigid body movements cause large mesh distortions and possibly entanglement of the computational grid. To remedy this problem, many

methods remesh the global domain when distortions become large, though remeshing is a computationally expensive process [79]. Another possible remedy is to allow the global mesh to move with the body in a rigid-body type of motion, but this creates challenges for the enforcement of boundary conditions at moving global boundaries.

Decomposing the global domain into a collection of subdomains, as in the class of zonal mesh methods, would allow for large rigid body movements of one domain within another, avoiding the need for remeshing and replacing global boundary conditions on moving domains with local interface conditions. Among this class, for example, are sliding mesh methods, which decompose the domain into *non*-overlapping subdomains, thus greatly restricting the types of grid motion that can be performed. While sliding mesh solvers can effectively perform simulations involving rotating geometries, they cannot be applied to important problems involving translational or deforming motion. Several variations of these solvers are commonly used for simulations of stirred reactors [105, 106, 107, 108, 109] and propellers/blades on rotorcraft [110, 111, 112, 113]. Sliding mesh methods are typically coupled with finite volume methods, although other solvers are also used, such as finite element methods [114, 109]. Global spatial accuracy for sliding mesh methods, which is most often limited by the accuracy of the interpolation of values among subdomains, reaches up to 2^{nd} -order [111, 113].

On the other hand, overlapping mesh methods allow for arbitrary rigid body motions, and do not have the same movement restrictions as sliding mesh solvers since constraints on mesh alignment are alleviated. While overlapping mesh methods are commonly used for rotating machinery simulations as well [115, 116, 60, 117, 118, 119], they have additionally been applied to simulate flows involving other types of body motions. These include flows around moving cylinders [61, 120], piston-driven flows [61, 121], biological flows such as flows through moving valves and hearts [122,

123], aerodynamic flows such as three element airfoils [59], plunging wings [120], and store separation from a wing [124, 125, 126, 59, 127], as well as flows around moving ships [128, 129].

Moving overlapping mesh methods are based on overlapping domain decomposition methodologies which were discussed previously. While there are few techniques among stationary overlapping grid methods which exceed fourth-order global accuracy [34, 130, 65], and the situation is even more challenging for moving overlapping mesh methods. The literature search did not result in any methods with *verified* global accuracy higher than the second-order [61]. While some methods use higher-order (fourth or fifth) integration schemes within individual subdomains [122, 129], trilinear interface interpolation is usually employed, which restricts the *global* accuracy to second.

1.3 Overview and Accomplishments

In this dissertation, a spectrally-accurate moving overlapping mesh method is introduced for the incompressible fluid flow around moving rigid bodies, that is based on a spectral-element method using the ALE formulation of the governing equations. The Spectral Element Method, which can be perceived as a high-order extension of the Finite Element Method, divides a domain into several conforming and adjacent elements [56, 10, 57]. The volume within each element is discretized using N^{th} -order tensor-product Lagrange interpolating polynomials on Gauss-Lobatto-Legendre nodal points. Approximations in all elements are coupled at the boundaries to form a global solution [57], which achieves spectral convergence with p - (polynomial order) refinement.

The moving overlapping mesh methodology utilizes N^{th} -order Lagrangian interpolation, on non-uniform Gauss-Lobatto-Legendre grids, to arrive at a spectrally-

accurate (with p -refinement) interpolation scheme that allows us to maintain the global spectral accuracy of the coupled solution. The method also achieves high-order spatial convergence with h - (grid) refinement, where the order of accuracy is equal to the polynomial order (N). The explicit temporal extrapolation coupling scheme is used for coupling the solutions of incompressible Navier-Stokes equations on moving overlapping grids, and is consistent with the global temporal accuracy of the SEM solver. This novel scheme achieves a specified order of accuracy with a small number of iterations. It ensures that the temporal error between the values in the subdomains at the interfaces is equal to the temporal accuracy of the underlying SEM solver. Assuming that this holds true, the interface values need not be an *exact* match, while still maintaining the global accuracy of the solution. While the accuracy of the extrapolation scheme is maintained in the absence of iterations, the stability of the formulation can generally be improved by implementing a low number of Schwarz-like iterations [75].

While the underlying ALE and multidomain coupling formulation is flexible enough to handle any arbitrary prescribed body motion, including deforming movements, such as, for example, in biolocomotion, this dissertation concentrates on prescribed solid motion as a rigid body, thus undermining the need for mesh deformation. However, the methodology will be described in a general sense, which would allow one to reproduce it, if needed, for any arbitrary motions.

This moving overlapping mesh methodology can have important implications in the advancement of aerospace, mechanical and other branches of engineering. In this dissertation, the methodology is extended to pressing problems in aerodynamics by investigating of the influence of the upstream disturbances on the flow around pitching airfoils. This problem has direct relevance to aircraft and helicopter aerodynamics, the efficiency of wind turbine energy production, and the design of other rotating

machinery. Research regarding pitching airfoils typically focuses on the effects that pitching frequency, pitching amplitude, Reynolds number, and Mach number, have on the aerodynamics of the flow, yet, unsteady inflow conditions can also affect the aerodynamics of pitching airfoils in a critical manner. For example, turbulence, gusts, and vortices caused by upstream structures, objects, or atmospheric conditions create unsteady inflow conditions for helicopter rotors, wind turbine blades, and wings on maneuvering aircraft, that can greatly affect the aerodynamic forces.

The present work describes the following specific achievements I have made throughout the progression of my research:

- Advanced the development of the overlapping mesh methodology for stationary meshes
- Performed tests for the validation of the stationary overlapping mesh methodology
- Extended development of the overlapping mesh methodology for use with moving subdomains
- Validated the moving overlapping mesh methodology
- Ensured adequate parallel scalability and timing data of both methodologies
- Performed 2D and 3D simulations of dynamically pitching airfoils
- Compared results of dynamically pitching airfoil simulations with each other and with published experimental and numerical data
- Investigated the effects of upstream disturbances on dynamically pitching airfoils

The remainder of the paper will be structured as follows. In Chapter 2, the methodology for solving fluid flow problems on moving overlapping subdomains is put forth, including the mathematical derivation and numerical discretization of governing equations for fluid flow on subdomains using the ALE framework. In Chapters 3 and 4, results are presented for two- and three-dimensional validation simulations, in stationary and moving overlapping mesh tests, respectively, then Chapter 5 gives results from the parallel scalability and timing study. Chapter 6 utilizes the moving overlapping mesh methodology to analyze the effects of upstream disturbances on a dynamically pitching three-dimensional airfoil. A summary and description of future work is described in Chapter 7.

NUMERICAL METHODOLOGY

2.1 Mathematical Formulation

2.1.1 Governing Equations

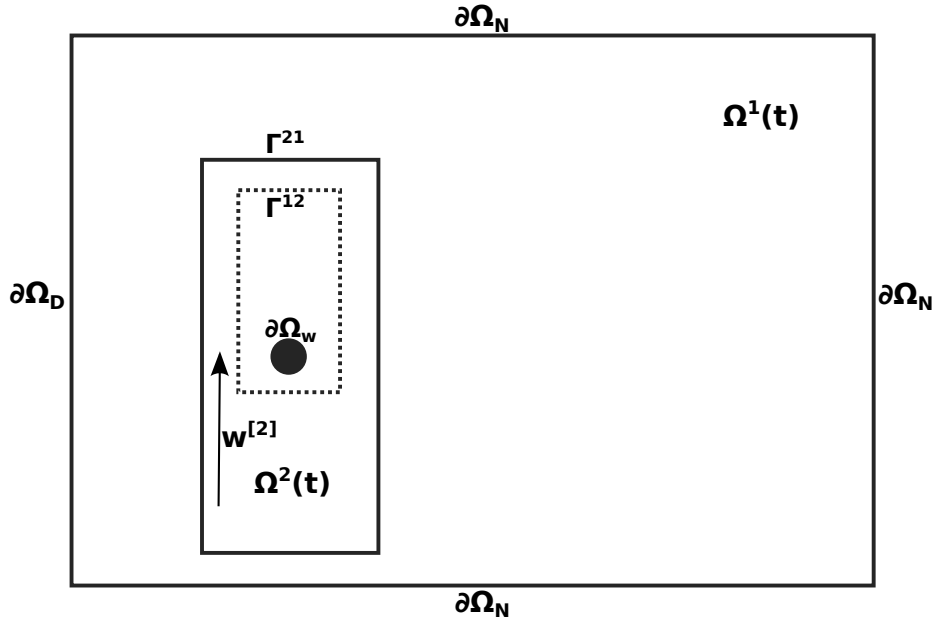


Figure 2.1: Schematic of the global domain and moving subdomains. The dashed line denotes a hole cut in the subdomain $\Omega^1(t)$, which is covered by $\Omega^2(t)$.

In the current methodology, a two or three dimensional global solution domain is decomposed into two overlapping subdomains, $\Omega^g(t) = \Omega^1(t) \cup \Omega^2(t)$, which may be time dependent and allowed to move with velocity $\mathbf{w}^{[i]}(\mathbf{x}, t)$, an example of which is seen in Figure 2.1. For the sake of generality, the methodology is formulated as if both domains are moving, assuming that a stationary domain has $\mathbf{w}^{[i]}(\mathbf{x}, t) = 0$. In this dissertation, subdomain movement is implemented as rigid bodies, although the Cartesian velocity $\mathbf{w}(\mathbf{x}, t)$ can still be a function of \mathbf{x} , for example in the case

of rotation. The fluid motion in each subdomain is governed by the incompressible Navier-Stokes equations which are represented in non-dimensional form in space R^d , as shown below

$$\Omega^1(t) \begin{cases} \frac{D \mathbf{u}^{[1]}}{Dt} = -\nabla p^{[1]} + \frac{1}{Re} \nabla^2 \mathbf{u}^{[1]} \\ \nabla \cdot \mathbf{u}^{[1]} = 0 \end{cases} \quad (2.1)$$

and

$$\Omega^2(t) \begin{cases} \frac{D \mathbf{u}^{[2]}}{Dt} = -\nabla p^{[2]} + \frac{1}{Re} \nabla^2 \mathbf{u}^{[2]} \\ \nabla \cdot \mathbf{u}^{[2]} = 0 \end{cases} \quad (2.2)$$

where \mathbf{u} is the velocity vector, p is the pressure, D/Dt is the material derivative, and the Reynolds number, $Re = U L/\nu$, is based on a characteristic velocity, length scale and kinematic viscosity. The bracketed superscript signifies the corresponding subdomain.

The global domain boundary is defined $\partial\Omega^g = \partial(\Omega^1 \cup \Omega^2)$. Outflow boundaries, as presented on the right side of Figure 2.1, are described with Neumann conditions, $\partial\Omega_N^g(t)$, and inflow boundaries (left side of Figure 2.1) are described with Dirichlet conditions, $\partial\Omega_D^g(t)$. Local subdomain boundaries that are not also part of the global boundary are termed interface boundaries, $\Gamma^{ij} \equiv \partial\Omega^i(t) \setminus \partial\Omega^g(t)$. Conditions at moving solid wall boundaries are defined within a local subdomain, $\partial\Omega_W^i(t)$. The generalized initial and boundary conditions for each subdomain are presented below

$$\mathbf{u}^{[1],[2]}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega^{1,2}(0) \quad (2.3a)$$

$$\mathbf{u}^{[1],[2]}(\mathbf{x}, t) = \mathbf{u}_d(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega_D^g(t) \quad (2.3b)$$

$$\nabla \mathbf{u}^{[1],[2]}(\mathbf{x}, t) \cdot \hat{\mathbf{n}} = 0, \quad \mathbf{x} \in \partial\Omega_N^g(t) \quad (2.3c)$$

$$\mathbf{u}^{[1],[2]}(\mathbf{x}, t) = \mathbf{w}^{[1],[2]}(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega_W^{1,2}(t) \quad (2.3d)$$

$$\mathbf{u}^{[1]}(\mathbf{x}, t) = \mathbf{u}^{[2]}(\mathbf{x}, t) \quad \mathbf{x} \in \Gamma^{12} \parallel \Gamma^{21} \quad (2.3e)$$

with \mathbf{n} the unit outward pointing surface normal.

2.1.2 Arbitrary Lagrangian-Eulerian Formulation

A fluid matter in computational physics is typically treated as a continuum and its motion governed by the Navier-Stokes Equations (2.1) and (2.2). In the Lagrangian formulation, the observer follows material particles as they move, while in the Eulerian formulation observation is done in the laboratory frame, where coordinates remain fixed. From a computational standpoint, the mesh used in the Lagrangian formulation follows the fluid particles, as in the computation of free surface flows, for example. Although difficulties arise when severe mesh deformations occur, especially in regions with large velocity gradients and reverse flow zones. In the Eulerian formulation, the mesh remains fixed, allowing for large fluid deformations, though complexities arise in the computation of moving body problems.

In the current methodology, a well known hybrid approach is employed, called the Arbitrary Lagrangian-Eulerian (ALE) formulation, which decouples the mesh motion from the fluid motion and formulates the equations of fluid motion in a coordinate system moving with the computational mesh. In what follows, a brief description of the idea is presented [102, 57, 131].

The position of a material point, or particle, within a continuous media is expressed in terms of *material coordinates*,

$$\mathbf{X} = (X_1, X_2, X_3). \quad (2.4)$$

that are related to the initial positions of the fluid particles within the media.

On the other hand, within the laboratory frame *spatial coordinates* are fixed in time

$$\mathbf{x} = (x_1, x_2, x_3), \quad (2.5)$$

and the relationship between material and spatial coordinates can be expressed in terms of a mapping that returns the spatial coordinates of a particle at a specified time when given its material coordinates

$$x_i = \mathcal{F}(X_i, t). \quad (2.6)$$

The inverse of this mapping returns the material coordinates of a particle, given its spatial coordinates

$$X_i = \mathcal{F}^{-1}(x_i, t). \quad (2.7)$$

The ALE formulation introduces a third coordinate system, also dependent on space and time, which we will call the *reference coordinate* system, χ . This coordinate system is defined for a reference domain which is representative of the computational mesh and is allowed to move in an arbitrary manner. A mapping can be defined that returns the reference coordinates of a particle when given its spatial coordinates

$$\chi_i = \mathcal{G}(x_i, t), \quad (2.8)$$

with inverse mappings also defined.

Derivatives with respect to time of a physical quantity related to the flow can be expressed in terms of any of the coordinate systems defined above. The derivative with respect to time in the material frame (often called the Lagrangian or material derivative) of some physical quantity $f(x_i, t)$, is defined

$$\left. \frac{\partial f}{\partial t} \right|_{X_i}, \quad (2.9)$$

where the derivative is taken with X_i held constant. Derivatives with respect to time in the laboratory frame (Eulerian derivative)

$$\left. \frac{\partial f}{\partial t} \right|_{x_i} \quad (2.10)$$

and derivatives with respect to time in a reference frame (ALE derivative)

$$\left. \frac{\partial f}{\partial t} \right|_{x_i} \quad (2.11)$$

are likewise defined. Thus the material velocity of a particle is given by

$$u_i = \left. \frac{\partial x_i}{\partial t} \right|_{X_i},$$

or in vector form

$$\mathbf{u} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{x}}. \quad (2.12)$$

Due to its ease of handling large fluid distortions, fluid flow is typically expressed using the Eulerian formulation, where quantities in the conservation equations (2.1) and (2.2) are given in terms of the laboratory frame, using coordinates \mathbf{x} . Thus, convective terms arise to account for the motion of the particles relative to the laboratory frame [131]. Thus, the material derivative can be expressed using the multivariate chain rule to give

$$\begin{aligned} \left. \frac{\partial f(x_i, t)}{\partial t} \right|_{X_i} &= \left. \frac{\partial f}{\partial t} \right|_{x_i} + \frac{\partial f}{\partial x_i} \left. \frac{\partial x_i}{\partial t} \right|_{X_i} \\ &= \left. \frac{\partial f}{\partial t} \right|_{x_i} + \frac{\partial f}{\partial x_i} u_i \end{aligned} \quad (2.13)$$

or in vector form

$$\begin{aligned} \frac{Df}{Dt} &= \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + \frac{\partial f}{\partial \mathbf{x}} \cdot \mathbf{u} \\ &= \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + \mathbf{u} \cdot \nabla f \end{aligned} \quad (2.14)$$

where $\frac{D}{Dt}$ is introduced to denote a material (or Lagrangian) derivative, as expressed in (2.1) and (2.2). Note that previous equations can be generalized by replacing the scalar function f with some vector function [57].

In a similar manner the mesh velocity, \mathbf{w} , is defined using the reference domain

$$w_i = \left. \frac{\partial x_i}{\partial t} \right|_{\chi_i},$$

or

$$\mathbf{w} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\boldsymbol{\chi}}, \quad (2.15)$$

and an ALE derivative is likewise formulated to account for convective terms that arise due to the relative motion between the reference and laboratory coordinate systems

$$\begin{aligned} \left. \frac{\partial f(x_i, t)}{\partial t} \right|_{\chi_i} &= \left. \frac{\partial f}{\partial t} \right|_{x_i} + \frac{\partial f}{\partial x_i} \left. \frac{\partial x_i}{\partial t} \right|_{\chi_i} \\ &= \left. \frac{\partial f}{\partial t} \right|_{x_i} + \frac{\partial f}{\partial x_i} w_i \end{aligned} \quad (2.16)$$

or expressed in vector format

$$\begin{aligned} \frac{\delta f}{\delta t} &= \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + \frac{\partial f}{\partial \mathbf{x}} \cdot \mathbf{w} \\ &= \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + \mathbf{w} \cdot \nabla f \end{aligned} \quad (2.17)$$

where $\frac{\delta}{\delta t}$ is introduced to denote an ALE derivative.

The material derivative can now be formulated for use with the reference domain by expressing (2.14) in terms of (2.17) [57]

$$\begin{aligned} \frac{Df}{Dt} &= \frac{\delta f}{\delta t} - \mathbf{w} \cdot \nabla f + \mathbf{u} \cdot \nabla f \\ &= \frac{\delta f}{\delta t} + (\mathbf{u} - \mathbf{w}) \cdot \nabla f. \end{aligned} \quad (2.18)$$

We see that if $\mathbf{w} = 0$, the traditional formulation of the material derivative is recovered as used in the Eulerian description, and if $\mathbf{w} = \mathbf{u}$ the convective terms vanish leaving only a partial derivative holding $\boldsymbol{\chi}$ constant as used in the Lagrangian description.

Equation (2.18) can be generalized by replacing the scalar function f with a vector function, such as velocity, \mathbf{u} [57]. It is used in the Navier-Stokes equations (2.1) and (2.2) for moving domain fluid flow problems, which thus become for the decomposed global domain

$$\Omega^1(t) \begin{cases} \frac{\delta \mathbf{u}^{[1]}}{\delta t} + (\mathbf{u}^{[1]} - \mathbf{w}^{[1]}) \cdot \nabla \mathbf{u}^{[1]} = -\nabla p^{[1]} + \frac{1}{Re} \nabla^2 \mathbf{u}^{[1]} \\ \nabla \cdot \mathbf{u}^{[1]} = 0 \end{cases} \quad (2.19)$$

and

$$\Omega^2(t) \begin{cases} \frac{\delta \mathbf{u}^{[2]}}{\delta t} + (\mathbf{u}^{[2]} - \mathbf{w}^{[2]}) \cdot \nabla \mathbf{u}^{[2]} = -\nabla p^{[2]} + \frac{1}{Re} \nabla^2 \mathbf{u}^{[2]} \\ \nabla \cdot \mathbf{u}^{[2]} = 0 \end{cases} \quad (2.20)$$

where initial and boundary conditions as given in (2.3) remain unchanged. We see that the governing equations are now formulated with respect to the moving mesh, which allows for convenient numerical integration of equations of motion with respect to moving domains.

2.2 Numerical Formulation

Equations (2.19) and (2.20) in moving overlapping subdomains are coupled through the interface conditions given in (2.3e). In the numerical solution of these equations, the values at interfaces are determined through the explicit interpolation from the solution in the adjacent subdomain at several previous time instances, thus obtaining the interface value, $\mathbf{u}_{int}^{[j]}(\mathbf{x}, t)$. This value essentially serves as a boundary condition, $\mathbf{u}^{[i]}(\mathbf{x}, t) = \mathbf{u}_{int}^{[j]}(\mathbf{x}, t)$, for the subdomain Ω^i on the boundary Γ^{ij} at time t . Thus, solutions of (2.19) and (2.20) can be computed independently after interface values are exchanged.

Hence, the numerical formulation within each subdomain is identical. In the following sections, the subdomain superscripts ($\Omega^i \rightarrow \Omega$, $\mathbf{u}^{[i]} \rightarrow \mathbf{u}$, $\mathbf{w}^{[i]} \rightarrow \mathbf{w}$, $p^{[i]} \rightarrow p$) are omitted, except where additional clarity is needed.

2.2.1 Global Variational Form

The ALE formulation of the governing equations, (2.19) and (2.20), is cast into variational form by multiplying each equation with a test function $\mathbf{v}(\mathbf{x}, t) \in \mathcal{H}_0^1(\Omega(t))$, and then integrating over the whole domain. Note that both the test function, $\mathbf{v}(\mathbf{x}, t)$ and the domain $\Omega(t)$ are dependent on time due to the moving reference frame. The variational form of the problem becomes: Find $\mathbf{u}(\mathbf{x}, t) \in \mathcal{H}_b^1(\Omega(t))$, $p(\mathbf{x}, t) \in \mathcal{L}^2(\Omega(t))$ such that

$$\begin{aligned} \left(\frac{\delta \mathbf{u}}{\delta t}, \mathbf{v} \right) + ((\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u}, \mathbf{v}) \\ + (\nabla p, \mathbf{v}) - \frac{1}{Re} (\nabla^2 \mathbf{u}, \mathbf{v}) = 0 \quad \forall \mathbf{v}(\mathbf{x}, t) \in \mathcal{H}_0^1(\Omega(t)) \\ -(q, \nabla \cdot \mathbf{u}) = 0 \quad \forall q(\mathbf{x}, t) \in \mathcal{L}^2(\Omega(t)), \end{aligned} \quad (2.21)$$

where the inner product of two scalar functions a and b is defined

$$(a, b) = \int_{\Omega(t)} a(\mathbf{x}, t) b(\mathbf{x}, t) dV, \quad \forall a, b \in \mathcal{L}^2(\Omega(t)), \quad (2.22)$$

and where $\mathcal{L}^2(\Omega(t))$ represents the space of square-integrable functions, and $\mathcal{H}^1(\Omega(t))$ represents the space of square-integrable functions whose first derivatives are square-integrable as well. $\mathcal{H}_0^1(\Omega(t))$ is the subspace of $\mathcal{H}^1(\Omega(t))$ with $\mathbf{v}(\mathbf{x}, t) = 0$ on Dirichlet, $\partial\Omega_D(t)$, and interface, Γ^{ij} , boundaries, while $\mathcal{H}_b^1(\Omega(t))$ is the subspace of $\mathcal{H}^1(\Omega(t))$ with $\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_d(\mathbf{x}, t)$ on $\partial\Omega_D(t)$, $\mathbf{u}(\mathbf{x}, t) = \mathbf{w}(\mathbf{x}, t)$ on $\partial\Omega_W(t)$, or $\mathbf{u}^{[i]}(\mathbf{x}, t) = \mathbf{u}_{int}^{[j]}(\mathbf{x}, t)$ on Γ^{ij} .

Integration by parts gives

$$(\nabla^2 \mathbf{u}, \mathbf{v}) \rightarrow \int_{\partial\Omega(t)} \mathbf{v} \nabla \mathbf{u} \cdot \hat{\mathbf{n}} dA - \int_{\Omega(t)} \nabla \mathbf{u} : \nabla \mathbf{v} dV. \quad (2.23)$$

where “ \cdot ” denotes a Frobenius inner product. The first integral on the right-hand side goes to zero since $\mathbf{v}(\mathbf{x}, t) = 0$ on the boundaries, thus the weak form of (2.21) is given by performing integration by parts on the terms $-\frac{1}{Re}(\nabla^2 \mathbf{u}, \mathbf{v})$ and $(\nabla p, \mathbf{v})$, giving $\frac{1}{Re}(\nabla \mathbf{u}, \nabla \mathbf{v})$ and $-(p, \nabla \cdot \mathbf{v})$ for those terms respectively.

Although test functions, $\mathbf{v}(\mathbf{x}, t)$ are time dependent, they are defined in the ALE formulation to satisfy the following property [57]

$$\frac{\delta \mathbf{v}}{\delta t} = 0. \quad (2.24)$$

In addition, the integrand of the unsteady term, $\delta \mathbf{u} / \delta t \cdot \mathbf{v}$, will give rise to inconsistencies upon temporal discretization unless the time derivative is taken outside of the integral [132]. This is done by first performing integration by parts on the unsteady term

$$\left(\frac{\delta \mathbf{u}}{\delta t}, \mathbf{v} \right) \rightarrow \int_{\Omega(t)} \frac{\delta \mathbf{u}}{\delta t} \cdot \mathbf{v} \, dV = \int_{\Omega(t)} \left[\frac{\delta (\mathbf{u} \cdot \mathbf{v})}{\delta t} - \mathbf{u} \cdot \frac{\delta \mathbf{v}}{\delta t} \right] dV, \quad (2.25)$$

where the second term in the integrand on the right hand side goes to zero because of the property defined in (2.24).

The Reynolds transport theorem for the time evolution of a volume integral with respect to the reference domain as derived by Truesdell and Toupin [132] (see also [57]), is given as

$$\frac{\delta}{\delta t} \int_{\Omega(t)} f \, dV = \int_{\Omega(t)} \left(\frac{\delta f}{\delta t} + f \nabla \cdot \mathbf{w} \right) dV. \quad (2.26)$$

This allows (2.25) to be expressed as

$$\int_{\Omega(t)} \frac{\delta (\mathbf{u} \cdot \mathbf{v})}{\delta t} dV = \frac{\delta}{\delta t} \int_{\Omega(t)} \mathbf{u} \cdot \mathbf{v} dV - \int_{\Omega(t)} (\nabla \cdot \mathbf{w}) \mathbf{u} \cdot \mathbf{v} dV. \quad (2.27)$$

Combining (2.27) with the convective terms, the first two inner products in (2.21) take the explicit form

$$\frac{\delta}{\delta t} \int_{\Omega(t)} \mathbf{u} \cdot \mathbf{v} dV + \int_{\Omega(t)} [(\mathbf{u} \cdot \nabla) \mathbf{u} - (\nabla \cdot \mathbf{w}) \mathbf{u} - (\mathbf{w} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} dV, \quad (2.28)$$

in which the convective terms can be expressed in compact form by utilizing the incompressibility constraint, $\nabla \cdot \mathbf{u} = 0$

$$\frac{\delta}{\delta t} \int_{\Omega(t)} \mathbf{v} \cdot \mathbf{u} \, dV + \int_{\Omega(t)} \mathbf{v} \cdot (\nabla \cdot [\mathbf{u}\mathbf{u} - \mathbf{u}\mathbf{w}]) \, dV. \quad (2.29)$$

Thus, the weak variational form of the global Navier-Stokes equations in the ALE formulation becomes

$$\begin{aligned} \frac{\delta}{\delta t} (\mathbf{u}, \mathbf{v}) + (\nabla \cdot [\mathbf{u}\mathbf{u} - \mathbf{u}\mathbf{w}], \mathbf{v}) \\ - (p, \nabla \cdot \mathbf{v}) + \frac{1}{Re} (\nabla \mathbf{u}, \nabla \mathbf{v}) = 0 \quad \forall \mathbf{v}(\mathbf{x}, t) \in \mathcal{H}_0^1(\Omega(t)) \\ - (q, \nabla \cdot \mathbf{u}) = 0 \quad \forall q(\mathbf{x}, t) \in \mathcal{L}^2(\Omega(t)), \end{aligned} \quad (2.30)$$

2.2.2 Spatial Discretization

In the present implementation of the spectral element method, a moving domain, $\Omega(t)$, is decomposed into several conforming elements

$$\Omega(t) = \sum_{k=1}^E \Omega_k(t), \quad (2.31)$$

where global inner products are expressed

$$(a, b)|_{\Omega(t)} = \sum_{k=1}^E (a_k, b_k)|_{\Omega_k(t)}. \quad (2.32)$$

The governing equations (2.30) are spatially discretized within each element by defining finite dimensional subspaces, $X^N \subset \mathcal{H}^1(\Omega(t))$ and $Y^N \subset \mathcal{L}^2(\Omega(t))$, onto which the search spaces, $\mathcal{H}^1(\Omega(t))$ and $\mathcal{L}^2(\Omega(t))$, are projected. The subspaces are defined

$$\begin{aligned} X^N &= \mathcal{H}^1(\Omega(t)) \cap \mathbb{P}_N \\ Y^N &= \mathcal{L}^2(\Omega(t)) \cap \mathbb{P}_{N-2}, \end{aligned} \quad (2.33)$$

with \mathbb{P}_N the space comprised of all N^{th} or lower-order polynomials.

A function, independent of time, exists for each element Ω_k that maps it from the ALE frame with reference coordinates $\boldsymbol{\chi} \in \Omega_k$, to a d-dimensional primary element, $\hat{\Omega} = [-1, +1]^d$, with primary coordinates, $\mathbf{r} \in \hat{\Omega}$:

$$\mathbf{r} = \widetilde{\mathcal{M}}_k(\boldsymbol{\chi}) \quad (2.34)$$

with the inverse

$$\boldsymbol{\chi} = \widetilde{\mathcal{M}}_k^{-1}(\mathbf{r}). \quad (2.35)$$

Recall that the time dependent function, $\mathcal{G}(\mathbf{x}, t)$, defined in (2.8), and its inverse, determine reference coordinates, $\boldsymbol{\chi}$, from spatial coordinates, \mathbf{x} , and vice versa. Thus, to map element k from its geometry and orientation in the laboratory frame to a d-dimensional primary element a double mapping must take place

$$\mathbf{r} = \widetilde{\mathcal{M}}_k(\mathcal{G}(\mathbf{x}, t)), \quad (2.36)$$

where the inverse returns the element to the laboratory frame

$$\mathbf{x} = \mathcal{G}^{-1}(\widetilde{\mathcal{M}}_k^{-1}(\mathbf{r}, t), t). \quad (2.37)$$

However, from a practical computational perspective, a time dependent mapping determines the primary element directly from the element in the laboratory frame with a function $\mathcal{M}_k(\mathbf{x}, t)$, where the tilde has been dropped

$$\mathbf{r} = \mathcal{M}_k(\mathbf{x}, t) \equiv \widetilde{\mathcal{M}}_k(\mathcal{G}(\mathbf{x}, t)) \quad (2.38)$$

and vice-versa

$$\mathbf{x} = \mathcal{M}_k^{-1}(\mathbf{r}, t) \equiv \mathcal{G}^{-1}(\widetilde{\mathcal{M}}_k^{-1}(\mathbf{r}, t), t). \quad (2.39)$$

Collocation points within the primary element are defined with Gauss-Lobatto Legendre (GL) quadrature in the velocity space, $\xi_j \in [-1, +1]$, and Gauss Legendre

(G) quadrature used to define nodal points in the pressure space, $\eta_j \in]-1, +1[$. Thus posing the problem (2.30) with finite-dimensional subspaces: Find $\mathbf{u}(\mathbf{r}, t) \in X_b^N$, $p(\mathbf{r}, t) \in Y^N$ such that

$$\begin{aligned} \frac{\delta}{\delta t}(\mathbf{u}, \mathbf{v})_{GL} + (\nabla \cdot [\mathbf{u}\mathbf{u} - \mathbf{u}\mathbf{w}], \mathbf{v})_{GL} \\ + \frac{1}{Re}(\nabla \mathbf{u}, \nabla \mathbf{v})_{GL} - (p, \nabla \cdot \mathbf{v})_G = 0 \quad \forall \mathbf{v}(\mathbf{r}, t) \in X_0^N \end{aligned} \quad (2.40)$$

$$-(q, \nabla \cdot \mathbf{u})_G = 0 \quad \forall q(\mathbf{r}, t) \in Y^N \quad (2.41)$$

where the global inner products (2.32) on G and GL nodes are denoted.

Lagrange interpolating polynomials, $\phi_i(r)$, are employed as basis functions to span the discretized velocity space \mathbb{P}_N , and pressure space \mathbb{P}_{N-2} , with the Lagrange polynomials satisfying the constraint

$$\phi_i(\xi_j) = \delta_{ij}, \quad (2.42)$$

where δ_{ij} is the Kronecker delta.

A major benefit of Gaussian quadratures is they allow for high order polynomials to be numerically integrated with high levels of accuracy. A polynomial, $f(r)$, is integrated along a 1-dimensional primary length, $-1 \leq r \leq 1$, using GL quadrature

$$\int_{-1}^1 f(r) dr \approx \sum_{l=0}^N \omega_l f(\xi_l), \quad (2.43)$$

with ξ_l the GL quadrature collocation points, and their associated weights ω_l , or using G quadrature

$$\int_{-1}^1 f(r) dr \approx \sum_{l=0}^N \sigma_l f(\eta_l), \quad (2.44)$$

where η_l are the G quadrature collocation points, and σ_l are their corresponding weights [57].

A scalar field $f_k(\mathbf{x})$ in $\Omega_k(t)$, is discretized for a one-dimensional element

$$f_k(\mathcal{M}_k^{-1}(r, t))|_{\Omega_k(t)} \approx \sum_{i=0}^N f_{k,i} \phi_i(r), \quad r \in [-1, 1], \quad (2.45)$$

and for a two-dimensional element

$$f_k(\mathcal{M}_k^{-1}(r, t))|_{\Omega_k(t)} \approx \sum_{i=0}^N \sum_{j=0}^N f_{k,ij} \phi_i(r_1) \phi_j(r_2), \quad r_1, r_2 \in [-1, 1]^2, \quad (2.46)$$

where Lagrange polynomials obey (2.42) and $\mathcal{M}_k^{-1}(r, t)$ returns the spatial coordinates of the element (2.38). We now have an approximation for $f_k(\mathbf{x})$ that is continuous throughout the element, $\Omega_k(t)$, once the coefficients $f_{k,ij}$ are determined.

2.2.3 One-Dimensional Operators

In this section, the matrix discretizations for the inner products in (2.40) and (2.41) are presented, first expressed for one dimensional variable (unbolded), on the primary interval $r \in [-1, 1]$. Inner products are performed locally for each element, with element index, k , denoting a local variable.

The inner product in the first term on the left side of equation (2.40) is discretized as

$$\begin{aligned} (u(r, t), v(r, t))_{k, GL} &= \int_{\hat{\Omega}} u_k(r, t) v_k(r, t) dr \\ &\approx \int_{\hat{\Omega}} \left(\sum_{i=0}^N u_{k,i}(t) \phi_i(r) \right) \left(\sum_{j=0}^N v_{k,j}(t) \phi_j(r) \right) dr \\ &\approx \sum_{j=0}^N v_{k,j}(t) \sum_{i=0}^N u_{k,i}(t) \int_{\hat{\Omega}} \phi_i(r) \phi_j(r) dr \\ &\approx \underline{v}_k^T(t) \hat{B} \underline{u}_k(t) \end{aligned} \quad (2.47)$$

where a hat over an unbolded variable denotes a one dimensional matrix operator for the primary length, and an underline denotes a vector of coefficients, $\underline{v} =$

(v_0, v_1, \dots, v_N) . The mass matrix, \hat{B} , can be computed using GL quadrature weights

$$\begin{aligned}\hat{B}_{ij} &= \int_{\hat{\Omega}} \phi_i(r) \phi_j(r) dr \\ &\approx \sum_{l=0}^N \omega_l \phi_i(\xi_l) \phi_j(\xi_l).\end{aligned}\tag{2.48}$$

Due to the Lagrangian basis function property in (2.42), the mass matrix simplifies to an $(N+1) \times (N+1)$ diagonal matrix with the quadrature weights along the main diagonal, $\hat{B} = \text{diag}(\omega_l)$.

In a similar manner, the inner product in the third term on the left side of equation (2.40) is discretized

$$\begin{aligned}\left(\frac{\partial u(r, t)}{\partial r}, \frac{\partial v(r, t)}{\partial r}\right)_{k, GL} &= \int_{\hat{\Omega}} u'_k(r, t) v'_k(r, t) dr \\ &= \int_{\hat{\Omega}} \left(\sum_{i=0}^N u_{k,i}(t) \phi'_i(r)\right) \left(\sum_{j=0}^N v_{k,j}(t) \phi'_j(r)\right) dr \\ &= \sum_{j=0}^N v_{k,j}(t) \sum_{i=0}^N u_{k,i}(t) \int_{\hat{\Omega}} \phi'_i(r) \phi'_j(r) dr \\ &= \underline{v}_k^T(t) \hat{A} \underline{u}_k(t)\end{aligned}\tag{2.49}$$

where the stiffness matrix, \hat{A} , in terms of GL quadrature becomes

$$\begin{aligned}\hat{A}_{ij} &= \int_{\hat{\Omega}} \phi'_i(r) \phi'_j(r) dr \\ &= \sum_{l=0}^N \omega_l \phi'_i(\xi_l) \phi'_j(\xi_l).\end{aligned}\tag{2.50}$$

In terms of the spectral differentiation matrix

$$\hat{D}_{ij} \equiv \left. \frac{d\phi_j}{dr} \right|_{r=\xi_i}\tag{2.51}$$

and using GL quadrature, the stiffness matrix (2.50) can be written

$$\hat{A}_{ij} = \sum_{l=0}^N \omega_l \hat{D}_{li} \hat{D}_{lj}.\tag{2.52}$$

The pressure term in equation (2.40) is represented in weak form using G quadrature. However, $\nabla \cdot \mathbf{v}$ values are defined at GL points, and therefore, the pressure values are interpolated to GL quadrature for compatibility. Thus the pressure term is discretized

$$\begin{aligned}
\left(p(r, t), \frac{\partial v(r, t)}{\partial r} \right)_{k, G} &= \int_{\hat{\Omega}} p_k(r, t) v'_k(r, t) dr \\
&\approx \int_{\hat{\Omega}} \left(\sum_{i=1}^{N-1} p_{k,i}(t) \phi_i(r) \right) \left(\sum_{j=0}^N v_{k,j}(t) \phi'_j(r) \right) dr \\
&\approx \sum_{j=0}^N v_{k,j}(t) \sum_{i=1}^{N-1} p_{k,i}(t) \int_{\hat{\Omega}} \phi_i(r) \phi'_j(r) dr \\
&\approx \underline{v}_k^T(t) \hat{\hat{D}}^T \underline{p}_k(t)
\end{aligned} \tag{2.53}$$

where $\hat{\hat{D}}$ is a local Derivative matrix defined using a weighted interpolation operator and the weighted one-dimensional derivative matrix. The inner product in the continuity equation (2.41) in one dimension is discretized in a similar manner

$$\left(q(r, t), \frac{\partial u(r, t)}{\partial r} \right)_G \approx \underline{q}_k^T(t) \hat{\hat{D}} \underline{u}_k(t). \tag{2.54}$$

For the nonlinear convective term (expressed in long format as in (2.28)), each term is integrated separately, and later combined for a full expression. The first inner product using one dimensional variables is discretized

$$\begin{aligned}
\left(u(r, t) \frac{du(r, t)}{dr}, v(r, t) \right)_{k, GL} &= \int_{\hat{\Omega}} u_k(r, t) u'_k(r, t) v_k(r, t) dr \\
&\approx \int_{\hat{\Omega}} \left(\sum_{i=0}^N u_{k,i}(t) \phi_i(r) \right) \left(\sum_{j=0}^N u_{k,j}(t) \phi'_j(r) \right) \left(\sum_{m=0}^N v_{k,m}(t) \phi_m(r) \right) dr \\
&\approx \sum_{m=0}^N v_{k,m}(t) \sum_{j=0}^N u_{k,j}(t) \sum_{i=0}^N u_{k,i}(t) \int_{\hat{\Omega}} \phi_i(r) \phi'_j(r) \phi_m(r) dr
\end{aligned} \tag{2.55}$$

and similarly for the other two terms

$$\left(w(r, t) \frac{du(r, t)}{dr}, v(r, t) \right)_{k, GL} \approx \sum_{m=0}^N v_{k,m}(t) \sum_{j=0}^N u_{k,j}(t) \sum_{i=0}^N w_{k,i}(t) \int_{\hat{\Omega}} \phi_i(r) \phi_j'(r) \phi_m(r) dr \quad (2.56)$$

and

$$\left(\frac{dw(r, t)}{dr} u(r, t), v(r, t) \right)_{k, GL} \approx \sum_{m=0}^N v_{k,m}(t) \sum_{j=0}^N u_{k,j}(t) \sum_{i=0}^N w_{k,i}(t) \int_{\hat{\Omega}} \phi_i'(r) \phi_j(r) \phi_m(r) dr \quad (2.57)$$

Combining the three convective terms gives

$$\left(\left[u \frac{du}{dr} - w \frac{du}{dr} - \frac{dw}{dr} u \right], v \right)_{k, GL} \approx \underline{v}_k^T(t) \hat{C}^k(t) \underline{u}_k(t) \quad (2.58)$$

The time dependent convective operator, $\hat{C}^k(t)$, can be computed using GL quadrature weights

$$\begin{aligned} \hat{C}_{ij}^k(t) = & \sum_{m=0}^N (u_{k,m}(t) - w_{k,m}(t)) \int_{\hat{\Omega}} \phi_m(r) \phi_j'(r) \phi_i(r) dr \\ & - \sum_{m=0}^N w_{k,m}(t) \int_{\hat{\Omega}} \phi_m'(r) \phi_j(r) \phi_i(r) dr. \end{aligned} \quad (2.59)$$

Utilizing GL quadrature, $\hat{C}_{ij}^k(t)$ becomes

$$\begin{aligned} \hat{C}_{ij}^k(t) = & \sum_{m=0}^N (u_{k,m}(t) - w_{k,m}(t)) \sum_{l=0}^N \omega_l \phi_m(\xi_l) \phi_j'(\xi_l) \phi_i(\xi_l) dr \\ & - \sum_{m=0}^N w_{k,m} \sum_{l=0}^N \omega_l \phi_m'(\xi_l) \phi_j(\xi_l) \phi_i(\xi_l) dr, \end{aligned} \quad (2.60)$$

which, upon using the property in (2.42), simplifies to

$$\hat{C}_{ij}^k(t) = (u_{k,m}(t) - w_{k,m}(t)) \omega_i \hat{D}_{ji} - \sum_{m=0}^N w_{k,m}(t) \omega_i \hat{D}_{mi} \delta_{ij}. \quad (2.61)$$

Notice that the convective operator is a nonlinear, time dependent operator defined on each element k due to its dependence on velocity, u , while all other (linear) operators in SEM are not element-dependent and not time-dependent. Additional details are found in [57] and [133] concerning the discretization of convective, pressure, and continuity terms, with corresponding operators.

2.2.4 Extension to Multiple Dimensions

In order to extend the applicability of the derived operators for use with multi-dimensional calculations, the tensor-product operations must be performed on the one-dimensional operators. As an example, the one-dimensional mass matrix is extended to three-dimensions by performing two tensor products

$$\underline{\hat{B}} = \hat{B} \otimes \hat{B} \otimes \hat{B}, \quad (2.62)$$

which has dimensions $(N + 1)^d \times (N + 1)^d$, where the underline denotes a multidimensional operator. Thus, inner products are given

$$(u(t), v(t))_{k, GL} = \underline{v}_k^T(t) \underline{\hat{B}} \underline{u}_k(t), \quad (2.63)$$

for multi-dimensional scalar variables on the primary element, where $\underline{u}_k(t)$ and $\underline{v}_k(t)$ denote coefficients of scalar variables in higher dimensions, $u(\mathbf{x}, t)$. Scalar coefficients of multiple-dimensional variables are stored in vector form as (in a two-dimensional case)

$$\underline{f}_k(t) = \begin{pmatrix} f_{k,1}(t) \\ f_{k,2}(t) \\ \vdots \\ f_{k,l}(t) \\ \vdots \\ f_{k,\mathcal{N}}(t) \end{pmatrix} = \begin{pmatrix} f_{k,00}(t) \\ f_{k,10}(t) \\ \vdots \\ f_{k,ij}(t) \\ \vdots \\ f_{k,NN}(t) \end{pmatrix}, \quad (2.64)$$

where coefficients are arranged in a vector from $l = 1$ to $l = \mathcal{N}$ with $l = 1 + i + (N+1)j$ and the length of the vector is $\mathcal{N} = (N+1)^2$. Coefficient arrays for three-dimensional variables are reordered in a similar manner.

2.2.5 Extension to Deformed Geometries

Additional modifications are made for geometries not conforming with the primary element definition, $[-1, 1]^d$. Recall that, in the current methodology, time dependent functions, (2.38) and (2.39), exist to map an element, Ω_k , from its physical coordinates in the laboratory frame to the primary frame, or vice-versa. Derivatives of one frame are mapped to the other using a matrix formulated by the chain rule.

$$\begin{pmatrix} \frac{\partial}{\partial r_1} \\ \vdots \\ \frac{\partial}{\partial r_d} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_1}{\partial r_1}(t) & \cdots & \frac{\partial x_1}{\partial r_d}(t) \\ \vdots & \ddots & \vdots \\ \frac{\partial x_d}{\partial r_1}(t) & \cdots & \frac{\partial x_d}{\partial r_d}(t) \end{pmatrix}_k \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_d} \end{pmatrix} \quad (2.65)$$

The determinant of this *chain rule* matrix is the time dependent transformation Jacobian

$$J_k(\mathbf{r}, t^n) = \begin{vmatrix} \frac{\partial x_1}{\partial r_1}(t) & \cdots & \frac{\partial x_1}{\partial r_d}(t) \\ \vdots & & \vdots \\ \frac{\partial x_d}{\partial r_1}(t) & \cdots & \frac{\partial x_d}{\partial r_d}(t) \end{vmatrix}_k \quad (2.66)$$

Derivation of operators for use with deformed geometries includes the Jacobian within the inner product calculations, and thus the Jacobian defined at the discrete collocation points is present in the definition of the matrix operators. For an element, Ω_k , in the laboratory frame, the unsteady term, for example, is expressed

$$(u(\mathbf{x}, t), v(\mathbf{x}, t))_{k, GL} = \underline{v}_k^T(t) \underline{B}_k(t) \underline{u}_k(t), \quad (2.67)$$

with an element index, k , and time-dependence, t , given to the mass operator. The operator hat is dropped to signify that the operator is now element and time dependent, no longer conforming to the primary element definition. The operator time dependence arises from the presence of a Jacobian, which is a function of time for moving grids, in its definition. For example, the mass operator in two dimensions:

$$\begin{aligned}\underline{B}_{k,ijj}(t) &= \int_{\hat{\Omega}} \phi_i \phi_i \phi_j \phi_j J_k(r_1, r_2, t) \, dr \\ &\approx \sum_{l=0}^N \sum_{m=0}^N \omega_l \omega_m \phi_i(\xi_l) \phi_i(\xi_l) \phi_j(\xi_m) \phi_j(\xi_m) J_k(\xi_l, \xi_m, t).\end{aligned}\quad (2.68)$$

2.2.6 Coupling Local Elements

The operators have, thus far, been expressed for systems within an individual local element. An aggregation of elemental inner products is performed to generate unassembled operators used for calculations within each global subdomain.

$$(u(\mathbf{x}, t), v(\mathbf{x}, t))_{GL}^n = \sum_{k=1}^E \underline{v}_k^T(t) \underline{B}_k(t) \underline{u}_k(t) = \underline{v}_L^T(t) \underline{B}_L(t) \underline{u}_L(t) \quad (2.69)$$

with subscript “ L ” used to represent a global unassembled array. This summation leads to block diagonal mass operator $\underline{B}_L(t) \equiv \text{diag}(\underline{B}_k(t))$, and subsequently block diagonal matrices for other operators $\underline{C}_L(t) \equiv \text{diag}(\underline{C}_k(t))$, $\underline{A}_L(t) \equiv \text{diag}(\underline{A}_k(t))$ and $\underline{D}_{L,i}(t) \equiv \text{diag}(\underline{D}_{k,i}(t))$. The subscript i in the interpolating derivative operator arises since a slightly different matrix is used for each dimension of the problem due to direction dependence in multidimensional vector lexicographical ordering. Notice that in unassembled arrays, which contain $E \cdot (N+1)^d$ values, continuity at element boundaries is not enforced, since adjacent elements need not contain identical values at their shared boundaries. Continuity is thus imposed by mapping an unassembled array (\underline{u}_L) into an assembled array (\underline{u}), in which adjacent elements reference the same values at shared boundaries. This process is performed with a connectivity operator, \underline{Q} , that maps unassembled arrays to assembled arrays, and vice versa, through *gather*

$$\underline{u}(t) = \underline{Q}^T \underline{u}_L(t) \quad (2.70)$$

and *scatter*

$$\underline{u}_L(t) = \underline{Q} \underline{u}(t) \quad (2.71)$$

operations. The absence of a subscript denotes a global assembled array, in which inter-element continuity is enforced by shared values at adjacent element boundaries. Assembled arrays thus contain fewer than $E \cdot (N + 1)^d$ values, since no duplicate entries exist for values that lie on boundaries of adjacent elements. The operator, \underline{Q} , is given by a matrix composed of “ones” and “zeros” which are arranged so that when \underline{Q} is multiplied with an assembled array, values located at an element boundary are assigned to every element that includes the boundary in the global unassembled array. Likewise, the transpose of \underline{Q} maps an unassembled array to a more condensed assembled array. The connectivity operator can act on both coefficient arrays and other operators [57]. Although an element moves and/or deforms, its relative position with respect to other elements in the subdomain remains the same, thus the connectivity operator is independent of time.

Gather and scatter operations are useful in global calculations of discretized values, as is seen in the inner-product of $u(\mathbf{x}, t)$ and $v(\mathbf{x}, t)$ using assembled velocity coefficient arrays operated upon by the unassembled mass operator

$$(u(\mathbf{x}, t), v(\mathbf{x}, t))_{GL} = \underline{v}_L^T(t) \underline{B}_L(t) \underline{u}_L(t) \rightarrow \underline{v}^T(t) \underline{Q}^T \underline{B}_L(t) \underline{Q} \underline{u}(t). \quad (2.72)$$

Global calculations for the other discretized terms in (2.40) and (2.41) is also performed by utilizing gather and scatter operations.

2.2.7 Extension to Vector Space

The multidimensional representation of the problem is given by the summation of individual scalar arrays

$$\begin{aligned}
(\mathbf{u}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t)) &= \sum_{i=1}^d (u_i(\mathbf{x}, t), v_i(\mathbf{x}, t)) \\
\Rightarrow (\mathbf{u}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t))_{GL} &= \sum_{i=1}^d \underline{v}_{i,L}(t) \mathbf{B}_L(t) \underline{u}_{i,L}(t) \\
\Rightarrow (\mathbf{u}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t))_{GL} &= \underline{\mathbf{v}}_L(t) \mathbf{B}_L(t) \underline{\mathbf{u}}_L(t)
\end{aligned} \tag{2.73}$$

where bolded variables represent vector-valued coefficient arrays and operators that act on such arrays. The vector-valued coefficient arrays, $\underline{\mathbf{u}}(t)$ and $\underline{\mathbf{v}}(t)$, are constructed by adjoining scalar-valued arrays, as $(\underline{u}_1(t), \underline{u}_2(t), \underline{u}_3(t))^T$. Operators for vector-valued arrays, $\mathbf{B}_L(t)$, are constructed with operators for scalar-valued arrays, $\underline{B}_L(t)$, to form block diagonal matrices.

2.2.8 Spatially Discretized Equations

Upon spatial discretization, the Navier-Stokes equations (2.1) and (2.2) become

$$\frac{d}{dt}(\mathbf{B}(t) \underline{\mathbf{u}}(t)) = -\underline{\mathbf{C}}(t) \underline{\mathbf{u}}(t) - \frac{1}{Re} \underline{\mathbf{A}}(t) \underline{\mathbf{u}}(t) + \tilde{\underline{\mathbf{D}}}^T(t) \underline{p}(t) \tag{2.74}$$

$$\tilde{\underline{\mathbf{D}}}(t) \underline{\mathbf{u}}(t) = 0 \tag{2.75}$$

where, recall the functions $\underline{\mathbf{u}}(t)$ and $\underline{p}(t)$ are functions of time, and time-dependent operators are defined by

$$\underline{\mathbf{A}}(t) = \underline{\mathbf{Q}}^T \underline{\mathbf{A}}_L(t) \underline{\mathbf{Q}}, \tag{2.76}$$

$$\underline{\mathbf{B}}(t) = \underline{\mathbf{Q}}^T \underline{\mathbf{B}}_L(t) \underline{\mathbf{Q}}, \tag{2.77}$$

$$\underline{\mathbf{C}}(t) = \underline{\mathbf{Q}}^T \underline{\mathbf{C}}_L(t) \underline{\mathbf{Q}}, \tag{2.78}$$

$$\tilde{\underline{\mathbf{D}}} = \tilde{\underline{\mathbf{D}}}_L \underline{\mathbf{Q}}. \tag{2.79}$$

2.2.9 Temporal Discretization

In the current implementation of the methodology, the spatially discretized governing equations are temporally discretized using k^{th} -order backward differentiation (BDF k), which is an implicit temporal discretization method based on a truncated Taylor series of the time derivative [57]. Assuming constant time stepping, the BDF k scheme applied to the spatially discretized momentum equation (2.74) takes the form

$$\frac{1}{\Delta t} \sum_{p=0}^k \beta_{pk} \underline{\mathbf{B}}^{n-p} \underline{\mathbf{u}}^{n-p} = -\underline{\mathbf{C}}^n \underline{\mathbf{u}}^n - \frac{1}{Re} \underline{\mathbf{A}}^n \underline{\mathbf{u}}^n + \underline{\tilde{\mathbf{D}}}^{T,n} \underline{p}^n \quad (2.80)$$

where the superscript n denotes the current timestep and the coefficients β_{pk} are given in Table 2.1.

Note, however, that in the ALE formulation of the problem, the values $\underline{\mathbf{B}}^n \underline{\mathbf{u}}^n$ are defined at the GL points in the current mesh configuration (at t^n) while the values $\underline{\mathbf{B}}^{n-p} \underline{\mathbf{u}}^{n-p}$ are defined at the GL points in a previous mesh configuration (at t^{n-p}). Recall that in the ALE formulation, calculations are carried out with respect to the moving mesh, not with respect to the laboratory frame. Thus to illustrate that BDF k scheme gives approximations with the correct temporal order of accuracy in the laboratory frame of reference, let

$$\mathcal{B}(x, t) \equiv B^n u^n$$

and

$$\mathcal{B}(x - \Delta x, t - \Delta t) \equiv B^{n-1} u^{n-1}$$

for a one dimensional problem, where the change in spatial location must obey the constraint $\Delta x = w \Delta t$, with w the mesh velocity. The BDF2 scheme, for example,

then gives

$$\frac{\delta}{\delta t}\mathcal{B}(x, t) \approx \text{BFD2}[\mathcal{B}] = \frac{1}{\Delta t} \left(\frac{3}{2}\mathcal{B}(x, t) - 2\mathcal{B}(x - w\Delta t, t - \Delta t) + \frac{1}{2}\mathcal{B}(x - 2w\Delta t, t - 2\Delta t) \right), \quad (2.81)$$

where the ALE derivative is again used for emphasis. Performing multivariable Taylor expansion for the last two terms gives

$$\begin{aligned} \mathcal{B}(x - w\Delta t, t - \Delta t) &= \mathcal{B} - \Delta t \frac{\partial \mathcal{B}}{\partial t} - w\Delta t \frac{\partial \mathcal{B}}{\partial x} + \\ &\quad \frac{1}{2} \left(\Delta t^2 \frac{\partial^2 \mathcal{B}}{\partial t^2} + 2w\Delta t^2 \frac{\partial^2 \mathcal{B}}{\partial x \partial t} + w^2 \Delta t^2 \frac{\partial^2 \mathcal{B}}{\partial x^2} \right) + \mathcal{O}[\Delta t^3] \end{aligned} \quad (2.82)$$

and

$$\begin{aligned} \mathcal{B}(x - 2w\Delta t, t - 2\Delta t) &= \mathcal{B} - 2\Delta t \frac{\partial \mathcal{B}}{\partial t} - 2w\Delta t \frac{\partial \mathcal{B}}{\partial x} + \\ &\quad \left(2\Delta t^2 \frac{\partial^2 \mathcal{B}}{\partial t^2} + 4w\Delta t^2 \frac{\partial^2 \mathcal{B}}{\partial x \partial t} + 2w^2 \Delta t^2 \frac{\partial^2 \mathcal{B}}{\partial x^2} \right) + \mathcal{O}[\Delta t^3], \end{aligned} \quad (2.83)$$

where $\mathcal{B} \equiv \mathcal{B}(x, t)$ in (2.82) and (2.83). Inserting (2.82) and (2.83) into (2.81) gives

$$\frac{\delta}{\delta t}\mathcal{B}(x, t) \approx \text{BFD2}[\mathcal{B}] = \frac{\partial \mathcal{B}}{\partial t} + w \frac{\partial \mathcal{B}}{\partial x} + \mathcal{O}[\Delta t^2], \quad (2.84)$$

which is, indeed, a second order temporal approximation to the ALE derivative defined in (2.17).

Karniadakis et al. [134] showed that an explicit k^{th} -order extrapolation (EXTk) method for approximation of the convection term maintains the stability of the BDFk scheme, and thus also circumvents the need for implicit treatment of the nonlinear term

$$\underline{\mathbf{C}}^n \underline{\mathbf{u}}^n \cong \sum_{q=1}^k \gamma_{qk} \underline{\mathbf{C}}^{n-q} \underline{\mathbf{u}}^{n-q}, \quad (2.85)$$

where the coefficients γ_{qk} are given in Table 2.1. We can illustrate that the EXT k scheme maintains the desired temporal order of accuracy on moving meshes by expressing the non-linear term as

$$\mathcal{C}(x - w\Delta t, t - \Delta t) \equiv C^{m-1}u^{n-1},$$

for a one dimensional problem, thus giving for EXT2

$$\mathcal{C}(x, t) \approx \text{EXT2}[\mathcal{C}] = 2\mathcal{C}(x - w\Delta t, t - \Delta t) - \mathcal{C}(x - 2w\Delta t, t - 2\Delta t), \quad (2.86)$$

where upon substituting the multivariable Taylor expansions (2.82) and (2.83) gives

$$\mathcal{C}(x, t) \approx \text{EXT2}[\mathcal{C}] = \mathcal{C}(x, t) + \mathcal{O}[\Delta t^2]. \quad (2.87)$$

	β_{p1}	β_{p2}	β_{p3}	γ_{p1}	γ_{p2}	γ_{p3}
p=0	1	3/2	11/6			
p=1	-1	-2	-3	1	2	3
p=2		1/2	3/2		-1	-3
p=3			-1/3			1

Table 2.1: Coefficients for the BDF k and EXT k schemes, $k=1,2,3$ [57, 75]

After some rearrangement, the temporally and spatially discretized Navier-Stokes equations (2.74) and (2.75) become

$$\underline{\mathbf{H}}^n \underline{\mathbf{u}}^n - \underline{\tilde{\mathbf{D}}}^{T,n} \underline{p}^n = \underline{\mathbf{f}}^n \quad (2.88)$$

$$\underline{\tilde{\mathbf{D}}}^n \underline{\mathbf{u}}^n = 0 \quad (2.89)$$

where the Hamiltonian operator

$$\underline{\mathbf{H}}^n = \left(\frac{\beta_0}{\Delta t} \underline{\mathbf{B}}^n + \frac{1}{Re} \underline{\mathbf{A}}^n \right), \quad (2.90)$$

and the right hand side

$$\underline{\mathbf{f}}^n = - \sum_{p=1}^k \beta_{pk} \underline{\mathbf{B}}^{n-p} \underline{\mathbf{u}}^{n-p} - \sum_{q=1}^k \gamma_{qk} \underline{\mathbf{C}}^{n-q} \underline{\mathbf{u}}^{n-q}. \quad (2.91)$$

We see that explicit handling of the nonlinear convective terms permits the discretized governing equations to be expressed as a Stokes problem.

2.2.10 Neumann Boundaries

Homogeneous Neumann boundary conditions, on $\partial\Omega_N$, are satisfied in the weak form of the governing equations (2.30) due to integration-by-parts performed on the inner product, $(\nabla^2 \mathbf{u}, \mathbf{v})$, which is presented in (2.23). The resulting integral over the domain boundary, $\partial\Omega$, contains the term $\nabla \mathbf{u} \cdot \hat{\mathbf{n}}$, and thus vanishes due to the Homogeneous Neumann boundary conditions, $\nabla \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{n} = 0, \mathbf{x} \in \partial\Omega_N$. Furthermore, we see that the Homogeneous Neumann boundary conditions are automatically satisfied in SEM since test functions were chosen such that $\mathbf{v} \in \mathcal{H}_0^1(\Omega)$, so the integral under consideration vanishes regardless, as discussed in Section 2.2.1 [57].

2.2.11 Dirichlet and Moving Wall Boundaries

Boundary conditions at moving wall and Dirichlet boundaries are satisfied by adding an additional constraint to the problem. First, a description of how homogeneous, or zero, boundary conditions are prescribed and proceed to setting up inhomogeneous boundary conditions.

Homogeneous Conditions An unassembled Mask matrix, \underline{M}_L , is used in the discretized representation of the problem for the satisfying of homogeneous Dirichlet boundary conditions. The Mask matrix is constructed such that its main diagonal is composed of values of one, though in positions that correspond to nodes on Dirichlet

or moving wall boundaries the value is set to zero, thus satisfying $\mathbf{u}^n = 0$ on $\partial\Omega_D^{g,n}$ and $\partial\Omega_W^n$. In the current implementation of the methodology, boundary conditions are assigned to nodes only upon initialization of the simulation, so while mesh movement may change the physical position of the nodes, their position in the global arrays remains constant, and consequently the Mask matrix is independent of time. The Mask matrix may enforce homogeneous boundaries on, for example, the term involving the Mass operator

$$(\mathbf{u}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t)) \rightarrow \underline{\mathbf{v}}^{T,n} \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L \underline{\mathbf{B}}_L^n \underline{\mathbf{M}}_L \underline{\mathbf{Q}} \mathbf{u}^n. \quad (2.92)$$

Inhomogeneous Conditions Inhomogeneous Dirichlet and moving wall boundaries are given by prescribed non-zero values obtained from discretization of equations (2.3b) and (2.3d), respectively. To impose this condition, the solution, \mathbf{u}^n is separated into a homogeneous part, \mathbf{u}_0^n , that satisfies the homogeneous conditions on Dirichlet and moving wall boundaries, and an inhomogeneous part, \mathbf{u}_b^n , that satisfies conditions at Dirichlet, \mathbf{u}_d^n on $\partial\Omega_D^{g,n}$, and moving wall, \mathbf{w}^n on $\partial\Omega_W^n$, boundaries. The homogeneous part is also a solution of the discretized equations (2.88) and (2.89), and as such it remains on the left-hand side of the problem. The inhomogeneous part is given by any function that is continuous through the domain and satisfies the conditions on $\partial\Omega_D^{g,n}$ and $\partial\Omega_W^n$, and it thus alters the right-hand side of equations (2.88) and (2.89). Such a function, \mathbf{u}_b^n , can be constructed using any inexpensive projection method [57].

2.2.12 Interface Conditions

Search Step In our present moving overlapping mesh methodology, a search procedure is performed once at the beginning of each timestep, as illustrated the the schematic presented in Figure 2.3, whose purpose is to find and determine the pri-

mary coordinates of each interface node at the current timestep, $\mathbf{x}^{[i],n} \in \Gamma^{ij,n}$, in terms of its location within the other subdomain, $\Omega^{j,n}$. Upon initialization of the problem, each interface point is flagged in order that the search procedure be enacted only for interface points. Since each subdomain moves relative to the other, the locations of interface points within the other subdomain change at each timestep. Even in the case of one stationary subdomain in the laboratory frame, the positions of its interface nodes within the moving subdomain certainly change, and thus the coordinates must be updated at each timestep. Upon the occasion that an interface node moves outside of the other subdomain, user imposed Dirichlet boundary conditions will be enforced at that point.

The first task within the search step is to determine which element in $\Omega^{j,n}$ encompasses the physical coordinates of the interface point $\mathbf{x}^{[i],n} \in \Gamma^{ij,n}$. This is carried out by comparing the physical coordinates of the point with the lines (or faces) that define each element within $\Omega^{j,n}$. When the encapsulating element, $\Omega_k^{j,n}$, is determined, the coordinates of the interface point in the laboratory frame, $\mathbf{x}^{[i],n}$ are mapped to primary coordinates, $\mathbf{r}^{[j]} \in [-1, 1]^d$, of the element $\Omega_k^{j,n}$. Figure 2.2 shows this mapping from physical to primary coordinates for different timesteps. We see that the search procedure must necessarily be performed at each timestep.

The search is treated as an optimization procedure to determine an interface point's primary coordinates, $\mathbf{r}^{[j]}$, within the element $\Omega_k^{j,n}$

$$\min_{\mathbf{r} \in [-1, 1]^d} \mathbf{h}^n(\mathbf{r}), \quad (2.93)$$

where $\mathbf{h}^n(\mathbf{r}) = \left| \mathbf{x}^{[i],n} - \mathcal{M}_k^{-1,[j]}(\mathbf{r}, t^n) \right|^2$, and $\mathcal{M}_k^{-1}(\mathbf{r}, t^n)$ is the function (2.39) that maps primary coordinates, \mathbf{r} , to physical coordinates, $\mathbf{x}^{[j],n}$, at time t^n in the physical element $\Omega_k^{j,n}$. Newton-Raphson iterations are performed until optimization is

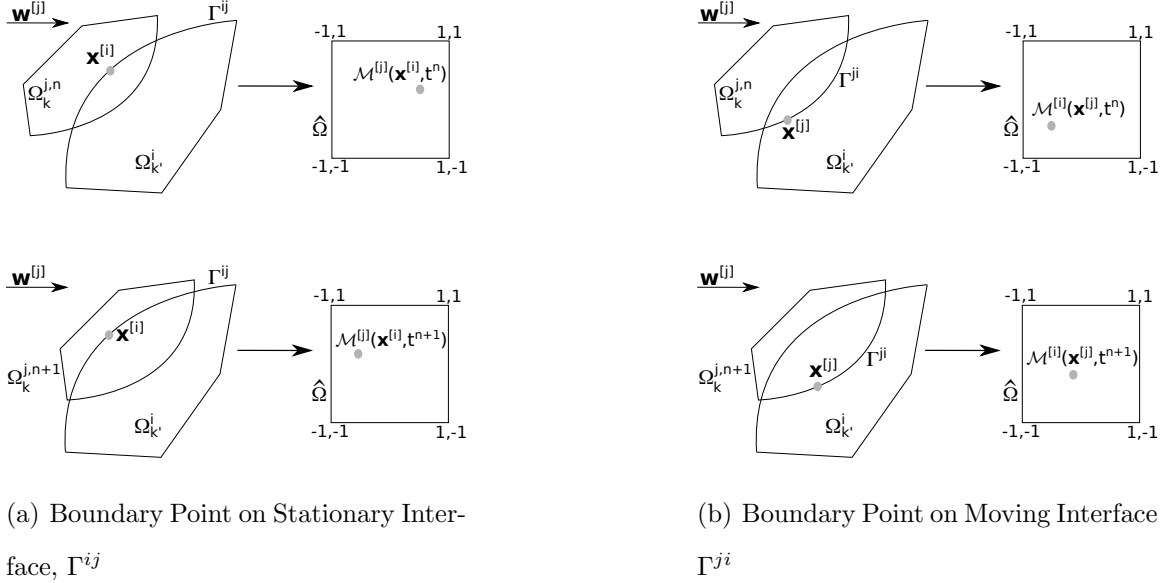


Figure 2.2: Illustrative representation of points on interface boundaries being mapped to reference coordinates at time t^n and t^{n+1} .

achieved:

$$\mathbf{r}^{p+1} = \mathbf{r}^p - \mathbf{S}^{-1,n}(\mathbf{r}^p) \mathbf{h}^n(\mathbf{r}^p), \quad (2.94)$$

where p is the iteration index, and $\mathbf{S}^n(\mathbf{r}) = \partial \mathbf{h}^n / \partial \mathbf{r}$.

Interpolation Step I will first describe a spatial interpolation approach to be done at each time step to find conforming values for the latest solution across the interface, and then address the temporal interpolation.

a) *Spatial Interpolation*

Suppose the solution at time step t^n has just been determined. Our task is to find the corresponding matching interface values with spectral accuracy. Thus, N^{th} -order Lagrangian interpolation is used that is consistent with the accuracy of the global SEM spatial scheme (2.46). Upon completion of the search step, the primary coordinates of each interface point are known within the corresponding elements in

the other subdomain. Consequently, the interpolated velocity value, $\mathbf{u}_{int}^{[j],n}(\mathbf{x}^{[i],n})$, is calculated from the solution in $\Omega^{j,n}$ with the interpolation formula

$$\mathbf{u}_{int}^{[j],n}(\mathbf{x}^{[i],n}) = \sum_{l=0}^N \sum_{m=0}^N \underline{\mathbf{u}}_{k,lm}^{[j],n} \phi_l(r_1^{[j]}) \phi_m(r_2^{[j]}), \quad \mathbf{x}^{[i],n} \in \Gamma^{ij}. \quad (2.95)$$

where the interpolated velocity is used at the corresponding interface point, $\mathbf{x}^{[i],n} \in \Gamma^{ij,n}$ in $\Omega^{i,n}$. Since velocity coefficients $\underline{\mathbf{u}}_{k,lm}^{[j],n}$ and associated Lagrange polynomials $\phi_l(r^{[j]})$ were previously determined by the SEM scheme, the correct determination of primary coordinates \mathbf{r} , which was performed in the search step, returns the spectrally accurate interpolated value $\mathbf{u}_{int}^{[j],n}(\mathbf{x}^{[i],n})$.

b) *Temporal Extrapolation*

To ensure specified temporal accuracy of the global time stepping scheme for the moving overlapping mesh methodology, an interface coupling scheme of the desired temporal accuracy must be developed. Here, we aim at designing a scheme that would keep a specified order of accuracy even without iterations. That would allow us to achieve a specified order of accuracy with a minimum number of iterations. The current work uses the IEXT m scheme developed previously for the stationary overlapping grid approach [75, 135]. To clarify the description, the scheme is first stated for stationary grids, and then applied to the moving grids.

For stationary grids, the IEXT m scheme takes solution from m previous time steps and uses explicit extrapolation of m^{th} order to find \mathbf{u} with m^{th} -order accuracy

$$\mathbf{u}_q^{[i],n} = \sum_{p=1}^m \gamma_{pm} \mathbf{u}_{int}^{[j],n-p}(\mathbf{x}_q^{[i]}), \quad \mathbf{x}_q^{[i]} \in \Gamma^{ij} \quad (2.96)$$

where q corresponds to a spatial index of discrete variables in a vector notation, and extrapolation weights, γ_{pm} , as used in the IEXT m scheme are found in Table 2.2 for up to $m = 3$. Thus, interpolated values of velocity are determined for previous timesteps

	γ_{p1}	γ_{p3}	γ_{p3}
p=1	1	2	3
p=2		-1	-3
p=3			1

Table 2.2: IEXTm schemes, m=1, 2, 3 [75]

using equation (2.95) and those values are utilized in (2.96) to satisfy conditions on stationary interface boundaries.

For moving grids, one has to be careful when applying this scheme, since the physical position of the mesh points change with time. The velocity value $\mathbf{u}_q^{[i],n}$ which corresponded to the interface boundary point $x_q^{[i]}$ in stationary grids, now corresponds to the interface boundary point which is now a function of time $x_q^{[i],n}$. Taking into account the temporal dependence of the interface points, equation (2.96) for moving grids in the current ALE formulation should be rewritten as

$$\mathbf{u}_q^{[i],n} = \sum_{p=1}^m \gamma_{pm} \mathbf{u}_{int}^{[j],n-p} (\mathbf{x}_q^{[i],n-p}), \quad \mathbf{x}_q^{[i],n-p} \in \Gamma^{ij,n-p} \quad (2.97)$$

Note that physical coordinates used for interpolation are taken at previous locations, x^{n-p} , rather than their location at the current time, x^n . Recall that in the ALE formulation of the problem, changes in fluid velocity over time are taken with respect to the reference (or ALE) frame, or with the reference coordinates, χ , held constant, as denoted by the ALE derivative used in equations (2.19) and (2.20). So it becomes necessary to interpolate at locations fixed to the moving interface points, since the added convective terms already take the relative mesh motion into account in the global calculations. Although, to ensure that the IEXTm scheme maintains the desired temporal order of accuracy with respect to the laboratory frame, $\mathbf{u}_{int}^{[j],n-p}$

is expressed as one dimensional velocity

$$u_{int}^{[j]}(x_q^{[i]} - w_q^{[i]} \Delta t, t - \Delta t) \equiv u_{int}^{[j],n-1}(x_q^{[i],n-1}),$$

which for IEXT2

$$\begin{aligned} u_{int}^{[j]}(x_q^{[i]}, t) &\approx \text{IEXT2}[u_{int}^{[j]}] = \\ &2u_{int}^{[j]}(x_q^{[i]} - w_q^{[i]} \Delta t, t - \Delta t) - u_{int}^{[j]}(x_q^{[i]} - 2w_q^{[i]} \Delta t, t - 2\Delta t), \end{aligned} \quad (2.98)$$

where upon substituting the multivariable Taylor expansions (2.82) and (2.83) gives

$$u_{int}^{[j]}(x_q^{[i]}, t) \approx \text{IEXT2}[u_{int}^{[j]}] = u_{int}^{[j]}(x_q^{[i]}, t) + \mathcal{O}[\Delta t^2]. \quad (2.99)$$

Explicit temporal extrapolation changes the stability properties of the global solver, even for stationary grids. The effects of IEXT m on stability were analyzed for a simplified example using the one-dimensional heat equation solved with a backward differentiation scheme [75]. It was found that larger mesh overlap size and an increased number of Schwarz-like iterations improved the stability of the simplified problem. Although not formally analyzed, similar stability issues were found in moving overlapping mesh incompressible flow simulations. To stabilize the solver in the present moving grid methodology, Schwarz-like iterations are performed in addition to the IEXT m scheme, similar to our previous stationary overlapping grid solver [135]. Interface conditions (2.3e) for the l^{th} iteration are defined

$$\left. \begin{aligned} \mathbf{u}_q^{[i],l=1,n} &= \sum_{p=1}^m \gamma_{pm} \mathbf{u}_{int}^{[j],l_{max},n-p}(\mathbf{x}_q^{[i],n-p}) \\ \mathbf{u}_q^{[i],l>1,n} &= \mathbf{u}_{int}^{[j],l-1,n}(\mathbf{x}_q^{[i],n}) \end{aligned} \right\} \quad \mathbf{x}_q^{[i]} \in \Gamma^{ij} \quad (2.100)$$

where the first iteration uses extrapolated values (2.97) from previous solutions at interface nodes, and succeeding iterations use the latest solution in the other subdomain for interpolated values. While velocity and pressure values are dependent on

the iteration count, the linear operators are not since geometry and position do not change with iterations. Since the nonlinear convection operator is treated explicitly, only values from previous timesteps are used in its calculation, and thus iterations do not affect its values at the current timestep.

Note that upon interpolation and extrapolation the boundary conditions on interfaces are enforced in a similar manner as Dirichlet and moving wall boundaries, with the unassembled Mask matrix zeroing out interface boundary points and with non-zero values at interfaces being treated as inhomogeneous Dirichlet boundaries, that is, assigning values at interfaces through a continuous function given on the right-hand side of the equations.

In the present moving overlapping mesh formulation, no more than three iterations were needed to attain stable solutions for the cases tested. While stability when using the IEXT scheme is necessarily enhanced with a small number of iterations, this is vastly different from implicit coupling methods that require iteration counts in the hundreds to achieve the desired temporal convergence.

2.2.13 *Mesh Velocity*

Mesh velocity is assigned on a point-by-point basis, with each GL node being assigned its own velocity value. Thus when prescribing mesh velocity, care must be taken to ensure that large mesh deformation or mesh entanglement is avoided. In all test simulations presented here, one of the meshes is held stationary while the other is translated and/or rotated as a rigid body, although the methodology allows for movement of both meshes, including the ability to handle deforming subdomains. Velocity values are prescribed to the discrete GL points once at the beginning of each timestep, after which the mesh geometry is updated accordingly (see Figure 2.3). The geometry of the grids remain unchanged for the remainder of the timestep.

Since the mesh geometry is updated once per timestep, the search step is only required once per timestep as well to determine the correct primary coordinates of each interface point in terms of the coincident elements in the neighboring mesh. The interpolation step, however, is required once per iteration to ensure that the most recent solution set is used for the interface conditions.

2.2.14 Iterative Solver

2.2.15 Fully-Discretized Equations

Decomposing $\underline{\mathbf{u}}$ for inhomogeneous boundary condition treatment gives the full discretized equations for two moving overlapping subdomains in block matrix form with iteration and timestep indices gives

$$\text{In } \Omega^1 : \begin{pmatrix} \underline{\underline{\mathbf{H}}}^{[1],n} & -\underline{\underline{\tilde{\mathbf{D}}}}^{T,[1],n} \\ -\underline{\underline{\tilde{\mathbf{D}}}}^{[1],n} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{[1],l,n} \\ \underline{\mathbf{p}}_L^{[1],l,n} \end{pmatrix} = \begin{pmatrix} \underline{\underline{\mathbf{F}}}^{[1],l,n}(\underline{\mathbf{u}}^{[2],n}) \\ \underline{\underline{F}}_p^{[1],l,n}(\underline{\mathbf{u}}^{[2],n}) \end{pmatrix}, \quad (2.101)$$

$$\text{In } \Omega^2 : \begin{pmatrix} \underline{\underline{\mathbf{H}}}^{[2],n} & -\underline{\underline{\tilde{\mathbf{D}}}}^{T,[2],n} \\ -\underline{\underline{\tilde{\mathbf{D}}}}^{[2],n} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{[2],l,n} \\ \underline{\mathbf{p}}_L^{[2],l,n} \end{pmatrix} = \begin{pmatrix} \underline{\underline{\mathbf{F}}}^{[2],l,n}(\underline{\mathbf{u}}^{[1],n}) \\ \underline{\underline{F}}_p^{[2],l,n}(\underline{\mathbf{u}}^{[1],n}) \end{pmatrix}. \quad (2.102)$$

where the double underlined assembled operators are now modified with the unassembled Mask matrix for satisfying Dirichlet, moving wall, and interface boundary conditions

$$\underline{\underline{\mathbf{A}}}^n = \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L \underline{\mathbf{A}}_L^n \underline{\mathbf{M}}_L \underline{\mathbf{Q}}, \quad (2.103)$$

$$\underline{\underline{\mathbf{B}}}^n = \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L \underline{\mathbf{B}}_L^n \underline{\mathbf{M}}_L \underline{\mathbf{Q}}, \quad (2.104)$$

$$\underline{\underline{\mathbf{C}}}^n = \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L^n \underline{\mathbf{C}}_L \underline{\mathbf{M}}_L \underline{\mathbf{Q}}, \quad (2.105)$$

$$\underline{\underline{\tilde{\mathbf{D}}}}^n = \underline{\tilde{\mathbf{D}}}_L^n \underline{\mathbf{M}}_L \underline{\mathbf{Q}} \quad (2.106)$$

and

$$\underline{\underline{\mathbf{H}}}^n = \left(\frac{\beta_0}{\Delta t} \underline{\underline{\mathbf{B}}}^n + \frac{1}{Re} \underline{\underline{\mathbf{A}}}^n \right). \quad (2.107)$$

The arrays on the right-hand side of (2.101) and (2.102) include the inhomogeneous Dirichlet, moving wall, and interface boundary conditions, all represented by $\underline{\mathbf{u}}_b$. Therefore, some of the values in $\underline{\mathbf{u}}_b$ are dependent upon interpolated/extrapolated values from the adjacent subdomain, and thus are dependent upon iteration count

$$\underline{\underline{\mathbf{F}}}^{[i],l,n}(\underline{\mathbf{u}}^{[j],n}) = \underline{\mathbf{f}}^{[i],n} - \underline{\mathbf{Q}}^T \underline{\underline{\mathbf{M}}}_L \underline{\underline{\mathbf{H}}}_L^{[i],n} \underline{\mathbf{u}}_{b,L}^{[i],l,n}(\underline{\mathbf{u}}^{[j],n}), \quad (2.108)$$

$$\underline{\underline{F}}_p^{[i],l,n}(\underline{\mathbf{u}}^{[j],n}) = \underline{\underline{\mathbf{D}}}_L^{[i],n} \underline{\mathbf{u}}_{b,L}^{[i],l,n}(\underline{\mathbf{u}}^{[j],n}), \quad (2.109)$$

with

$$\underline{\underline{\mathbf{H}}}_L^n = \left(\frac{\beta_0}{\Delta t} \underline{\underline{\mathbf{B}}}_L^n + \frac{1}{Re} \underline{\underline{\mathbf{A}}}_L^n \right). \quad (2.110)$$

Note that assembled operators are subdomain dependent, since they depend on different Jacobians, and time dependent, since Jacobians change as the mesh moves, although they are not iteration dependent for reasons stated in Section (2.2.12). Thus, every variable in equations (2.101) - (2.110) depend upon the mesh movement because of either the ever changing Jacobian, or the discrete mesh velocity essential in the determination of the convective terms. Mask matrices and Connectivity operators are subdomain dependent, although they do not depend upon time or iteration count.

The coupling terms in the right-hand side of (2.101) and (2.102), and the velocities $\underline{\mathbf{u}}_0^{l,n}$ are updated at the beginning of each iteration and the Stokes problems are solved independently for each subdomain. While bolded operators represent block-diagonal formulations of the scalar-valued operators, as discussed previously, the interpolating derivative operator is defined $\underline{\underline{\mathbf{D}}} = \left[\underline{\underline{\mathbf{D}}}_1 \quad \underline{\underline{\mathbf{D}}}_2 \quad \underline{\underline{\mathbf{D}}}_3 \right]$ to ensure proper multiplication of $\underline{\underline{\mathbf{D}}}^T$ and \underline{p}_L .

Details regarding the implementation of the Stokes solver for calculations of solutions in individual subdomains are found in [10, 57, 135].

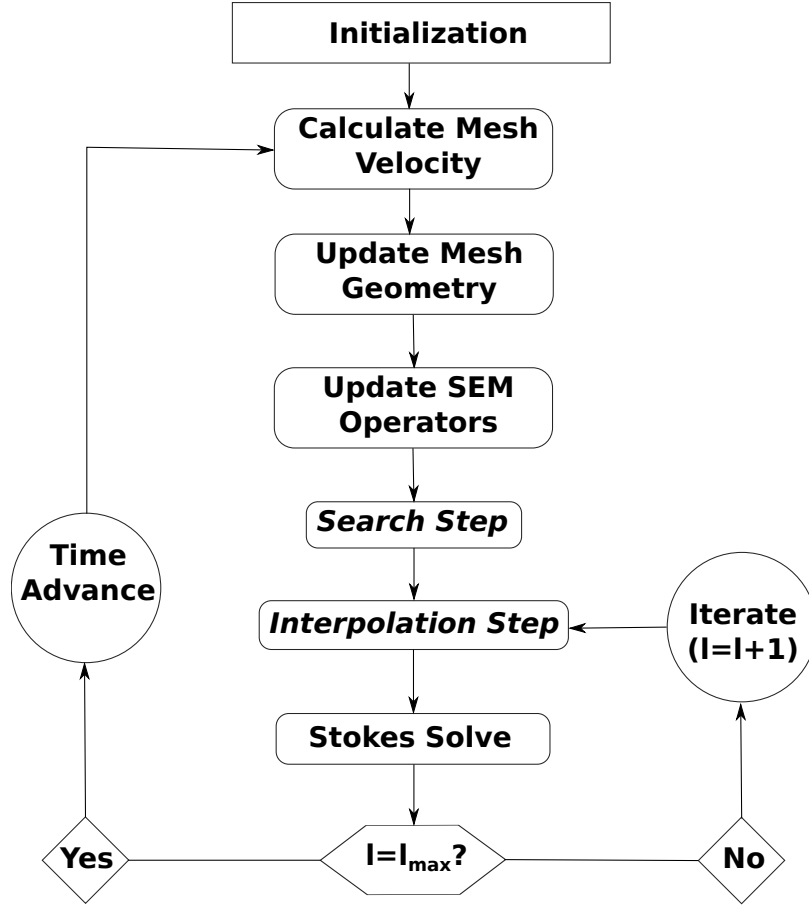


Figure 2.3: Basic procedures for computation of moving overlapping mesh methodology. Steps requiring global communication are in italic.

2.3 Parallel Communication

The dual-session communication framework, our previous development for stationary overlapping mesh problems, was utilized in the current methodology for the communication between two separate subdomains. The framework allows for independent computations in different sessions, while enabling the sessions to communicate with each other as needed. In our implementation for present validation tests, one session uses the SEM to solve for flow using the standard Eulerian formulation

of the governing equations, while the other session finds solutions on a moving domain with the ALE formulation of the problem described in Chapter 2. Within each session, communication among local processors is handled with MPI Intracommunicators, while global communication is accomplished through the establishment of an MPI Intercommunicator (Figure 2.4). Presently, all values are passed locally using Intracommunicators, while the interpolated values for use at interface boundaries are additionally passed to the other session via the Intercommunicator. This communication of interface values is the foundation for weak coupling between two subdomains, and occurs once per iteration.

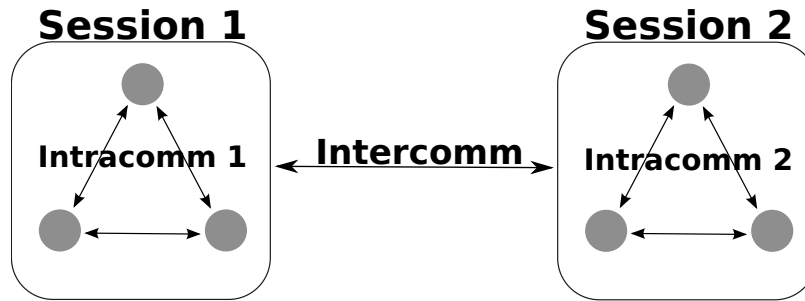


Figure 2.4: Dual-session communication framework for parallel communication among two independent sessions

When a simulation is first initiated, each processor scans through the GL points in its custody and flags any points with interface boundary conditions. Since interface boundaries in the current implementation are not altered throughout the simulation, this flagging is performed only once per simulation. The interface flags are saved, and the boundary conditions at those points are set to inhomogeneous Dirichlet conditions. The total number of interface points found within one session is divided equally among the processors in the other session, and the point identities (including physical coordinates in the laboratory frame) are sent via the Intercommunicator to processors in the other session. Upon receipt of point identities, the local session determines if each point lies within its subdomain boundary, and if it does not, the point is sent back

to its originating session where it is treated with user prescribed Dirichlet condition. For all flagged points within the subdomain's boundaries, the *Search Step* is initiated to determine, first, which local element encompasses each point. When such elements are found the points are sent to the local processors which are responsible for the said elements. The search step continues by determining the primary coordinated of each point with respect to the element in which it lies. When the search step is completed, the *Interpolation Step* is initiated to determine the spectrally accurate values for use at all interface points. The values are passed back to the neighboring session via the Intercommunicator, and are distributed locally through Intracommunicators to the processors requiring the interface point data. The interpolated values are utilized locally as inhomogeneous Dirichlet conditions at the interface boundaries.

Messages passed via the MPI communicators are optimized for parallel performance through utilization of the Crystal Router Algorithm [136]. Note that in the present formulation the mesh geometry is updated only at the beginning of a timestep, and thus the search step, requiring one global communication to and from each session, need be performed only once per timestep. Although, the interpolation step, also requiring one global communication to and from each session, is performed once per iteration to ensure the most up-to-date values are used at interface boundaries. All other computations are performed independently within each local session, allowing for optimum parallel scalability, as demonstrated in Chapter 5.

Chapter 3

STATIONARY OVERLAPPING MESH VALIDATION

In this chapter, I demonstrate performance of the developed overlapping grid method, on two- and three-dimensional laminar and turbulent problems, with the focus on spatial and temporal orders of convergence, accuracy in the presence of outflow disturbances and long integration times, and the ability of the method to reproduce correct turbulence characteristics on overlapping domains.

3.1 Convecting Walsh's Eddies

A simulation of convecting eddies in a two-dimensional periodic global domain was used to assess the spatial and temporal accuracy of the overlapping mesh method since an exact solution to the problem is readily available [137]. Walsh's discussion of non-convecting eddy solutions to the Navier-Stokes Equations states that if the initial conditions are set as $\tilde{\mathbf{u}} = \mathbf{a}$ where \mathbf{a} satisfies $\Delta \mathbf{a} = \lambda \mathbf{a}$, $\nabla \cdot \mathbf{a} = \mathbf{0}$ in Ω , then $\tilde{\mathbf{u}} = e^{\nu \lambda t} \mathbf{a}$, ν is kinematic viscosity, is a solution with the pressure defined as $\nabla \tilde{p} = -\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}$. The "tilde" denotes the values corresponding to a frame of reference where the locations of the eddies do not change with time (or convective frame of reference if the eddies convect). The eigenvalues λ can be expressed in terms of n and m , arbitrary non-negative integers, $\lambda = -(n^2 + m^2)$, with corresponding eigenfunctions formed by the product of $\sin[n\tilde{x}]$ or $\cos[n\tilde{x}]$ with $\sin[m\tilde{y}]$ or $\cos[m\tilde{y}]$. Linear combinations of the eigenfunctions form stream functions satisfying the Navier-Stokes Equations, from which velocity and pressure solutions can be found. For the eddies, convecting with the constant velocity (u_0, v_0) , the Walsh's solutions need to be transformed into a

stationary frame of reference by using the coordinate transformation: $x = \tilde{x} + u_0 t$, $y = \tilde{y} + v_0 t$. For our test case, the following exact solutions were used,

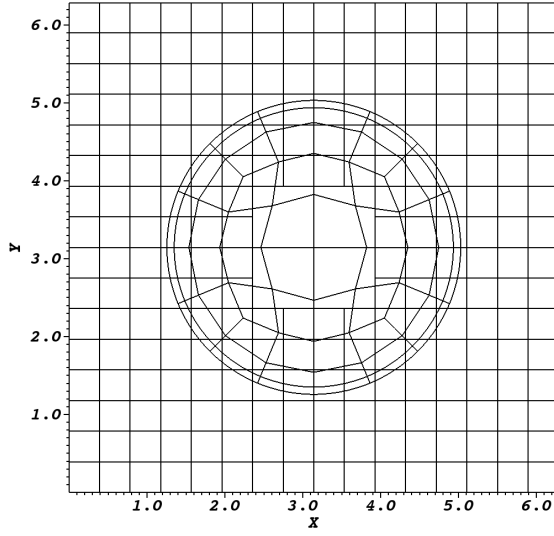
$$\begin{aligned} u(x, y, t) &= e^{-25t\nu} (-\cos[5(y - v_0 t)] + \cos[4(y - v_0 t)] \sin[3(x - x_0 t)]) + u_0, \\ v(x, y, t) &= e^{-25t\nu} (-\sin[5(x - x_0 t)] - \frac{3}{4} \cos[3(x - x_0 t)] \sin[4(y - v_0 t)]) + v_0, \end{aligned} \quad (3.1)$$

where $\nu = 0.05$. The corresponding solution for pressure (not presented in the original paper by Walsh [137]) is documented in Appendix A, together with the further discussion of the derivation. Since pressure in this case is not prescribed as a boundary condition, it is defined up to an arbitrary constant, thus the pressure is computed with the zero mean, both in numerical and exact solution.

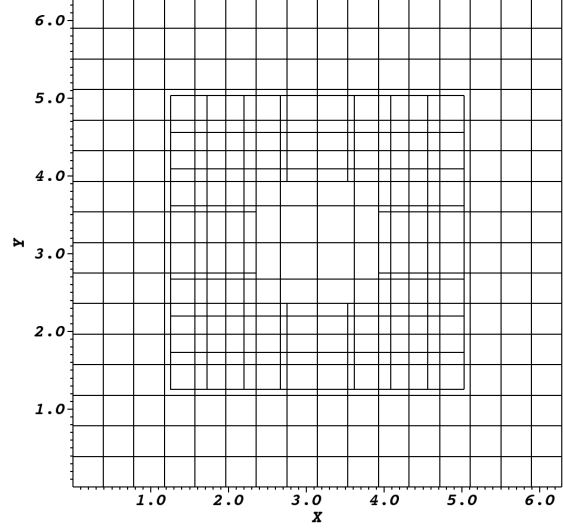
3.1.1 Mesh Configuration

The overlapping mesh configuration used for the present simulation consists of an exterior mesh with a vacancy in the middle and an interior mesh, either circular or square, that covers the vacancy (Figure 3.1). Note that the grid points (including on the interface boundaries) are not coincident with grid points in the adjacent subdomain. This is done in order to test for correct interpolation of values between the two meshes. Single mesh simulations, which were performed for comparison with two-mesh simulations, used a mesh with the same h-resolution as the exterior mesh shown in Figure 3.1.

Figure 3.2 presents the pressure contours for the overlapping simulations with circular and square interior meshes. The snapshots contain two sets of contour lines in sections where the subdomains overlap, although the continuity between the values in the two meshes make their differences indiscernible in most of the overlap region.

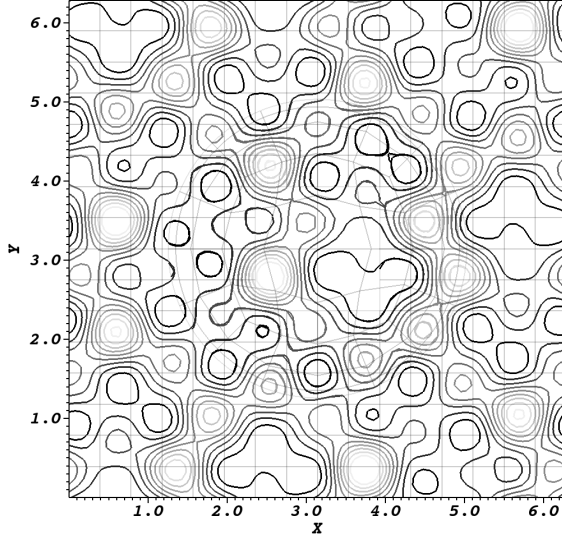


(a) Circular Interior Mesh

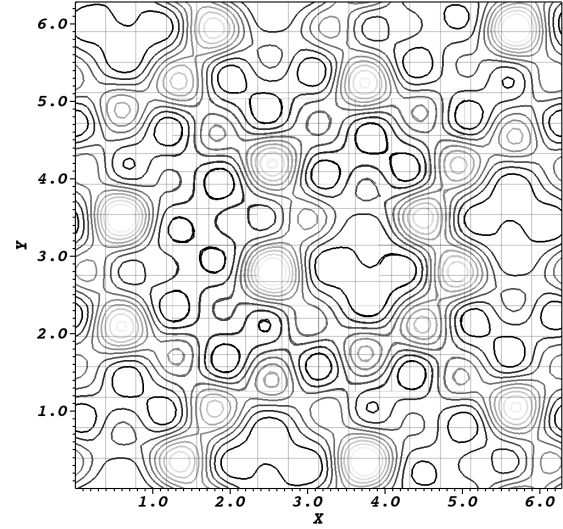


(b) Square Interior Mesh

Figure 3.1: Two-mesh domains for convecting eddies simulations, with element boundaries shown.



(a) Circular Interior Mesh



(b) Square Interior Mesh

Figure 3.2: Pressure contours taken after the 600th timestep using 7th-order polynomial approximation and third-order temporal scheme with $\Delta t = 1 \cdot 10^{-4}$. IEXT3 was performed using two iterations. The darkest lines represent a pressure of 0.8078 while the faintest lines represent a pressure of -2.710.

3.1.2 Accuracy

In this section, the spatial and temporal convergence of the grid overlapping method is investigated. The L^2 velocity error in each subdomain is calculated as

$$L^2 \text{ error } (\mathbf{u}^{[i]}) = \sqrt{\frac{\int_{\Omega^i} (\mathbf{u}_{\text{ex}}^{[i]} - \mathbf{u}_{\text{comp}}^{[i]})^2 dV}{2 V^{[i]}}, \quad (3.2)$$

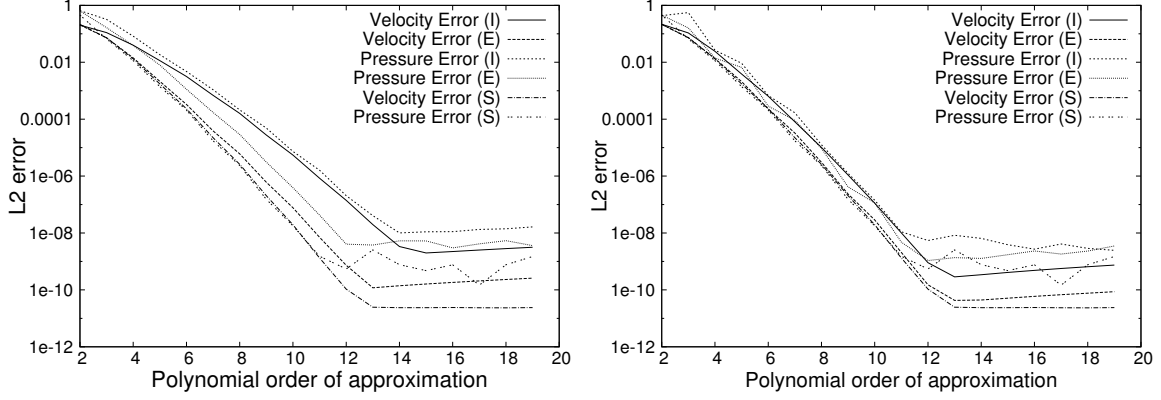
where $V^{[i]}$ is the volume of the subdomain Ω^i (the factor of two arises from the fact that for 2D problems two volume integrals are summed together, one for u velocity and one for v velocity, thus dividing by two results in the average), and the L^2 pressure error as

$$L^2 \text{ error } (p^{[i]}) = \sqrt{\frac{\int_{\Omega^i} (p_{\text{ex}}^{[i]} - p_{\text{comp}}^{[i]})^2 dV}{V^{[i]}}, \quad (3.3)$$

where $\mathbf{u}_{\text{ex}}^{[i]}$, $p_{\text{ex}}^{[i]}$ are the exact values and $\mathbf{u}_{\text{comp}}^{[i]}$, $p_{\text{comp}}^{[i]}$ are the computational approximations.

The spatial accuracy plots (Figure 3.3) show clear spectral convergence with p -refinement with respect to L^2 -errors in velocity and pressure. Notice also that a comparison of the plots for the circular and square interior meshes yields very similar results; the spatial errors for the simulation with a circular interior mesh being slightly larger than those for the square interior mesh case. This effect is due to a slightly slower (albeit still exponential) convergence of the SEM method with curvilinear meshes [138, 57, 139], and not due to the overlapping procedure. The spatial errors reported for single mesh simulations are slightly lower than their double mesh counterparts. This is attributed to the lower h -resolution of the interior mesh (see Figure 3.1), which also affects the accuracy of values used at interface boundaries of the exterior mesh.

The temporal accuracy plots (Figures 3.4 and 3.5) contain the results of simulations performed with IEXT k ($k = 1 - 3$) and third order time stepping compared with $\Delta t, (\Delta t)^2, (\Delta t)^3$ plots. The temporal convergence analysis was performed with



(a) Circular Interior Mesh

(b) Square Interior Mesh

Figure 3.3: Spatial accuracy for simulations using third-order time stepping scheme, with IEXT3 using three iterations. The I, E, and S in the key refer to the interior mesh in the two mesh simulation, the exterior mesh in the two mesh simulation, and the single mesh (square) simulation, respectively. The errors were collected after 1000 timesteps for simulations with $\Delta t = 1 \cdot 10^{-4}$.

17th order polynomials to make certain that the problem was fully converged spatially to ensure more accurate determination temporal convergence rates. However, since full spatial convergence is achieved in the current test problems for polynomial orders greater than 12, future temporal convergence test use lower order polynomials. As expected, the order of interface extrapolation is the limiting factor for the temporal accuracy, even though the third order temporal scheme was used in all cases. The results of the simulations generally meet the expected order of temporal convergence. Note that a larger number of iterations with a lower order interface extrapolation scheme will improve the temporal accuracy, though increase the computational cost.

Figures 3.6 and 3.7 illustrate the errors given by the same eddy simulation with Dirichlet boundary conditions prescribing the exact velocity at the global domain boundary. Both L^2 and L^∞ errors are plotted for spatial and temporal convergence.

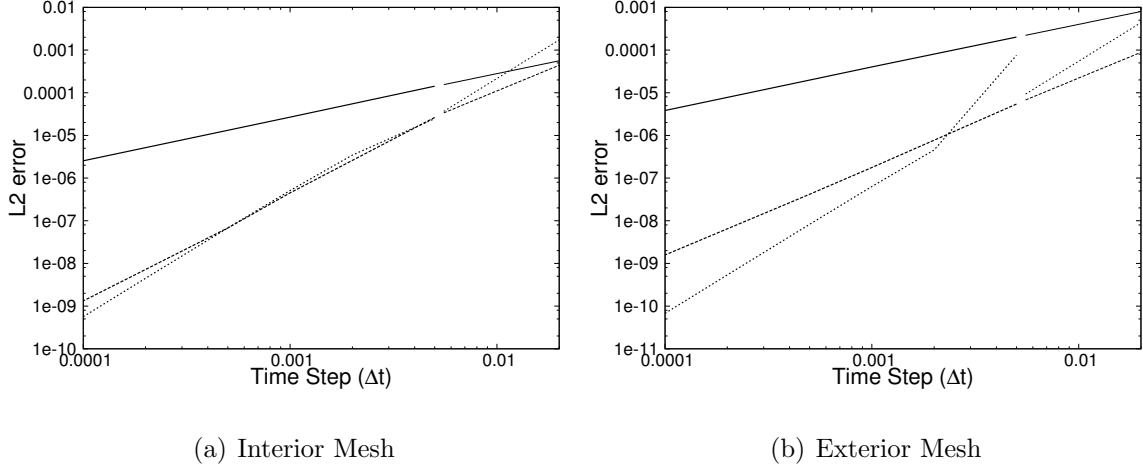


Figure 3.4: Temporal accuracy of the velocity for the square interior mesh simulation. The solid line represents the results for IEXT1, the dashed line represents IEXT2, and the dotted line represents IEXT3. In each case, the IEXT k scheme used three iterations. The lines to the right of the break are plots of Δt , $(\Delta t)^2$, $(\Delta t)^3$ for comparison with the simulation results. These results were collected for the 17th polynomial order of approximation, at time $1 \cdot 10^{-1}$.

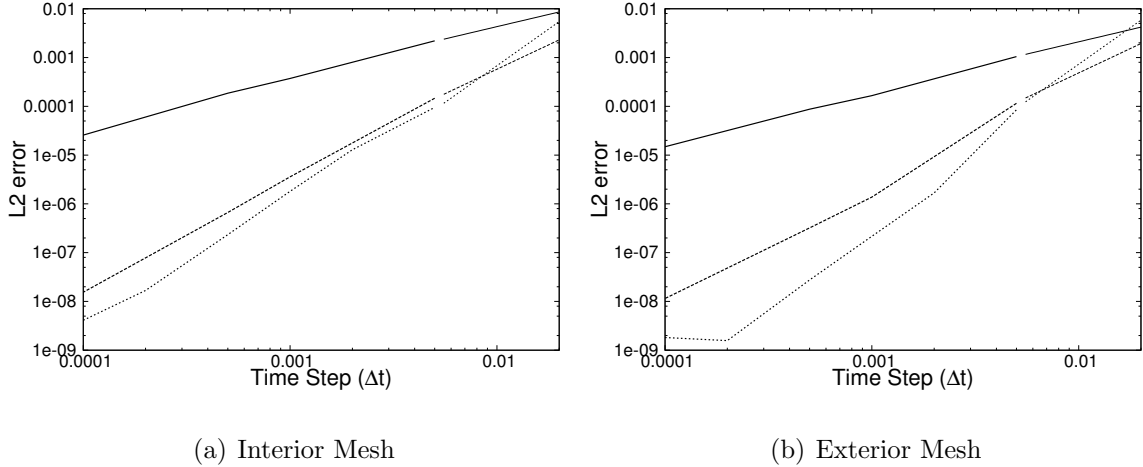


Figure 3.5: Temporal accuracy of the pressure for the square interior mesh simulation. The solid line represents the results for IEXT1, the dashed line represents IEXT2, and the dotted line represents IEXT3. In each case, the interface extrapolation used three iterations. The lines to the right of the break are plots of Δt , $(\Delta t)^2$, $(\Delta t)^3$ for comparison with the simulation results. These results were collected for the 17th polynomial order of approximation, at time $1 \cdot 10^{-1}$.

Looking at the L^∞ norm of the error, defined by

$$L^\infty \text{ error } (u^{[i]}) = \frac{\max \left(\left| u_{\text{ex}}^{[i]} - u_{\text{comp}}^{[i]} \right| \right)}{\max \left(\left| u_{\text{ex}}^{[i]} \right| \right)}, \quad (3.4)$$

shows that the largest error within the global domain also achieves spectral spatial convergence and 3rd order temporal convergence for both velocity and pressure.

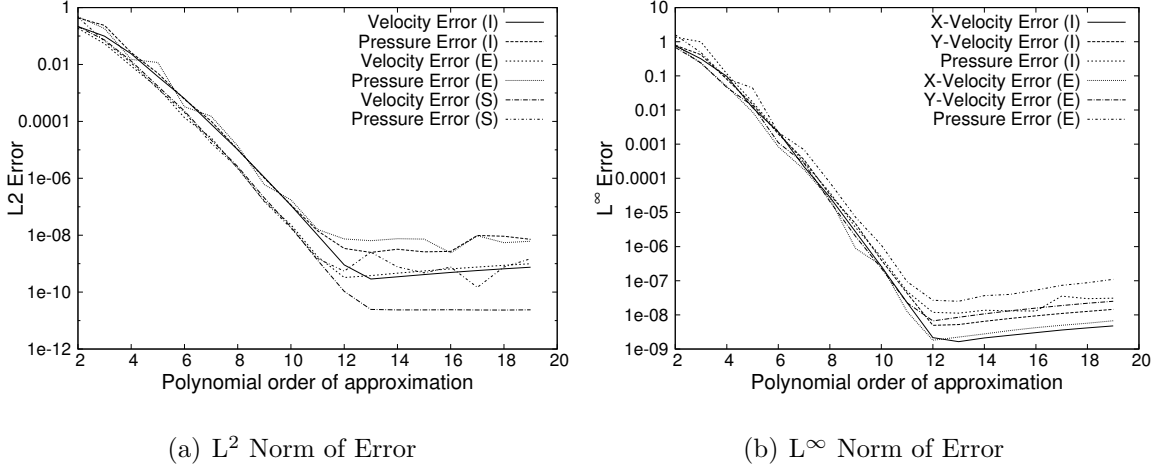


Figure 3.6: Spatial accuracy for square interior mesh tests with global Dirichlet boundary conditions using the third order time stepping scheme, with IEXT3 using three iterations. The I, E, and S in the key refer to the interior mesh in the two mesh simulation, the exterior mesh in the two mesh simulation, and the single mesh (square) simulation, respectively. The errors were collected after 1000 timesteps for simulations with $\Delta t = 1 \cdot 10^{-4}$.

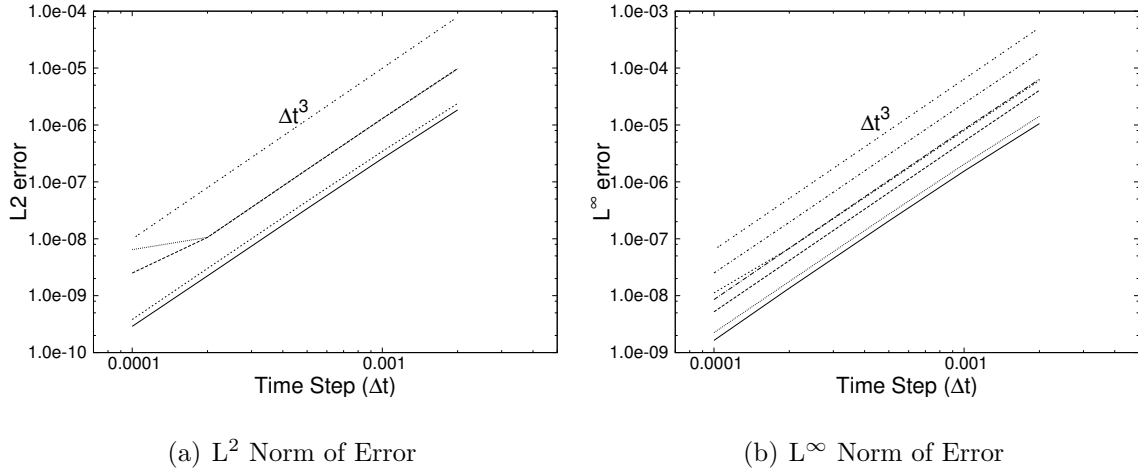


Figure 3.7: Temporal accuracy for square interior mesh simulations with global Dirichlet boundary conditions using the third order time stepping scheme, with IEXT3 using three iterations. The labeled line shows the expected order of temporal convergence, while the lines for velocities and pressures are displayed according to the legends in Figure 3.6 (Single mesh data is not included here). These results were collected for the 13th polynomial order of approximation, at time $1 \cdot 10^{-1}$.

Figure 3.8 shows the absolute errors of velocity magnitude in the interior and exterior subdomains. Since Dirichlet conditions are used on the global boundaries, it is expected that the largest errors in the exterior subdomain should be near the interface boundaries, as is the case. Interestingly, the largest errors within the interior subdomain do not lie near interface boundaries, but correspond to areas of largest fluid velocity magnitude. While there are distinct concentrated areas of larger error, all errors throughout the global domain are of the same order of magnitude.

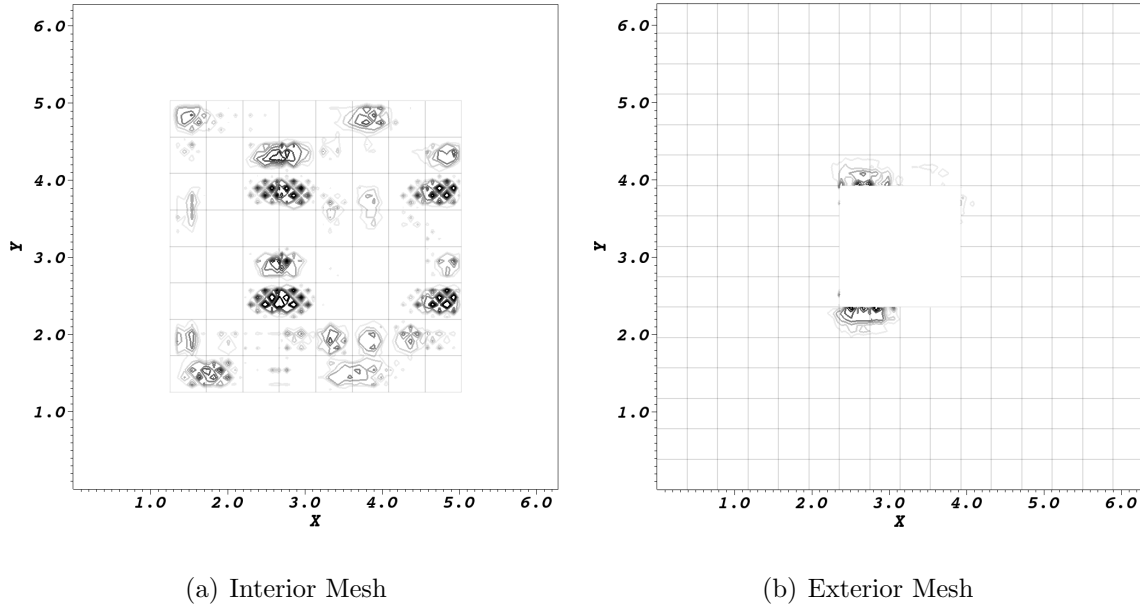
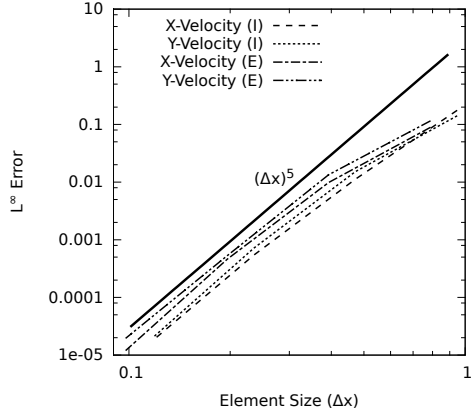


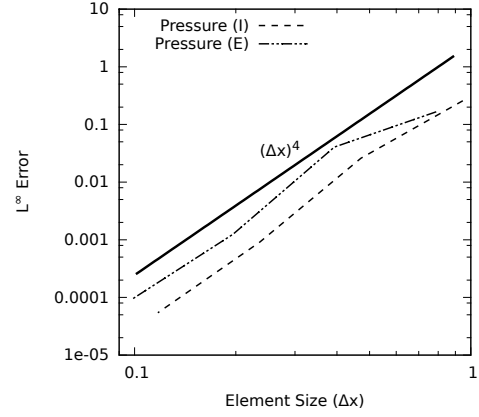
Figure 3.8: Contour plot of the absolute error in velocity magnitude values in the interior and exterior subdomains for a simulation with global Dirichlet boundary conditions, using 7th order polynomials, third order timestepping, and IEXT3 using three iterations. The snapshot is taken at time $1 \cdot 10^{-1}$. The darkest contour lines correspond to a value of $2.5 \cdot 10^{-4}$ and lightest contour lines $1.0 \cdot 10^{-4}$. The exterior subdomain has a maximum error of $2.799 \cdot 10^{-4}$ and the interior subdomain has a maximum error of $3.48 \cdot 10^{-4}$.

Investigation of the convergence rates (of the L^∞ norm of error) with respect to refinement of the average element size is illustrated in Figures 3.9 and 3.10 for square and circular interior mesh simulations with solutions approximated by 5th, 7th, and 9th order polynomials. We see that the h-refinement convergence tests generally show rates for velocity error of order h^N with square meshes, and convergence rates for pres-

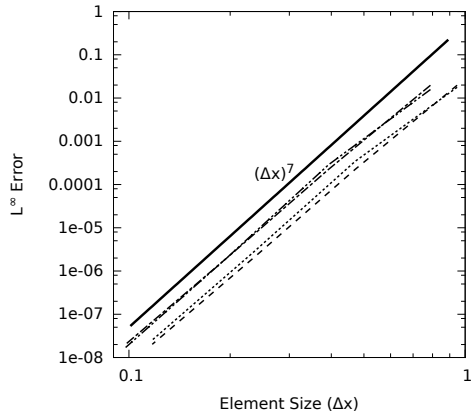
sure error of order h^{N-1} since pressure solutions are approximated on G rather than GL quadrature. Circular interior mesh simulations give velocity error convergence rates that are of order h^{N-1} for coarse h -resolutions, while pressure error generally maintains convergence rates of order h^{N-1} , as seen previously. Errors in circular mesh cases cease to decline at coarser resolutions than when square meshes are used. The lower order h -refinement convergence rate of the circular mesh is likely the leading cause of the slower, though still exponential, p -refinement convergence rate of the circular mesh illustrated in Figure 3.3.



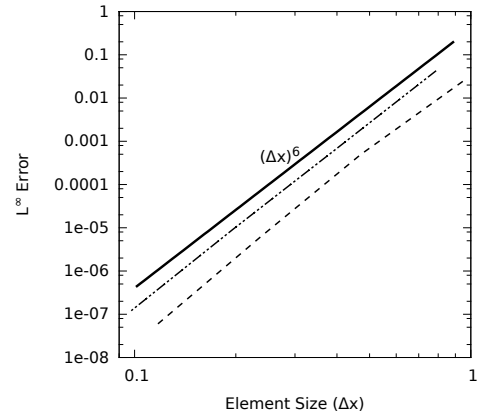
(a) Velocity Errors, 5th-Order Polynomials



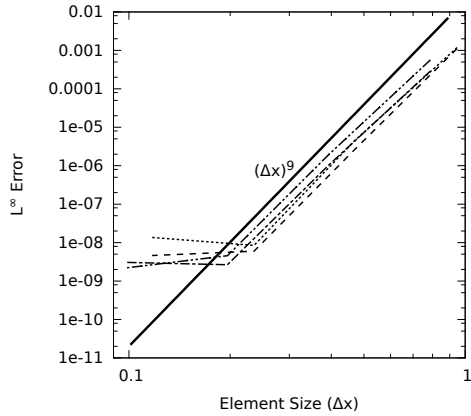
(b) Pressure Errors, 5th-Order Polynomials



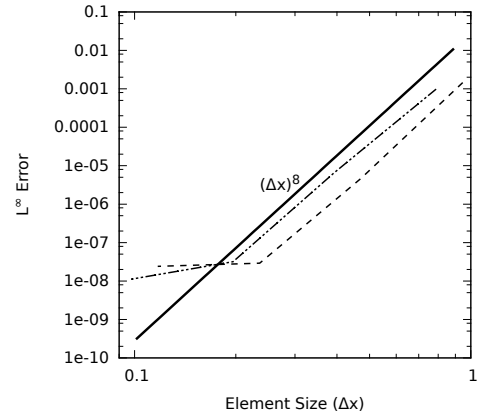
(c) Velocity Errors, 7th-Order Polynomials



(d) Pressure Errors, 7th-Order Polynomials

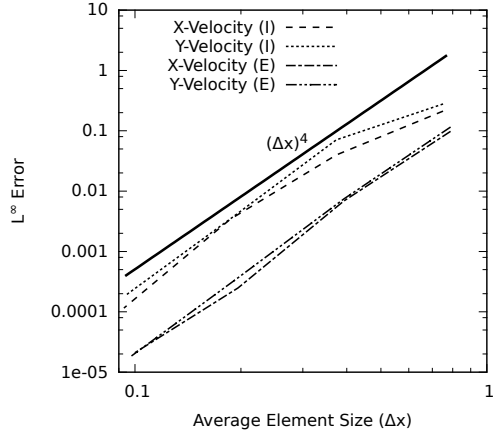


(e) Velocity Errors, 9th-Order Polynomials

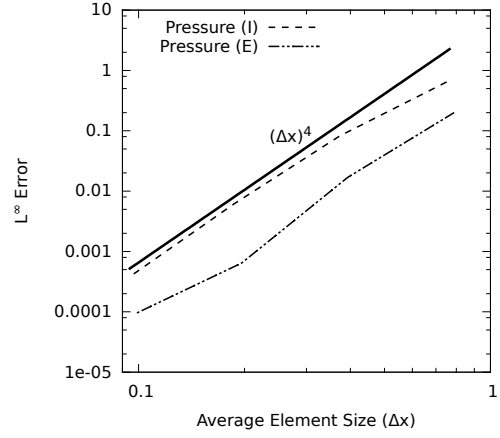


(f) Pressure Errors, 9th-Order Polynomials

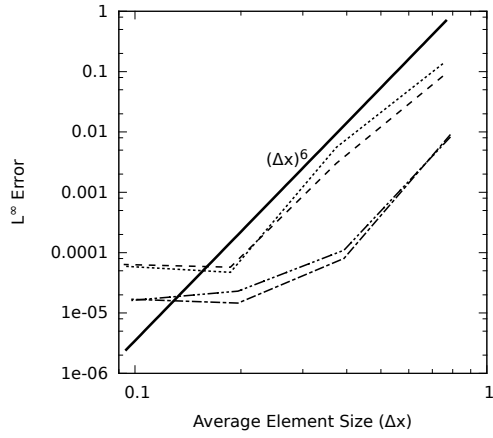
Figure 3.9: L^∞ error with respect to element size ($\Delta x \times \Delta x$) for square interior mesh simulations. Simulations were performed with $\Delta t = 1 \times 10^{-4}$ with the third order temporal scheme and IEXT3 using three iterations. Bold solid lines are the convergence rate reference lines, and are labeled with the corresponding convergence rate.



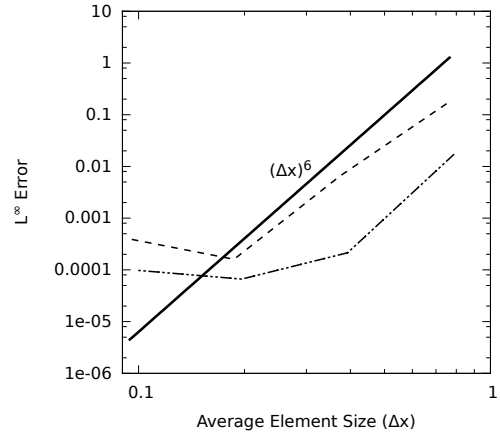
(a) Velocity Errors, 5th-Order Polynomials



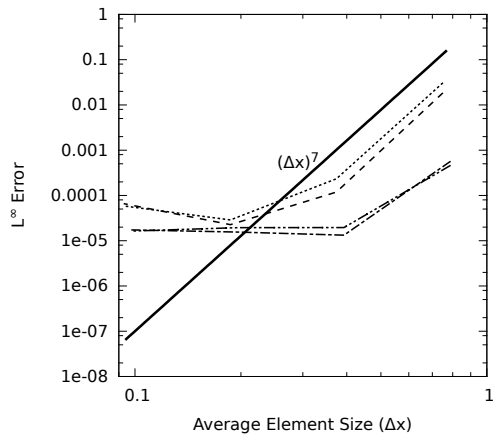
(b) Pressure Errors, 5th-Order Polynomials



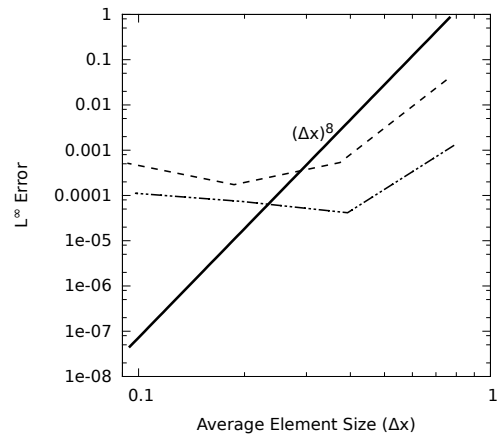
(c) Velocity Errors, 7th-Order Polynomials



(d) Pressure Errors, 7th-Order Polynomials



(e) Velocity Errors, 9th-Order Polynomials



(f) Pressure Errors, 9th-Order Polynomials

Figure 3.10: L^∞ error with respect to average element size ($\Delta x \times \Delta x$) for circular interior mesh simulations. See caption of Figure 3.9 for runtime parameters.

Initial tests show that the overlap size has little effect on the accuracy, and a large effect on stability. For convecting eddy test problems using square interior meshes with overlap size of about half of an element width, six iterations were needed for stability, while an overlap size that included only 2 GL points required nearly 500 iterations. An overlap size that included only one GL point proved to be unstable even up to 1000 iterations per timestep using 7th order polynomials. Future work will provide concrete details regarding how the accuracy and stability of the method is impacted by the size of the subdomain overlap.

The velocity and pressure profile plots for the overlapping mesh simulation are presented in Figure 3.11. As illustrated by the profile plots, good continuity exists in the transitions between the two meshes.

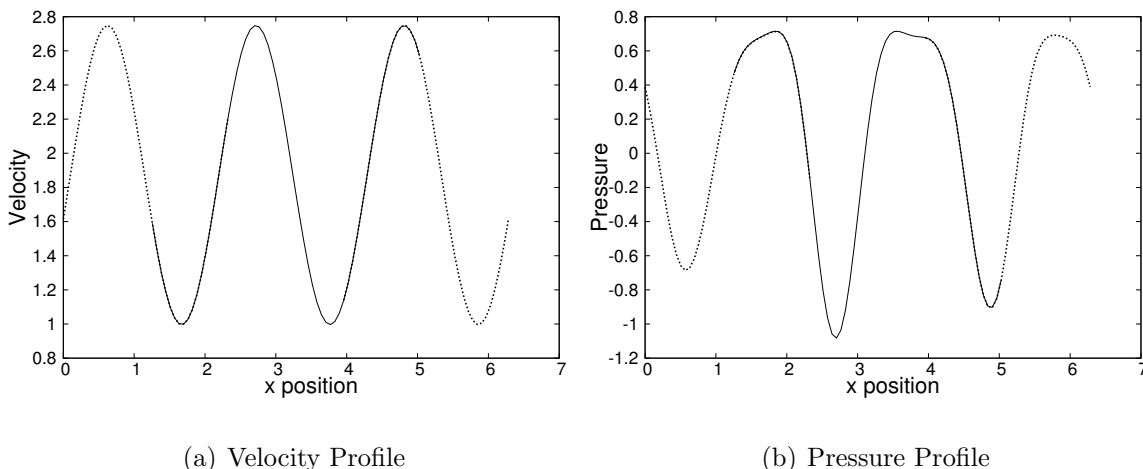


Figure 3.11: Cross-section profiles of the square interior mesh case through the center of the domain: The solid line illustrates the interior mesh data and the dotted line illustrates the exterior mesh data. These profiles were taken after 1000 timesteps from a third order simulation with IEXT3 using three iterations. The 7th order polynomial approximation was used with $\Delta t = 1 \cdot 10^{-4}$

3.2 Convecting Taylor Vortex

The computation of a two-dimensional convecting vortex traveling through a long, narrow domain composed of two overlapping subdomains with inflow-outflow global

boundary conditions is performed to test: 1) error behavior over long integration times; 2) influence of inexact outflow conditions on the overlapping grid solution. For that, two overlapping meshes are constructed to be long and narrow, with an overlapping region in the center of the global domain (see Figure 3.12). The vortex enters the left boundary of the global domain, traveling with a constant prescribed convecting velocity U_∞ , ultimately exiting the right boundary. The top, bottom, and left boundaries of the global domain are assigned Dirichlet boundary conditions for velocity, and the right boundary is assigned outflow conditions, defined by

$$p\mathbf{n} - \frac{1}{\text{Re}}\nabla\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega_O, \quad (3.5)$$

where $\partial\Omega_O$ is the outflow boundary, and \mathbf{n} is the vector normal to the outflow boundary.

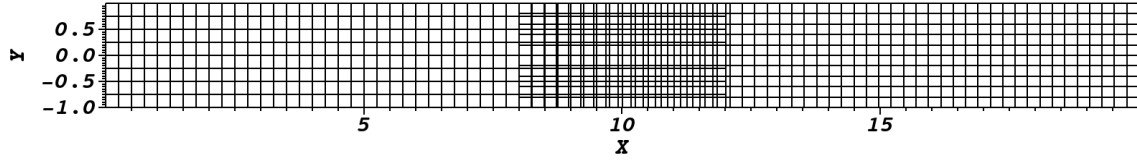


Figure 3.12: Mesh configuration for single convecting Taylor vortex simulations showing element boundaries.

The tangential velocity of a single incompressible Taylor vortex [140, 19, 141] in a convective frame of reference is prescribed by the equation

$$\tilde{u}_\theta = \frac{\mathcal{M}r}{16\pi\nu^2t^2} \exp\left(\frac{-\tilde{r}^2}{4\nu t}\right), \quad (3.6)$$

where \mathcal{M} is an invariant of the flow. As in the previous section, the variables with a tilde correspond to a convective frame of reference. The parameters t , ν , and \mathcal{M} are altered to obtain the desired vortex radius \tilde{r}_d , which is defined as the distance from the center of the vortex to the location of the maximum tangential velocity, and the desired maximum tangential velocity, $\tilde{u}_{\theta max}$. A relationship for the vortex radius is

found by solving $\frac{d\tilde{u}_\theta(r)}{d\tilde{r}} = 0$, giving

$$\tilde{r}_d = \sqrt{2\nu t_0} \quad (3.7)$$

from which ν and an arbitrarily chosen time, t_0 , can be manipulated to give the desired radius. Using this relationship, an equation is derived for \mathcal{M} that gives a vortex with the desired radius and maximum tangential velocity

$$\mathcal{M} = \frac{16\pi \cdot \tilde{u}_{\theta \max} \cdot \nu^2 \cdot t_0^2}{\sqrt{2\nu t_0}} e^{1/2}. \quad (3.8)$$

Since only a tangential velocity component exists, the momentum equation in the radial direction reduces to

$$\frac{\tilde{u}_\theta^2}{\tilde{r}} = -\frac{1}{\tilde{\rho}} \frac{\partial \tilde{p}}{\partial \tilde{r}}. \quad (3.9)$$

Thus, solving for the pressure of a non-convecting Taylor vortex gives

$$\tilde{p} = -\frac{\mathcal{M}^2 \tilde{\rho}}{256\pi^2 \nu^3 t^3} \exp\left(\frac{-\tilde{r}^2}{2\nu t}\right), \quad (3.10)$$

where \mathcal{M} is defined in equation (3.8).

Solution of the convecting vortex in a stationary frame of reference is then obtained by transforming polar coordinates of a convective frame into Cartesian coordinates of a stationary frame and adding $(U_\infty, 0)$ to the corresponding (\tilde{u}, \tilde{v}) velocity, while pressure is obtained from (3.10) with the account for coordinate transformation, as in Appendix A. As in the previous section, pressure with the zero mean is taken in both exact and numerical solutions.

In the present simulations, the maximum tangential velocity is given as a percentage of the convecting velocity, namely $\tilde{u}_{\theta \max}/U_\infty = 100\%$, which represents a very strong disturbance and a demanding test for the numerical method. Simulations were performed at a Reynolds number ($\text{Re} = U_\infty \tilde{r}_d/\nu$) of 140.

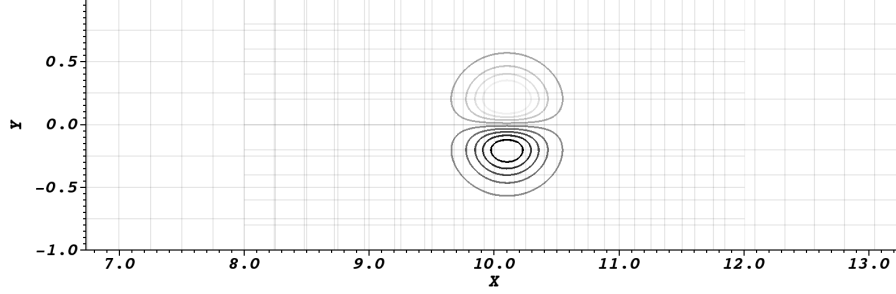


Figure 3.13: u -velocity contours when the vortex is in the overlapping region (1060^{th} timestep), using 9th-order polynomial approximations and 3^{rd} order temporal scheme, with $\Delta t = 1 \times 10^{-4}$. IEXT3 was used with four iterations. The darkest lines represent a nondimensional velocity of 1.277, and the faintest 0.7234.

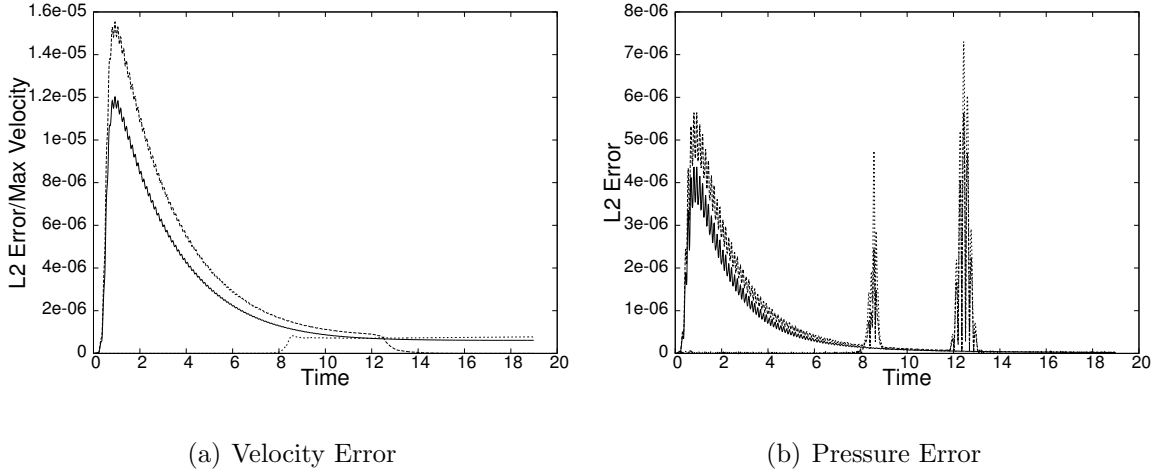


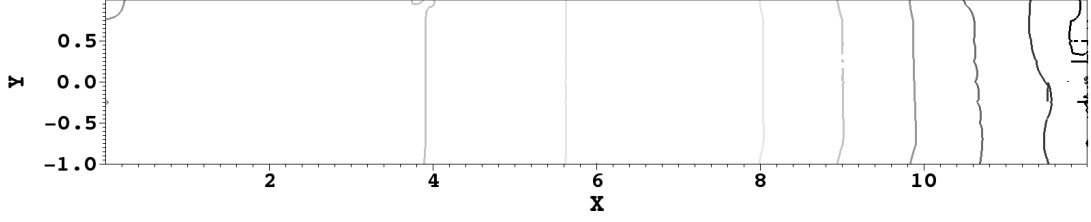
Figure 3.14: L^2 -errors as the vortex travels through the domain, where 9th order polynomials are used for spatial approximations and third order time-stepping was used with $\Delta t = 1 \times 10^{-4}$. IEXT3 with four iterations was used for interface values. The solid line represents the errors of the single mesh configuration, the dashed line represents the errors in the left mesh of the two-mesh case, and the dotted line represents the errors in the right mesh. The velocity errors have been normalized by the corresponding value of the maximum velocity at each time.

3.2.1 Accuracy

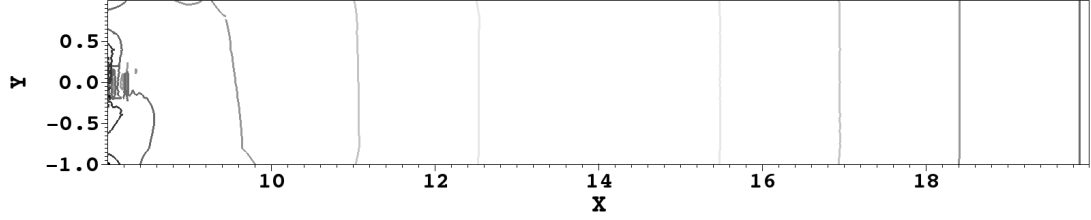
Figure 3.13 presents the u -velocity contours for the vortex as it resides in the overlap region. Contour lines are displayed by both the left and right meshes, although no detectable differences are observed. As the vortex travels through the domain the errors remain fairly constant (Figure 3.14). Note that in Figure 3.14 the errors as the vortex leaves the global domain through the the outflow boundary are not included,

to concentrate our attention on the errors as the vortex travels into and out of the overlapping region. The errors as the vortex leaves through the outflow boundary are several orders of magnitude larger (on the order of 10^{-3} for velocity and pressure) due to the outflow boundary prescribing non-exact pressure, thus altering the computed solution. The initial conditions prescribed in this problem place the vortex outside of the domain, giving nearly uniform unidirectional flow throughout the computational domain, thus initial computational errors are very low. As the vortex convects into the computational domain we see that the error values increase, as expected, for both single mesh and two mesh problems, after which the error gradually decreases. As the vortex enters and leaves the overlapping region the pressure errors spike since pressure is not interpolated at interface boundaries, although the errors are localized and do not significantly propagate into the domain (See Figure 3.15). The velocity errors remain largely unaffected by the vortex movement across the interfaces. The large errors experienced as the vortex crosses the global outflow boundary occur in both single mesh and two mesh problems, and they arise due to prescribed pressure at the outflow boundary that does not match the exact solution. Figure 3.16 shows the pressure error contours in the right subdomain when the outflow errors are at a maximum (which occurs when the center of the vortex is aligned with the outflow boundary of the global domain). The contours are given on a log-scale so that the faintest line represents a pressure error value of 5×10^{-6} while the darkest lines, representing a pressure error value of 1×10^{-3} , are localized at the outflow boundary.

Convergence tests, illustrated in Figure 3.17, show spectral convergence with respect to polynomial-order refinement, and the expected third-order temporal convergence. Errors were recorded when the vortex was centered in the overlapping region.



(a) Errors in Left Subdomain @ $t = 12.6$



(b) Errors in Right Subdomain @ $t = 8.5$

Figure 3.15: Pressure errors ($|p_{exact} - p_{comp}|$) throughout the domain as the vortex enters (b) and leaves (a) the overlapping region. 9th order polynomials are used for spatial approximations and third order time-stepping was used with $\Delta t = 1 \times 10^{-4}$. IEXT3 with four iterations was used for interface values. The darkest contour lines represent an error value of 1.4×10^{-6} and the lightest contours represent an error value of 2.3×10^{-7} .

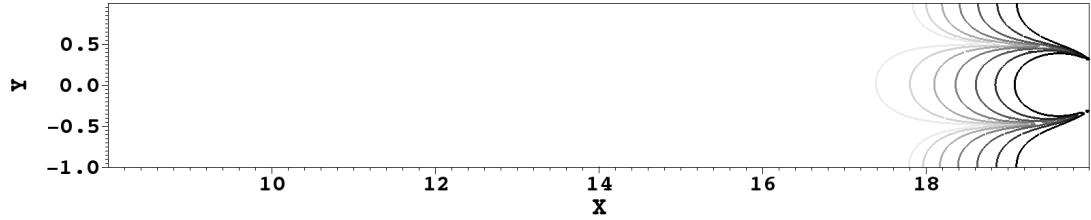
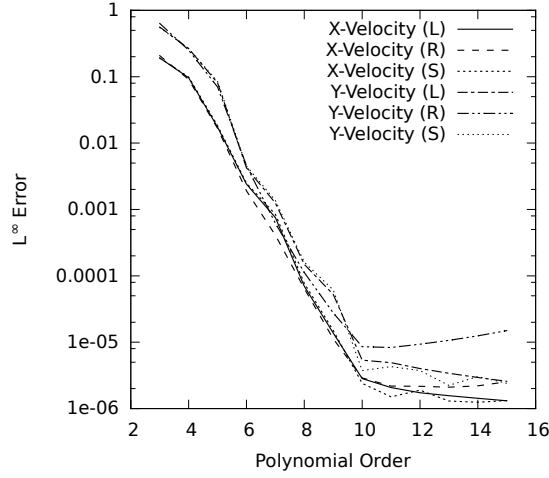
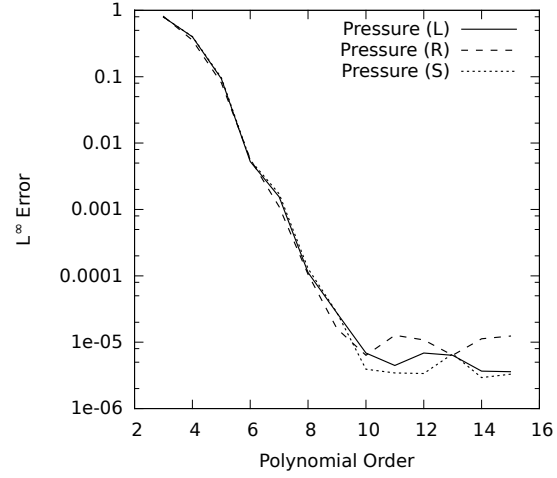


Figure 3.16: A snapshot of the pressure error contours at $t = 20.5$ when the errors due to the vortex leaving the global domain reach a maximum. The contour lines are given in logscale with the darkest lines representing a value of 1×10^{-3} and the lightest lines representing 5×10^{-6} . For runtime parameters see the caption of Figure 3.15.

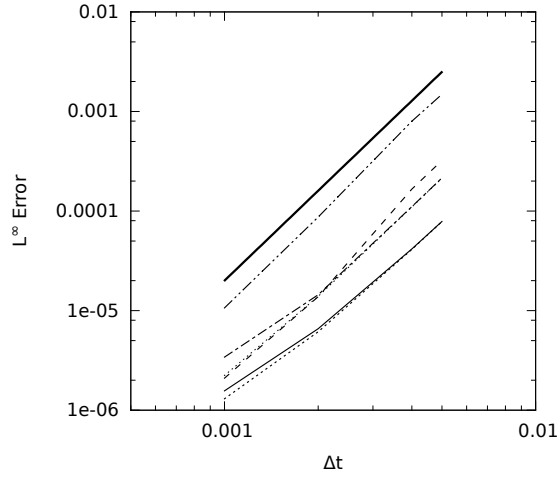
The profile plots in Figure 3.18 are taken through the center of the vortex when the vortex resides in the overlapping region. The values in the overlapping region show strong agreement between results in the two subdomains.



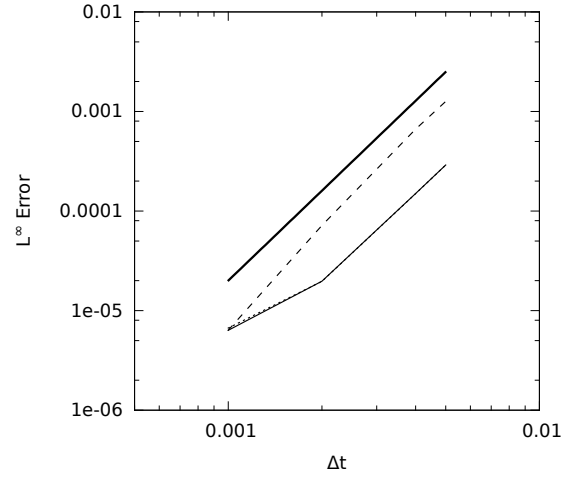
(a) Velocity Errors with p-Refinement



(b) Pressure Errors with p-Refinement



(c) Velocity Errors with Changing Δt

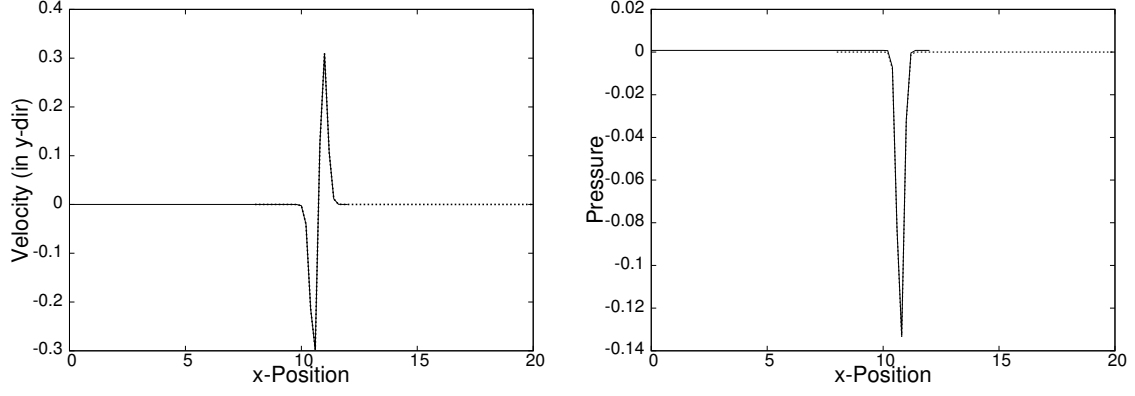


(d) Pressure Errors with Changing Δt

Figure 3.17: L^∞ errors with respect to polynomial order and Δt refinement for the left (L) and right (R) meshes in two mesh simulations and for single mesh (S) simulations. In the temporal convergence plots, the line types correspond to those used in the spatial convergence plots, and the bold solid line is the Δt^3 reference. Spatial convergence plots used timestep $\Delta t = 1 \times 10^{-3}$, and Temporal convergence plots were performed with 14th-order polynomials. In all cases, the BDF3/IEXT3 schemes were used with four iterations per timestep.

3.3 Turbulent Pipe Flow

In this section, a simulation of fully-developed turbulent pipe flow is performed to assess the applicability of the developed overlapping grid method to three-dimensional



(a) Velocity Profile

(b) Pressure Profile

Figure 3.18: Profile plots through the center of the vortex when the vortex resides in the overlapping region: The left mesh data is illustrated by a solid line while the right mesh data is illustrated by a dotted line. 9th order polynomials and a third order temporal scheme with $\Delta t = 1 \times 10^{-4}$ were used for solution approximations. Four iterations were used each timestep with the IEXT3 scheme.

turbulent flow cases. A direct numerical simulation (DNS) of the flow was performed with a bulk Reynolds number $Re_b = \frac{\overline{U_b} D}{\nu} = 5300$ to correlate with experiments performed in [142]. Sinusoidal perturbations in the θ and z directions were given in the initial flow conditions to enable more rapid transition to a turbulent state. The results were compared with the experimental and numerical data widely available from [142].

3.3.1 Mesh Configuration

A cross-sectional view of the inside and outside meshes is illustrated in Figure 3.19, and the length of the global pipe domain is 6 times longer than the pipe diameter D . The inside mesh contains a total of 1716 elements while the outside mesh contains 6000 elements, giving ~ 0.6 M and ~ 2.1 M gridpoints in the inside and outside meshes respectively using 6th-order polynomial approximations. Periodic boundary conditions are assigned at the ends of the global pipe domain. Single mesh simulations

were performed on a mesh of the same configuration as the inside mesh of Figure 3.19, though with 7th-order polynomial approximations.

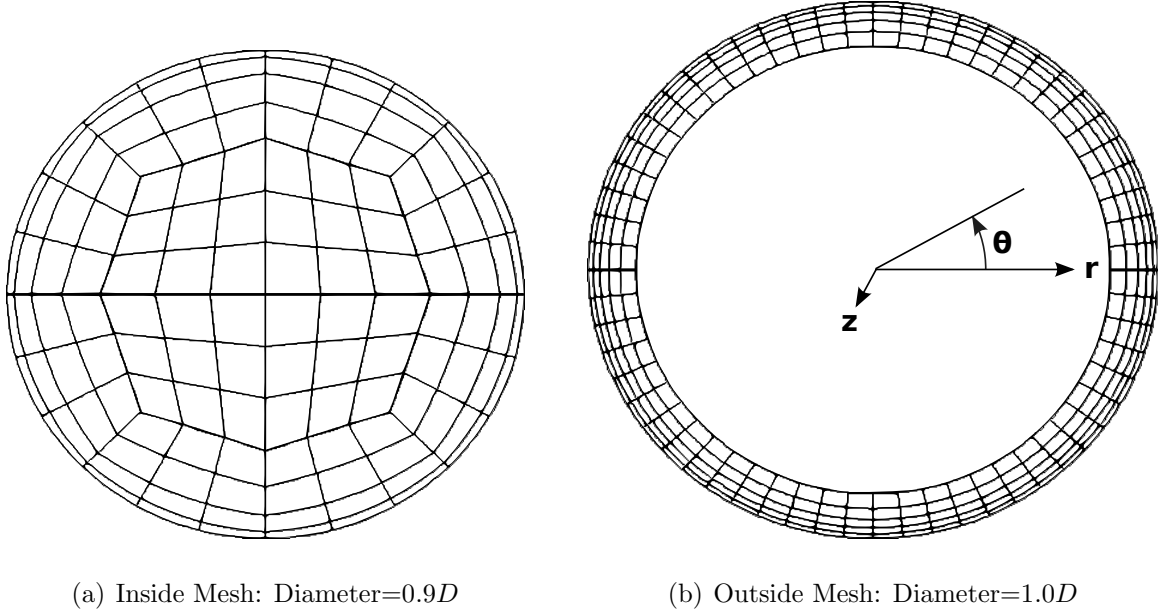


Figure 3.19: Cross section views of the inside and outside meshes for turbulent pipe flow (only the element boundaries are shown and not the GL points).

The overlapping grid methodology allows for efficient placement of refined elements in the locations where it is most needed. Thus, our outside mesh has much finer resolution in all radial, azimuthal and streamwise directions, enabling accurate representation of the boundary layer structures, while the inside mesh can be significantly coarser, minimizing the total number of elements and improving aspect ratios of the cells compared to a single domain. Since Gauss-Lobatto quadrature is used to determine the positions of the nodes between element boundaries, the spacing is not constant. Given this grid structure, the largest spacing among the nodes in elements near the center of the pipe (given in Cartesian coordinates) are $\Delta x^+ = \Delta y^+ \approx 8.121$ ($\Delta x = \Delta y \approx (2.250 \times 10^{-2})D$) and $\Delta z^+ \approx 39.044$ ($\Delta z \approx (1.082 \times 10^{-1})D$). (Wall units, or plus units, are given by $x^+ = u_\tau x / \nu$ and so forth for the other coordinate directions, where $u_\tau = \sqrt{\tau_w / \rho}$ is the friction velocity). The smallest spacing

among nodes in the center-most elements are $\Delta x^+ = \Delta y^+ \approx 2.941$ ($\Delta x = \Delta y \approx (8.150 \times 10^{-3})D$) and $\Delta z^+ \approx 14.138$ ($\Delta z \approx (3.918 \times 10^{-2})D$). The largest spacings among nodes in the outermost element of the outside mesh are $\Delta r^+ \approx 0.7311$ ($\Delta r \approx (2.026 \times 10^{-3})D$), $(\Delta r\theta)^+ \approx 4.4$ ($\Delta r\theta \approx (1.219 \times 10^{-2})D$), $\Delta z^+ \approx 25.4$ ($\Delta z \approx (7.039 \times 10^{-2})D$), while the smallest spacings (including nearest the wall) are $\Delta r^+ \approx 0.2648$ ($\Delta r \approx (7.338 \times 10^{-4})D$), $(\Delta r\theta)^+ \approx 1.6$ ($\Delta r\theta \approx (4.434 \times 10^{-3})D$), $\Delta z^+ \approx 9.2$ ($\Delta z \approx (2.550 \times 10^{-2})D$). Note that the width of the outermost element near the wall is $\Delta r_{el}^+ = 3.1$ ($\Delta r_{el} = (8.590 \times 10^{-3})D$) which means that this entire element lies well within the viscous sublayer which has a thickness of about 5 wall units. Comparatively, the point nearest the wall given in [142] is $\Delta r^+ = 0.94$.

3.3.2 Forcing

In the absence of a prescribed pressure gradient, the flow rate in a periodic pipe will eventually tend to zero due to viscous dissipation. Thus, a mechanism for sustaining the flow rate must be implemented in the simulations. In this work, a flow control scheme based upon incremental adjustment of the pressure gradient for maintenance of the bulk flow rate, U_b , is utilized. The adjusted pressure gradient is determined using the current and previous values of the bulk flow rate, U_b^n and U_b^{n-1} , and the target bulk flow rate, \overline{U}_b :

$$\left(\frac{\partial p}{\partial z}\right)^n = \alpha \left(\frac{1 + \overline{U}_b/U_b^n}{2}\right) \left(\frac{\partial p}{\partial z}\right)^{n-1}, \quad (3.11)$$

where

$$\alpha = 1 - \frac{D}{\overline{U}_b \Delta t} \left(\frac{U_b^n - U_b^{n-1}}{\overline{U}_b}\right). \quad (3.12)$$

In single domain simulations the determination of the current bulk flow rate is a straightforward task using fluid velocity and the domain volume

$$U_b^n = \frac{1}{V} \int_{\Omega} u^n dV. \quad (3.13)$$

However, in overlapping mesh simulations, a simple summation of the flow rates within each subdomain accounts for the value in the overlapping region twice,

$$U_b^n \neq \frac{1}{V^{[1]}} \int_{\Omega^1} u^{[1],n} dV + \frac{1}{V^{[2]}} \int_{\Omega^2} u^{[2],n} dV. \quad (3.14)$$

Thus, a modified determination of the global bulk flow rate must be devised to circumvent the repeated values. For that, the modified flow rates at individual subdomains

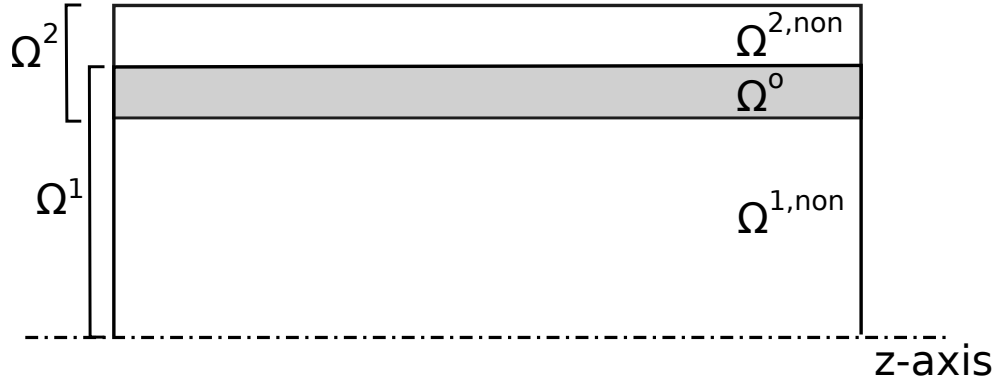


Figure 3.20: An axisymmetric cross section view of the inside and outside subdomains. Shaded areas denote regions where inside and outside subdomains overlap. Different regions are represented by different superscripts, namely, ‘o’ denotes an overlapping region, and ‘non’ a non-overlapping region, while subdomain reference numbers are also included.

are defined by

$$U_b^{[i],n} = \frac{1}{V^{[i]}} \int_{\Omega^i} \gamma u^{[i],n} dV, \quad (3.15)$$

where

$$\gamma = \begin{cases} 1 & \text{in } \Omega^{i,non}, i = 1, 2 \\ \frac{1}{2} & \text{in } \Omega^o \end{cases} \quad (3.16)$$

and domain superscripts correspond to those illustrated in Figure 3.20. The sum of the modified flow rates gives the correct value for the global flow rate:

$$U_b^n = U_b^{[1],n} + U_b^{[2],n}. \quad (3.17)$$

In general, the overlap region does not coincide with the element boundaries, thus the weights γ for the integration in Eq. (3.15) are assigned on a GL pointwise basis. If an element lies completely within the overlap region, the integration (3.15) effectively returns half of the actual flow rate in the element. However, if an element partially lies within the overlap region, only some of the GL points will be halved, resulting in a non-smooth integrand, thus reducing the accuracy of the numerical integration based on GL quadrature. While decreased accuracy in the calculation of U_b for a small portion of the domain will affect the value given for the global bulk flow rate, its effect will be small since the lower-order integration is localized to affected elements.

The global bulk flow rate (3.17) of the decomposed domain is used to adjust the global pressure gradient (3.11), thereby maintaining a prescribed value of \overline{U}_b . The effectiveness of the current flow control scheme to maintain a prescribed flow rate for the overlapping turbulent case is demonstrated in Figure 3.21. Pressure gradient (3.11) is subsequently used to calculate the wall shear stress $\tau_w = -\frac{D}{4} \frac{\partial p}{\partial z}$ and friction velocity $u_\tau = \sqrt{\tau_w / \rho}$ for the normalization of turbulent statistics.

3.3.3 Accuracy

The present DNS simulations were performed using second order time-stepping with $\Delta t = 1 \times 10^{-3}$, while IEXT2 was employed with two iterations for interface values. Second order temporal schemes, rather than third as in other tests, were used in this case to achieve an increase in computational speed, to allow for turbulent statistics to be collected over longer time periods. The flow was allowed to develop

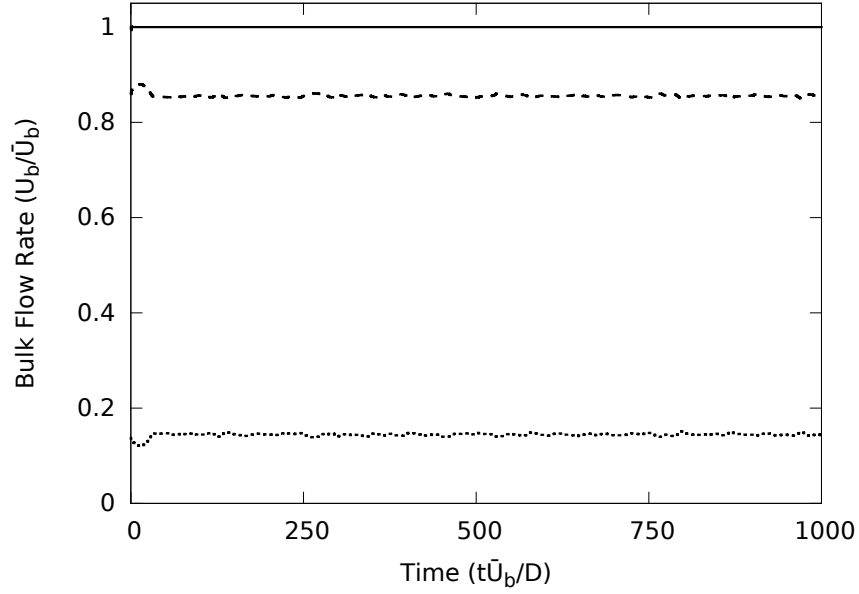


Figure 3.21: Modified flow rates of the individual domains, and the global bulk flow rate. The dashed line represents the modified flow rate of the inside mesh, the dotted line represents the modified flow rate of the outside mesh, the solid line represents their sum, which is the global bulk flow rate controlled to \bar{U}_b .

until the non-dimensional time $t\bar{U}_b/D = 1350$ was reached, after which turbulent statistics was collected for another 500 time units. Figures 3.22 and 3.23 show the results of the present simulation compared to the numerical and experimental results found in [142] for the mean normalized velocity and the root mean squared velocity. The present results give turbulent statistics that correlate well with the published data from several simulations and experiments. The present simulation gives a centerline Reynold's Number $Re_c = \frac{U_c D}{\nu} = 6815$ compared to $Re_c = 6950$ in the DNS results [142], and Re_τ Reynolds number $Re_\tau = \frac{u_\tau D}{\nu} = 360.9$ compared to the value of 360 reported in [142].

The velocity fluctuation contours shown in Figures 3.24 and 3.25 display good continuity between the two subdomains. The figures show that, in general, the velocity fluctuations near the pipe wall are larger than the fluctuations near the center. Evidence of the ejection of lower speed fluid (dashed contour lines) from the boundary

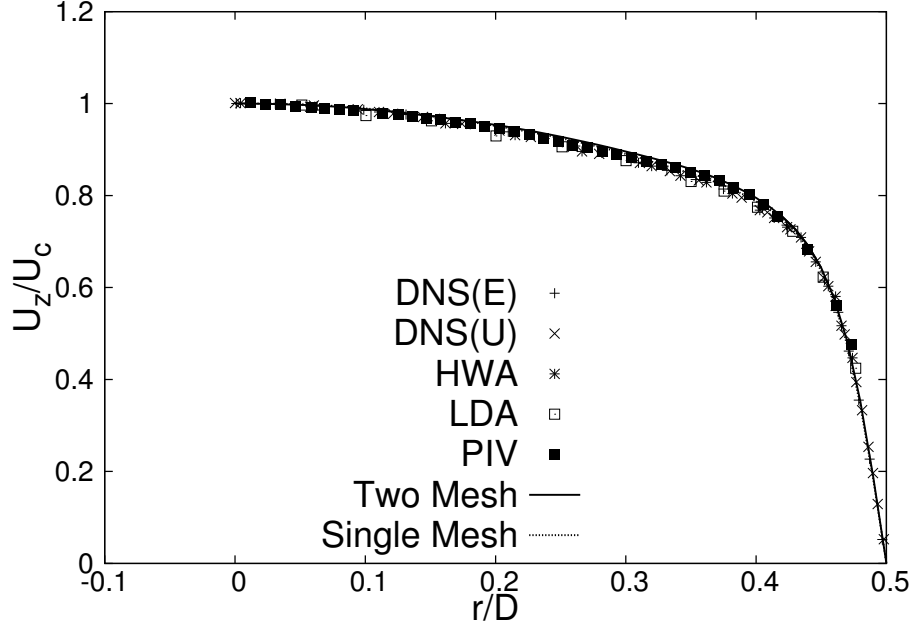


Figure 3.22: Average velocity profile of turbulent pipe flow normalized by the centerline velocity: The symbols represent the data presented in [142] (DNS(E): direct numerical simulation performed by Eggels & Nieuwstadt, DNS(U):direct numerical simulation performed by Unger & Friedrich, HWA: hot-wire anemometry, LDA: laser Doppler anemometry, PIV: particle image velocimetry) while the lines represent the data found using the simulation discussed here. Lines for the inside and outside meshes are both displayed on the plot, though the same line style is used to improve readability. The present simulation data was found using 6^{th} -order polynomials and second order temporal scheme with $\Delta t = 1 \times 10^{-3}$. IEXT2 was performed at interfaces with two iterations for multimesh simulations.

layer and the sweeping motion of higher speed fluid (solid contour lines) toward the pipe wall is seen in the mushroom shaped contour lines displayed in Figure 3.25.

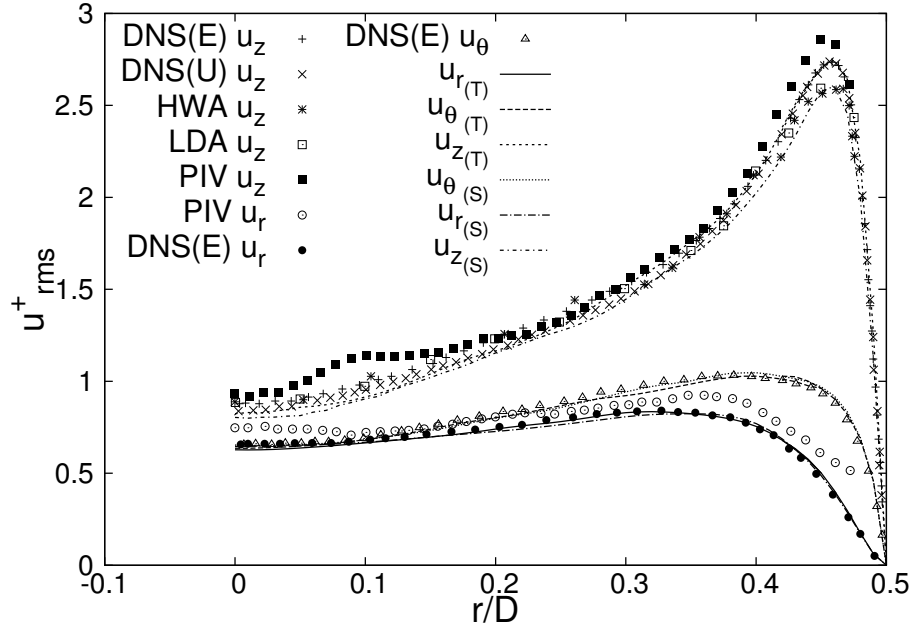
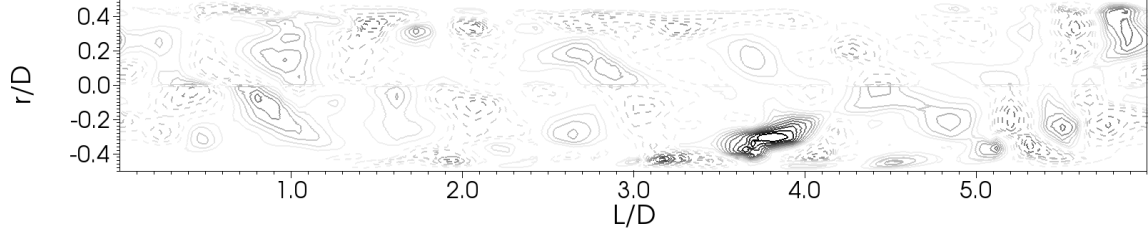
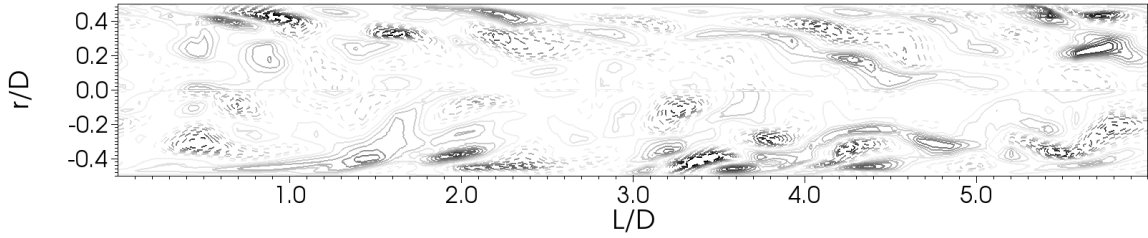


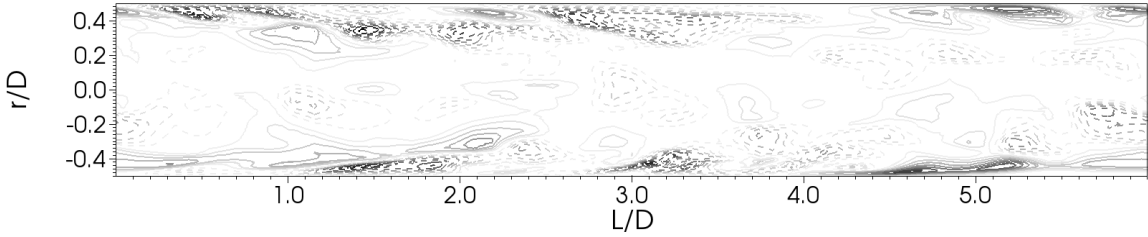
Figure 3.23: u_{rms}^+ profile of turbulent pipe flow. The symbols represent the data presented in [142] while the lines represent the data obtained from the simulation discussed here. The T and S subscripts refer to the present two-mesh and the single mesh simulations, respectively. Refer to the caption of figure 3.22 for parameter specifications and details regarding abbreviations.



(a) Radial Velocity Fluctuations



(b) Azimuthal Velocity Fluctuations



(c) Axial Velocity Fluctuations

Figure 3.24: Contours of fluid velocity fluctuations in a length-wise cross section of the pipe at non-dimensional time of 3000. Solid and dashed contours represent positive and negative velocity fluctuations, respectively. The darkest contour lines in the radial (a) and azimuthal (b) velocity fluctuation plots represent values of 0.2 and -0.2 for solid and dashed, with step size of 0.02 between contours. The darkest contour lines in axial velocity fluctuation plot (c) represent values of 0.45 and -0.45 for solid and dashed, with step size of 0.045 between contours. Refer to the caption of figure 3.22 for parameter specifications.

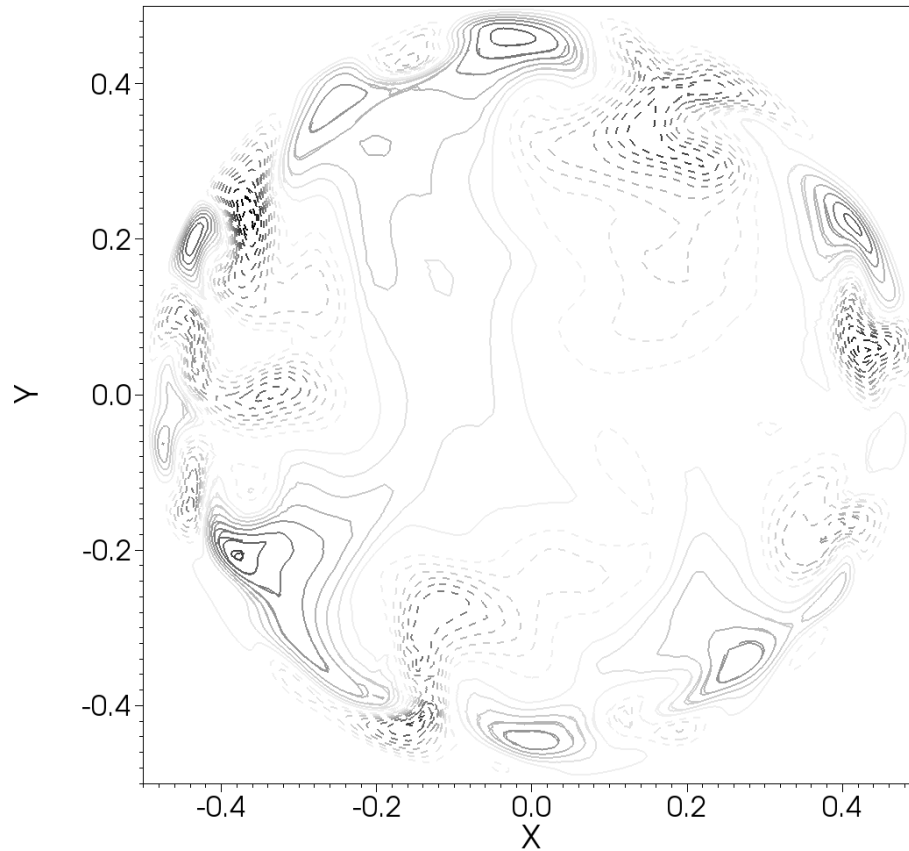


Figure 3.25: Contours of axial fluid velocity fluctuations in a cylindrical cross section of the pipe. Refer to the caption of Figure 3.24 for details regarding axial velocity fluctuation contours.

MOVING OVERLAPPING MESH VALIDATION

The capability of the moving overlapping grid method will be demonstrated through two- and three-dimensional validation simulations. The spatial and temporal accuracy of the scheme will be displayed, and the method's ability to realistically model fluid flow when influenced by moving rigid bodies is illustrated.

4.1 Convecting Two-Dimensional Eddies

The spatial and temporal convergence rates were determined with a simulation of convecting eddies in a periodic global domain, comprised of two subdomains, for which an exact solution exists [137]. Walsh [137] proposed an exact solution to the incompressible Navier-Stokes equations for non-convecting eddies with initial conditions $\tilde{\mathbf{u}}(\mathbf{x}, 0) = \mathbf{a}(\mathbf{x})$. Specifically, if $\Delta \mathbf{a} = \lambda \mathbf{a}$ and $\nabla \cdot \mathbf{a} = \mathbf{0}$ in Ω , then $\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{t}) = \mathbf{e}^{\nu \lambda \mathbf{t}} \mathbf{a}(\mathbf{x})$, where ν is the kinematic viscosity, with pressure determined by $\nabla \tilde{p}(\mathbf{x}, t) = -\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{t}) \cdot \nabla \tilde{\mathbf{u}}(\mathbf{x}, \mathbf{t})$. The tilde indicates that the values are taken in the frame of reference where the eddies' locations are stationary in time.

We can express the eigenvalues as $\lambda = -(n^2 + m^2)$, where n and m are non-negative integers, which gives eigenfunctions that are products of $\text{Sin}[n\tilde{x}]$ or $\text{Cos}[n\tilde{x}]$ with $\text{Sin}[m\tilde{y}]$ or $\text{Cos}[m\tilde{y}]$. Stream functions that solve the governing equations are formed using linear combinations of the eigenfunctions. For convecting eddies, the solution is taken from the frame of reference where the eddies remain stationary, which is accomplished through the coordinate transformation $x = \tilde{x} + u_0 t$, $y = \tilde{y} + v_0 t$, where (u_0, v_0) is the convection velocity. For the present verification simulations, the

following velocity solution is used

$$\begin{aligned} u(x, y, t) &= e^{-25t\nu}(-\cos[5(y - v_0t)] + \cos[4(y - v_0t)]\sin[3(x - u_0t)]) + u_0, \\ v(x, y, t) &= e^{-25t\nu}(-\sin[5(x - u_0t)] - \frac{3}{4}\cos[3(x - u_0t)]\sin[4(y - v_0t)]) + v_0. \end{aligned} \quad (4.1)$$

The solution for pressure is determined from $\nabla \tilde{p}(\mathbf{x}, t) = -\tilde{\mathbf{u}}(\mathbf{x}, t) \cdot \nabla \tilde{\mathbf{u}}(\mathbf{x}, t)$, a full expression and a detailed derivation is found in [135]. Zero mean values for exact and computational pressure solutions are used for comparison, since the incompressible governing equations allow for any arbitrary mean pressure.

4.1.1 Simulation Configuration

The global $2\pi \times 2\pi$ two-dimensional domain is decomposed into an interior and exterior mesh as discussed in Section 3.1 for stationary subdomains. The exterior mesh contains a vacancy at its center which is covered by the interior mesh. Two configurations of interior grids were used, a circular mesh and a square mesh, as seen in Figure 3.1. In all simulations the exterior mesh was held stationary and the interior mesh was constrained to move in a prescribed fashion. Note that this motion is a type of “pseudo”, or “virtual”, motion as the movement of the mesh should have no effect on the flow since there are no solid structures contained within. In the first set of verification cases, both circular and square interior meshes were constrained to rotate, while in succeeding simulations sliding motion was prescribed.

Simulations were performed with $Re = 20$, where the Reynolds number is defined, $Re = L^*U^*/\nu^*$, with L^* the reference length, U^* the reference velocity, and ν^* the kinematic viscosity. Quantities with stars denote the dimensional quantities, and the ones without stars are non-dimensional. All non-dimensional length and velocity variables are given here, with the presumption that corresponding dimensional counterparts were normalized by L^* taken here as $L^* = L_x^*/(2\pi)$, where L_x^* is the domain

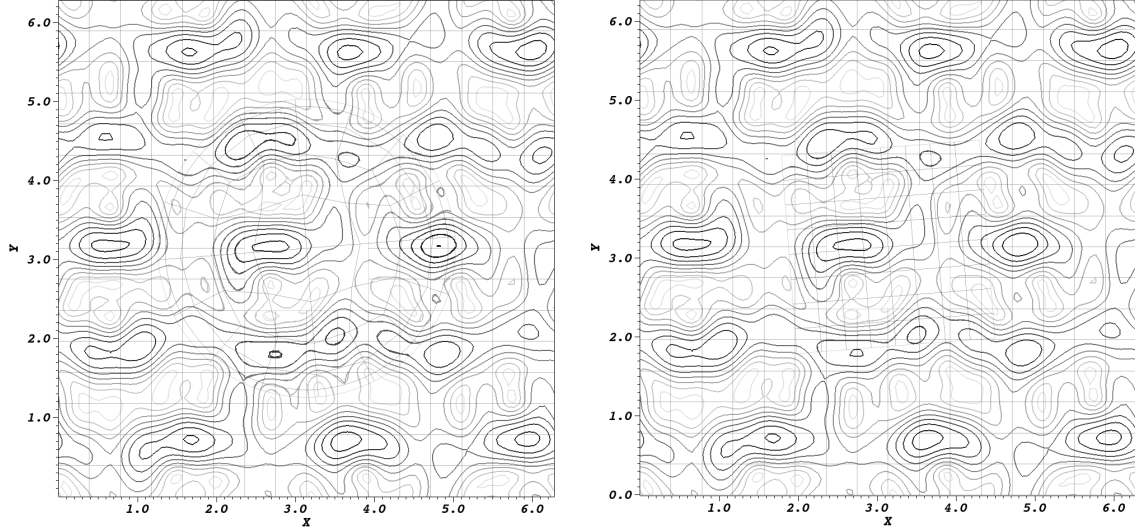
length in x direction, and by U^* equal to the eddy convection velocity in x -direction as discussed below.

4.1.2 Rotating Mesh Accuracy

The first set of validation simulations investigates the spatial and temporal accuracy of our moving overlapping mesh method with rotating grids. Interior meshes were rotated counter-clockwise as rigid bodies about their center of mass, which also coincided with the center of the global domain $[\pi, \pi]$, with non-dimensional angular velocity $\Omega = \Omega^* L^* / U^* = \pi/4$.

In Figure 4.1, velocity magnitude contours are shown for rotating interior mesh simulations with fluid convection velocity $u_0 = u_0^* / U^* = 1$ and $v_0 = v_0^* / U^* = 0.3$, one with the circular and one with the square interior mesh. This particular convection velocity was chosen so that direct comparison with the results of stationary overlapping mesh case published in [135] was possible. Note that in the overlap region, two sets of contour lines are presented, although the continuity of values between the two subdomains is such that discrepancies between values in the overlap region are difficult to visually detect.

Results are compared with data from the previously validated overlapping mesh methodology for stationary meshes, where Figure 4.2 shows velocity errors with respect to p -refinement. Results from the moving overlapping mesh methodology closely correlate with the stationary overlapping mesh data. The exterior mesh data achieves full convergence at 14th-order polynomial approximations in circular interior mesh simulations, and 13th-order polynomial approximations for square interior mesh simulations, using the chosen non-dimensional timestep of $\Delta t = U^* \Delta t^* / L^* = 1 \times 10^{-4}$. Circular interior mesh simulations converge slightly slower (although still exponentially) than the square interior mesh simulations. This is a result of the slightly lower



(a) Circular Interior Mesh

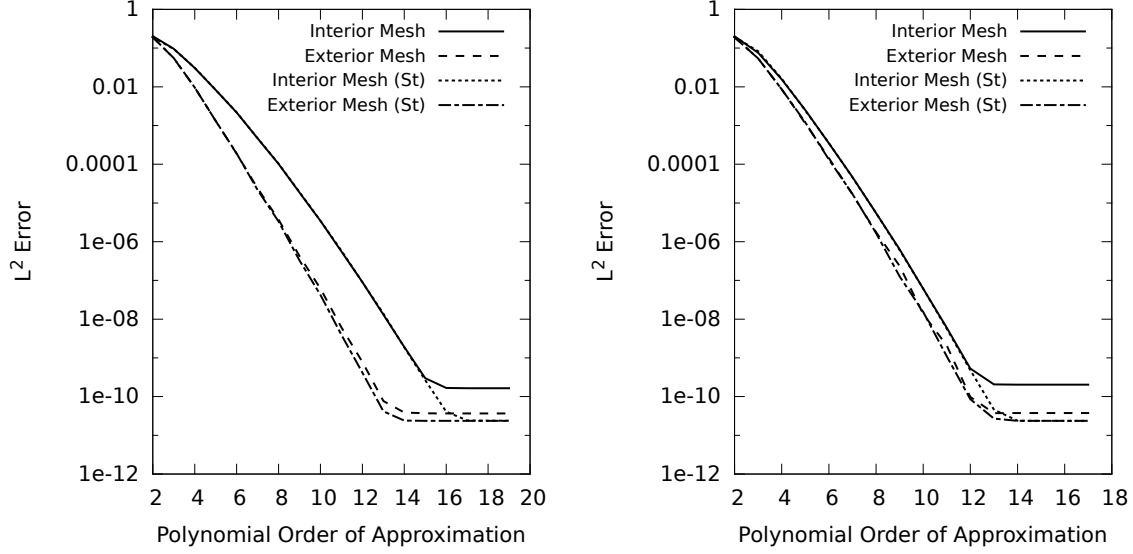
(b) Square Interior Mesh

Figure 4.1: Velocity magnitude contours of rotating interior mesh simulations at 1000 timesteps. The darkest contour lines represent a non-dimensional velocity magnitude of three, while the lightest lines, zero. The simulations were performed with 8th-order polynomial approximations, and $\Delta t = U^* \Delta t^* / L^* = 1 \times 10^{-4}$ using second order temporal accuracy and IEXT2, with two iterations per timestep.

convergence rate of the SEM with curvilinear meshes [138, 57, 139, 135], and not a result of the moving overlapping mesh methodology. In spite of slightly different slope values, each mesh in each case achieves spectral spatial convergence, using spectral Lagrangian interpolation at moving subdomain interfaces.

Temporal accuracy of the same rotating interior mesh configuration is presented in Figure 4.3. The accuracy in each case achieves the expected third-order convergence rate, where the four iterations for IEXT3 were used.

Pressure accuracy also achieves spectral spatial convergence and the desired third-order temporal convergence rate for each subdomain in both rotating circular and square interior mesh cases. Figure 4.4 shows the data for the case with a rotating square interior mesh.



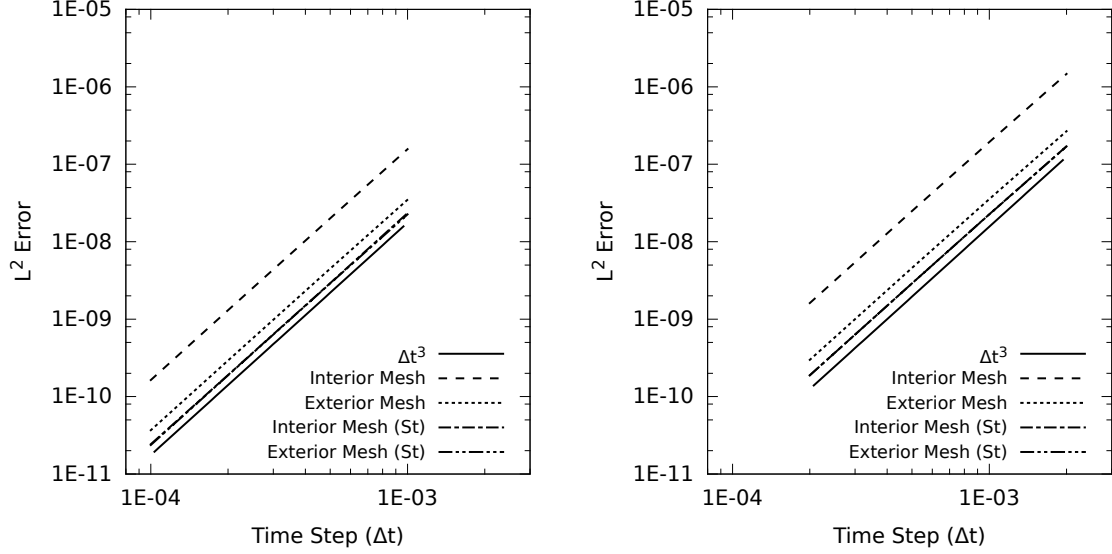
(a) Circular Interior Mesh

(b) Square Interior Mesh

Figure 4.2: Velocity errors with respect to changing solution polynomial order. The tests were performed with timestep $\Delta t = 1 \times 10^{-4}$ using BDF3 and IEXT3, with four iterations per timestep. The (St) denotes data from stationary overlapping mesh simulations. The interior mesh in both sets of tests was rotated with non-dimensional angular velocity $\Omega = \pi/4$, and errors were collected after 1000 timesteps.

4.1.3 Sliding Mesh Accuracy

Accuracy of data from sliding interior mesh simulations is comparable to that of the rotating simulations discussed above. With sliding mesh simulations for this configuration, one needs to be careful that the interior mesh does not slide out of the vacancy, because interface boundaries would then become Dirichlet boundaries which would need to be prescribed - a situation which is possible to accommodate, but was not desirable for a consistent convergence testing in the present simulations. With the sliding mesh velocity of $w = w^*/U^* = 1.5$ used in the current test case, the square interior mesh, for example, would begin to slide out of vacancy at $t \approx 0.733$. For the determining the accuracy of the method, which is the goal here, errors are recorded at time $t = 0.1$. Data for sliding square mesh cases are presented here, but note that sliding circular mesh simulations yield similar results, and conclusions drawn for



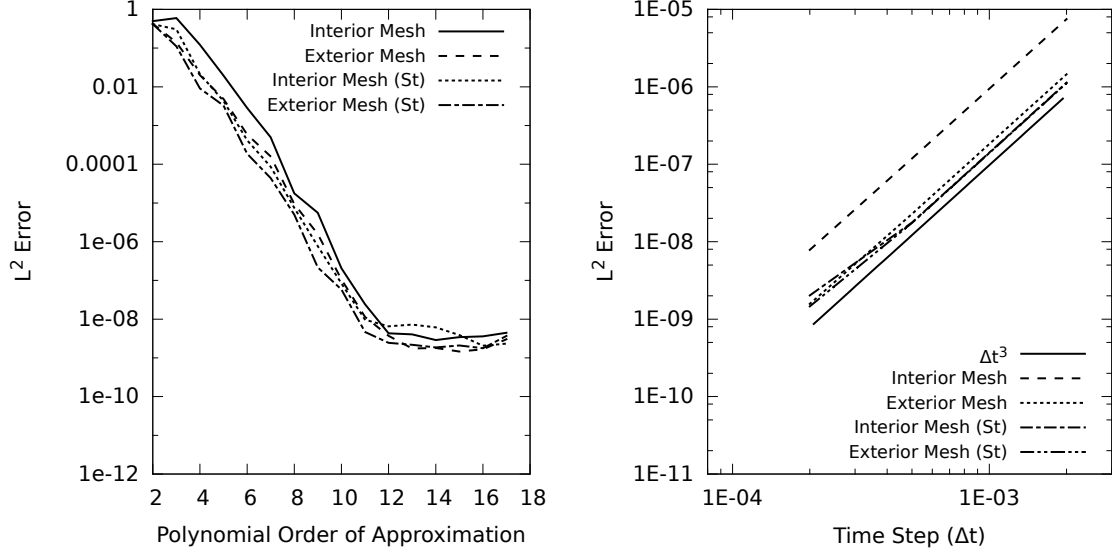
(a) Circular Interior Mesh

(b) Square Interior Mesh

Figure 4.3: Velocity errors with respect to changing timestep. The tests were performed with polynomial order of 17, and used BDF3 and IEXT3, with four iterations per timestep. Errors were collected at non-dimensional time $t = 0.1$. The (St) denotes data from stationary overlapping mesh simulations, and a reference line is displayed for comparison to the expected convergence rate. The interior mesh in both sets of tests was rotated with non-dimensional angular velocity $\pi/4$.

the former can also be applied to the latter. Figure 4.5 illustrates the spatial and temporal convergence of convecting eddy simulations with the interior mesh sliding to the right with velocity $w = 1.5$.

Similar to the rotating mesh cases, the results in the exterior mesh fully converge at 13th-order polynomial approximations, and interior mesh at 14th-order polynomial approximations for the chosen timestep $\Delta t = 1 \times 10^{-4}$. The temporal accuracy plot shows that the expected third-order temporal convergence is maintained using third-order interface extrapolation (four iterations were performed). Pressure data also exhibits spectral spatial accuracy and third-order temporal accuracy as illustrated in Figure 4.6. Note that the errors for the sliding mesh simulations are actually slightly

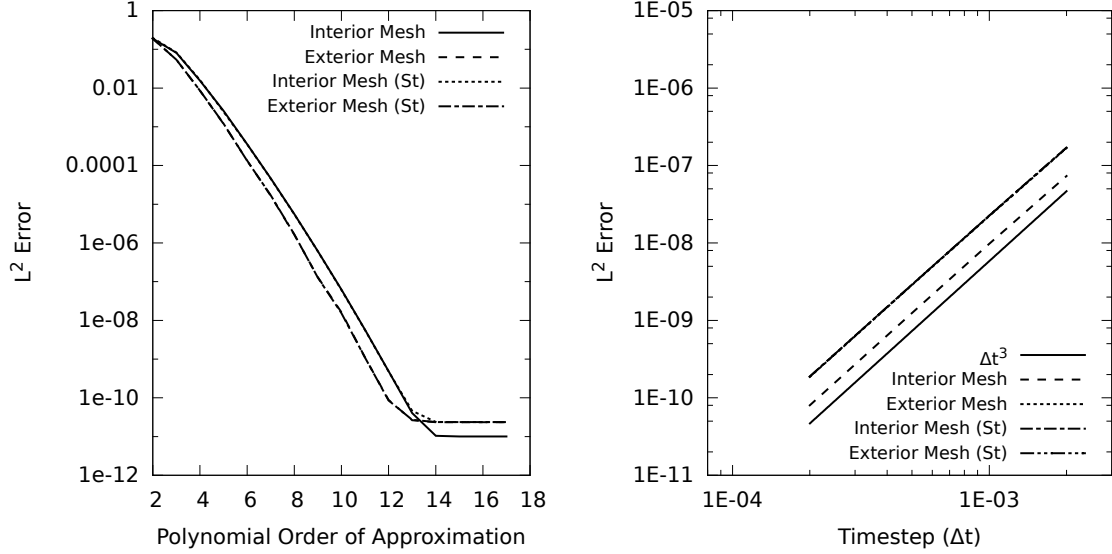


(a) Spatial Convergence

(b) Temporal Convergence

Figure 4.4: Pressure errors for square interior mesh simulations. All tests were performed using BDF3 and IEXT3 with four iterations per timestep. The spatial convergence tests were performed with timestep $\Delta t = 1 \times 10^{-4}$ and results were collected after 1000 timesteps. The temporal convergence tests were performed with 17th-order polynomials and results were collected at $t = 0.1$. The (St) denotes data from stationary overlapping mesh simulations. The interior mesh in both sets of simulations was rotated with non-dimensional angular velocity $\Omega = \pi/4$.

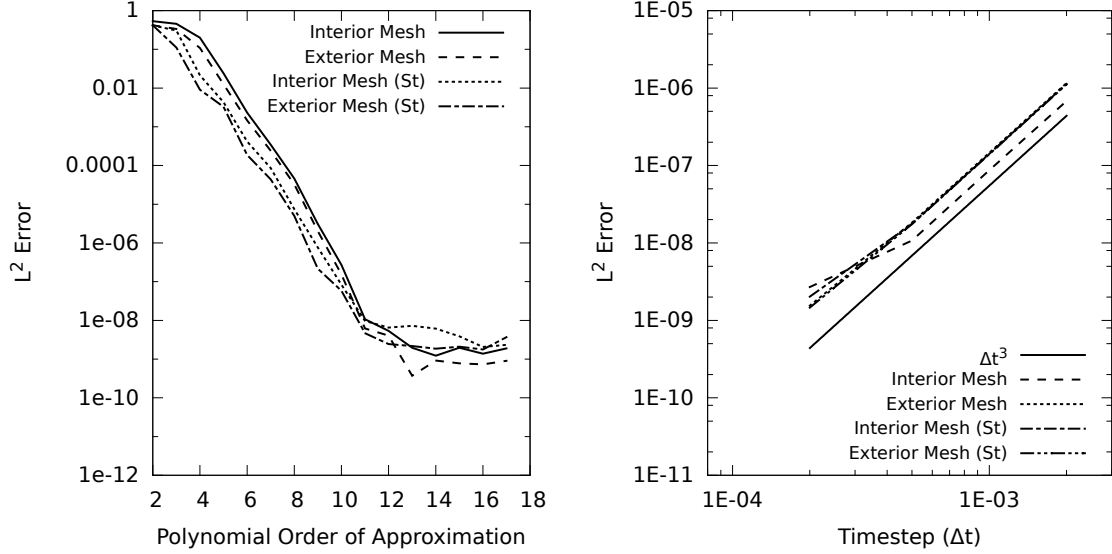
lower than in a corresponding overlapping stationary mesh case, a fact that will be explained shortly.



(a) Spatial Convergence

(b) Temporal Convergence

Figure 4.5: Velocity errors for square sliding interior mesh simulations. The spatial accuracy tests were performed with $\Delta t = 1 \times 10^{-4}$ (errors collected after 1000 timesteps), and the temporal accuracy tests were performed with polynomial order of 17 (errors collected at $t = 0.1$). Each case used BDF3 and IEXT3, with four iterations per timestep. The (St) denotes data from stationary overlapping mesh simulations [135], and a reference line is displayed in the temporal accuracy plot for comparison to the expected convergence rate. The interior mesh was prescribed a sliding velocity of $w = 1.5$ to the right.



(a) Spatial Convergence

(b) Temporal Convergence

Figure 4.6: Pressure errors for square sliding interior mesh tests. The spatial accuracy tests were performed with $\Delta t = 1 \times 10^{-4}$ (errors collected after 1000 timesteps), and the temporal accuracy tests were performed with polynomial order of 17 (errors collected at $t = 0.1$). Each case used BDF3 and IEXT3, with four iterations per timestep. The (St) denotes data from stationary overlapping mesh simulations [135], and a reference line is displayed in the temporal accuracy plot for comparison to the expected convergence rate. The interior mesh was prescribed a sliding velocity $w = 1.5$ to the right.

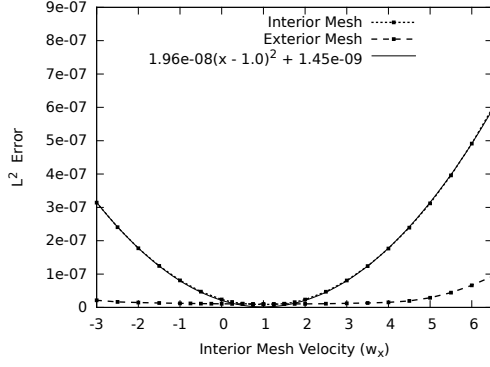
4.1.4 Influence of Mesh Velocity

The value of mesh velocity introduces another possible parameter affecting global error. Figures 4.7 and 4.8 show how the errors are affected by the velocity of the moving interior mesh. In all sliding mesh cases the velocity error is at a minimum when the mesh velocity (w_x) is equal to the convection velocity (u_0). In cases using the second order temporal scheme, the errors in the moving mesh increase quadratically with the absolute value of the relative mesh velocity ($|w_x - u_0|$), and in cases using first order timestepping, the errors increase nearly linearly. The errors seen in the stationary exterior mesh remain fairly constant since nothing changes in each subsequent test except for the values interpolated at its interface boundaries.

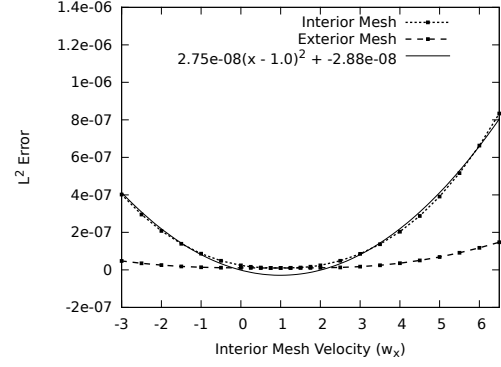
We see similar quadratic and linear increase in velocity errors for the rotating interior mesh tests as the angular velocity (w_Ω) of the interior mesh increases (Figure 4.8). Minimum errors in these rotating mesh tests are found when $w_\Omega = 0$. While errors increase with moving mesh velocity, they generally remain of the same order of magnitude in the range of mesh velocities investigated.

Note that although the absolute value of the errors increase, spatial errors still converge exponentially, and temporal errors with third-order, as illustrated in Figure 4.9 for sliding square interior mesh simulations.

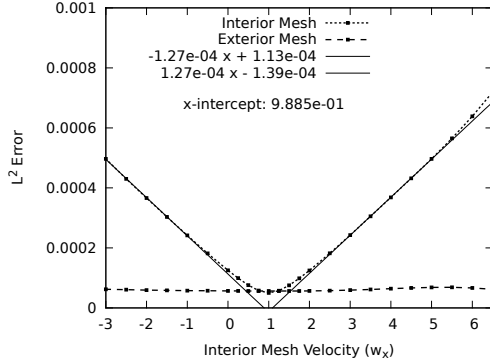
Recall that looking at eddies convecting to the right with velocity u_r in a stationary frame of reference is analogous to stationary eddies being viewed in a frame moving to the left with velocity u_r . Thus, varying the fluid convection velocity (u_0) should yield similar results to varying the mesh velocity (w_x). In Figure 4.10 the errors reported by the interior mesh for two sets of simulations are compared, in one set the interior mesh velocity is set to zero and the fluid convection velocity (u_0) is changed, and in the other set, u_0 is kept at 0 and w_x is changed. The comparison is made by plotting



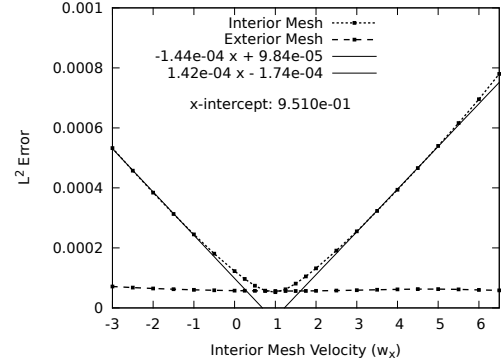
(a) Sliding Interior Square Mesh using BDF2/IEXT2



(b) Sliding Interior Circular Mesh using BDF2/IEXT2



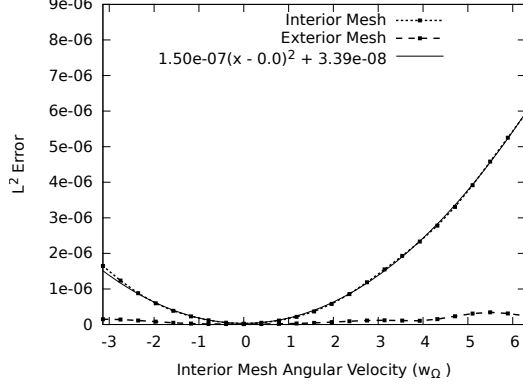
(c) Sliding Interior Square Mesh using BDF1/IEXT1



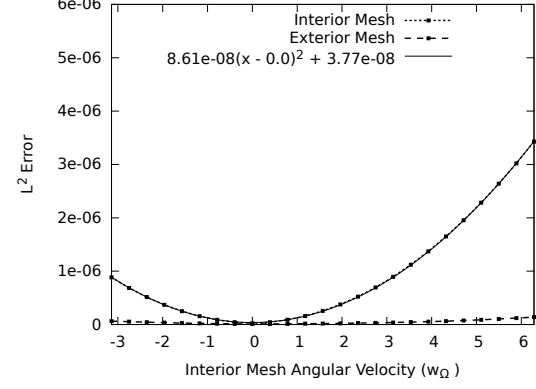
(d) Sliding Interior Circular Mesh using BDF1/IEXT1

Figure 4.7: Velocity errors with respect to sliding mesh velocity. 14th-order polynomials were used for solution approximations to ensure full spatial convergence and a timestep of $\Delta t = 1 \times 10^{-4}$ was used. Errors were collected at $t = 0.1$. Convection velocity was set: $u_0 = 1$ and $v_0 = 0$.

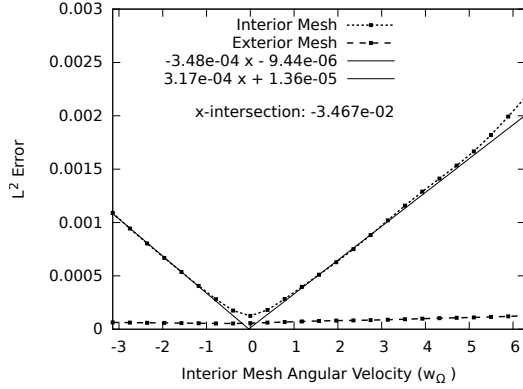
errors against the relative velocity ($u_r = u_0 - w_x$) for the two simulation sets. We see that the two sets of data indeed give very similar error results, though we see that at large relative velocities, u_r , the sliding mesh simulations report slightly lower errors than stationary mesh simulations. This occurs since the exterior mesh maintains slightly higher accuracy in sliding mesh simulations because it remains stationary (having a smaller relative velocity), thus the interior mesh, in these cases, interpolate



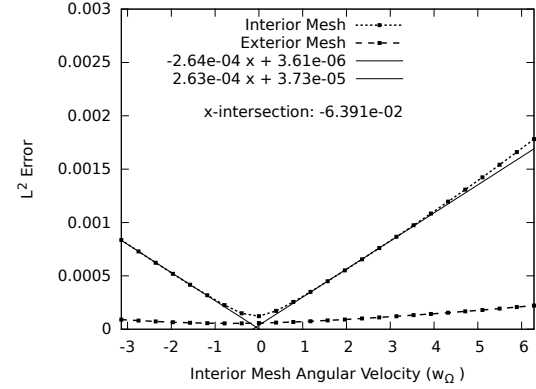
(a) Rotating Interior Square Mesh using BDF2/IEXT2



(b) Rotating Interior Circular Mesh using BDF2/IEXT2



(c) Rotating Interior Square Mesh using BDF1/IEXT1

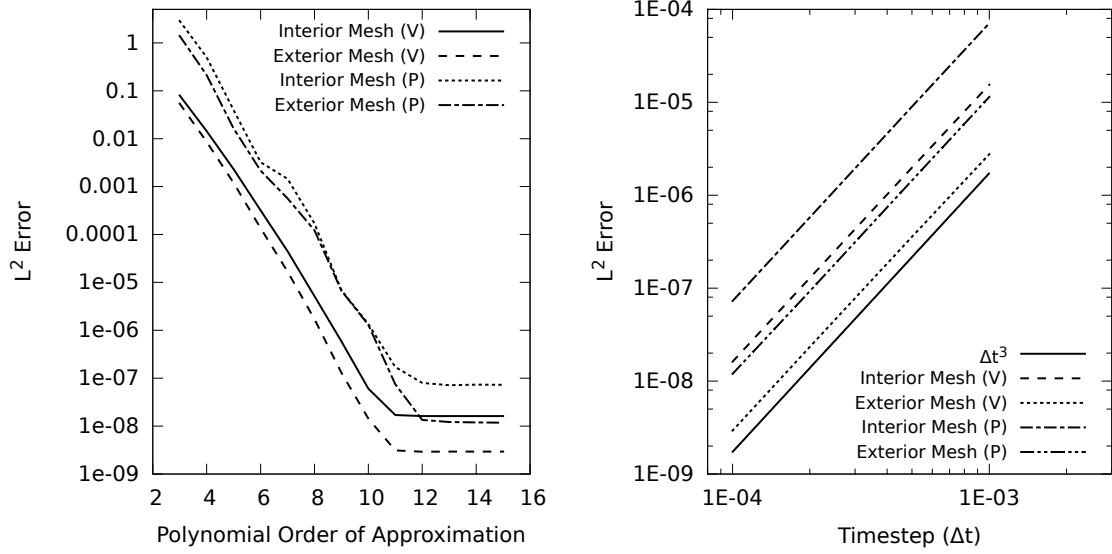


(d) Rotating Interior Circular Mesh using BDF1/IEXT1

Figure 4.8: Velocity errors with respect to rotating mesh angular velocity. 14th-order polynomials were used for solution approximations to ensure full spatial convergence and a timestep of $\Delta t = 1 \times 10^{-4}$ was used. Errors were collected at $t = 0.1$. Convection velocity was set: $u_0 = 1$ and $v_0 = 0$.

values at their interface boundaries that are more accurate, consequently improving the solution accuracy.

We see from Figures 4.7 and 4.8 that the rate at which errors increase is dependent upon the order of the temporal scheme used. In general, the Courant number, expressed using the one-dimensional formulation for simplicity, in moving subdomains can be expressed as $C = u_r \Delta t / \Delta x$, where each of the values $u_r = |u - w|$, w , and



(a) Spatial Convergence

(b) Temporal Convergence

Figure 4.9: Velocity (V) and pressure (P) errors for square sliding interior mesh tests, with prescribed sliding velocity $w = 6.5$ to the right. The spatial accuracy tests were performed with $\Delta t = 1 \times 10^{-4}$ (errors collected after 1000 timesteps), and the temporal accuracy tests were performed with polynomial order of 17 (errors collected at $t = 0.1$). Each case used BDF3 and IEXT3, with four iterations per timestep. A reference line is displayed in the temporal accuracy plot for comparison to the expected convergence rate. Convection velocity is the same as the convergence rate tests discussed above ($u_0 = 1.0, v_0 = 0.3$).

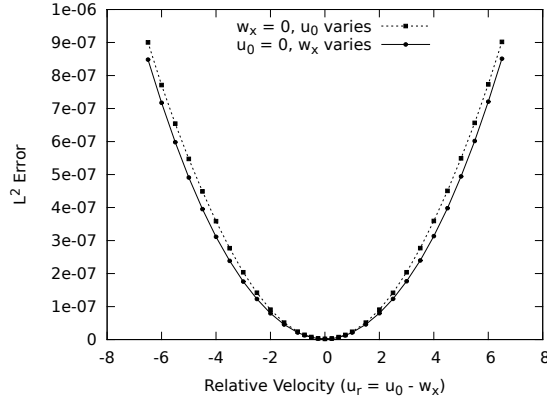
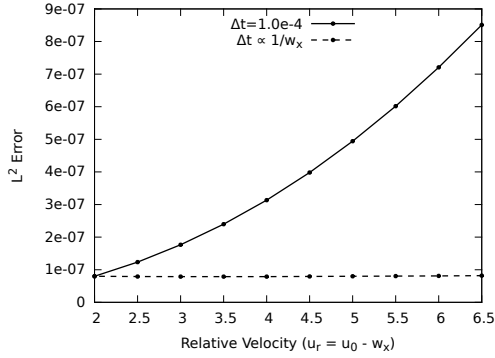
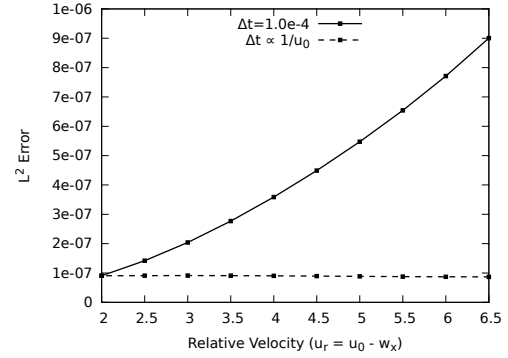


Figure 4.10: Velocity errors in the interior mesh with respect to relative velocity. Simulations were performed with 14th-order polynomial approximations and $\Delta t = 1 \times 10^{-4}$. Second-order temporal scheme and IEXT2 were used with two iterations.

Δx are taken at individual GL points. The value of the relative velocity u_r will be dominated by the mesh velocity, w , for large enough w values. Thus by changing Δt inversely proportionally to w maintains the value for the Courant number, and the errors then remain quite constant as illustrated in Figure 4.11. The same result is accomplished by changing Δt inversely proportionally to the convection velocity, u , in a stationary mesh case, as illustrated in Figure 4.11.



(a) Sliding Square Interior Mesh with $u_0 = 0$



(b) Stationary Square Interior Mesh ($w_x = 0$)

Figure 4.11: Velocity errors in the interior mesh with respect to relative velocity, u_r . Simulations were performed with 14th-order polynomial approximations with the second-order temporal scheme and IEXT2 were used with two iterations. Simulations represented by the solid line were all performed with $\Delta t = 1 \times 10^{-4}$, and the dashed lines represent data from simulations where Δt was varied inversely proportional to the mesh velocity (a) or fluid convection velocity (b). Thus, for $u_r = 2$, $\Delta t = 1 \times 10^{-4}$ and for $u_r = 4$, $\Delta t = 5 \times 10^{-5}$ when considering the dashed lines.

4.2 Two-Dimensional Oscillating Cylinder

The fluid flow around moving cylinders has been a subject of great interest since the early research of Magnus [143] in 1852, and since then, the topic of oscillating, or vibrating, cylinders in cross-flow has been the focus of several experimental and computational studies [144, 2, 145, 146, 147]. In the 1960's, Koopmann [2] and others performed several experiments regarding uniform flow past oscillating cylinders and established conditions “for which the vortex wake frequency is controlled by the driving frequency of the cylinder”. Koopmann’s findings regarding ‘lock-in’ frequencies have become a standard by which others have validated the use of new experimental and computational methods. In 1998, Blackburn et al. [1] published detailed results regarding the forces acting on an oscillating cylinder in steady crossflow using simulations that utilized a spectral element code, similar to the current method, although performed with a single mesh in a moving reference frame. This published data established detailed characteristics of the flow which allows for comparison and validation of the present work. The following data seeks to illustrate that the moving overlapping mesh methodology accurately describes the fluid flow interacting with two-dimensional moving solid bodies.

For the present oscillating cylinder simulations, the domain is assigned a uniform free stream velocity U_∞^* at the left boundary with outflow conditions assigned on the opposing side. Outflow conditions are defined, in non-dimensional form,

$$p \hat{\mathbf{n}} - \frac{1}{\text{Re}} \nabla \mathbf{u} \cdot \hat{\mathbf{n}} = 0 \quad \text{on } \partial\Omega_O, \quad (4.2)$$

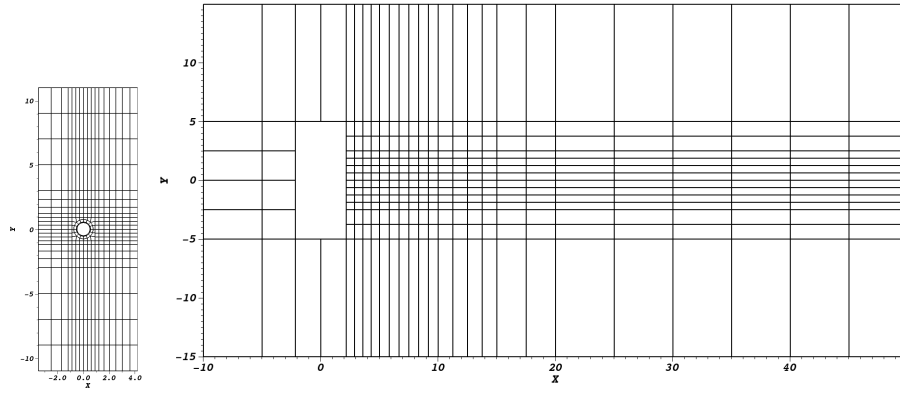
where $\hat{\mathbf{n}}$ is the normal unit vector and $\partial\Omega_O$ is the outflow boundary. Here, all spatial variables are non-dimensionalized with the cylinder diameter D^* , and velocity variables with the free-stream velocity U_∞^* , with $\text{Re} = U_\infty^* D^* / \nu^*$. Symmetry boundary conditions are imposed on the top and bottom of the domain. The horizontal velocity

of the cylinder is set to zero, while the vertical component of motion is governed by the equation

$$y(t) = y_0 + A \sin(2\pi f_0 t), \quad (4.3)$$

where $A = y_{max}^*/D^*$ (y_{max}^* is the largest vertical displacement of the cylinder) represents the non-dimensional amplitude and $f_0 = f_0^* D^*/U_\infty^*$ represents the non-dimensional frequency of oscillation. The cylinder is initially placed at (0,0) such that $y_0 = 0$ in (4.3), with the global domain spanning from $x = -10$ to 50 diameters, and $y = -15$ to 15 diameters. We also define the frequency ratio $F = f_0^*/f_v^*$ where f_v^* is the vortex shedding frequency of the *fixed* cylinder.

The global domain is decomposed into two overlapping subdomains, with the exterior mesh (see Figure 4.12) containing a vacancy for cylinder movement, and the interior mesh containing the 2D cylinder. During the simulations, the exterior mesh remains stationary while the interior mesh is constrained to move with the equation of oscillation given by (4.3). The interior mesh is constructed not to slide out of the global domain or out of the vacancy during the simulations.



(a) Interior

(b) Exterior Mesh

Mesh

Figure 4.12: Geometry of the oscillating cylinder case, with element boundaries shown

4.2.1 Stationary Cylinder

For accurate comparison with published data, the Strouhal number, or vortex shedding frequency, from flow around the *stationary* cylinder was first determined using our previously validated overlapping mesh methodology (for stationary meshes) [135]. Strouhal number is given as $St \equiv f_v^* D^* / U_\infty^*$, or just f_v with the current normalization. At Reynolds number of 200, our simulations produced a Strouhal number of 0.1998 compared with the Strouhal number found in Udaykumar et al. [147] of 0.198 and the value found in Williamson et al. [146] of 0.197. The mean drag of our simulation was 1.372, while Udaykumar et al. [147] found a value of 1.38, both at $Re = 200$.

Table 4.1 compares the values from our overlapping mesh methodology for fixed meshes, at $Re = 500$, with values published by Blackburn et al. [1] for the stationary cylinder in uniform flow, also with $Re = 500$. The Strouhal number is denoted by St , while \hat{C}_l represents the peak coefficient of lift, $\overline{C_d}$ is the mean (time-averaged) coefficient of drag, and $\overline{C_{pb}}$ is the mean base pressure coefficient calculated from the pressures at the furthest upstream (p_0^*) and downstream (p_{180}^*) points on the cylinder surface, $C_{pb} = 1 + 2(p_{180}^* - p_0^*) / \rho^* U_\infty^{*2}$. The drag and lift on a stationary cylinder in a uniform cross-flow changes with time due to vortex shedding. After a sufficient startup time, the solution reaches an asymptotic state, indicated by values that repeat periodically. The present simulation reached its asymptotic state at $t = t^* U_\infty^* / D^* \approx 100$, although averages were taken over the interval $300 \leq t \leq 600$.

	Blackburn et al. [1]	Present Data
St	0.2280	0.2281
\hat{C}_l	1.200	1.202
$\overline{C_d}$	1.460	1.461
$-\overline{C_{pb}}$	1.506	1.504

Table 4.1: Results for a stationary cylinder in uniform flow compared with data presented in [1]. The present data used 7th order polynomial approximations. Second order timestepping and IEXT2 were used with two iterations per timestep. Present averages were taken over the interval $300 \leq t \leq 600$. In the present simulation overlapping meshes were used.

4.2.2 Oscillating Cylinder

Figure 4.13 depicts the x -direction velocity contours of vortices shed from an oscillating cylinder as they travel from the interior mesh across the interface boundary to the exterior mesh. The overlap region contains two sets of contour lines, and visual inspection shows good continuity between the values reported by the two meshes in the overlap region.

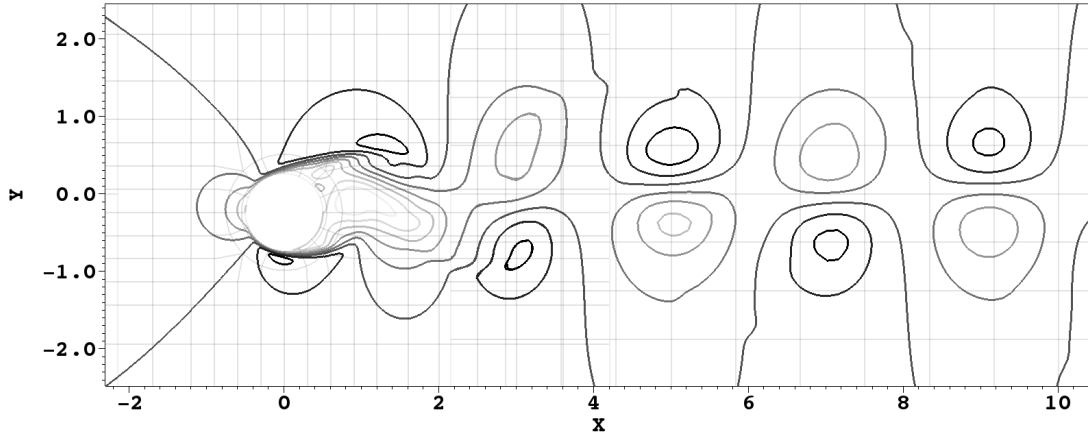


Figure 4.13: Enlarged plot of x -direction velocity contours of oscillating cylinder simulation at $Re=500$ with $A=0.25$ and $F=1.0$, at $t = 100$. The lightest lines represent velocity of $u^*/U_\infty^* = -0.5$ while the darkest lines represent $u^*/U_\infty^* = 1.5$. Solution approximations are calculated using 7th-order polynomials, with $\Delta t = 2.5 \times 10^{-3}$, second-order global timestepping scheme and IEXT2 with two iterations.

The results given in [1] include detailed information about the mean drag forces, the peak lift forces, and the mean pressure differences on the oscillating cylinder, which are compared in Table 4.2 for a frequency ratio of 1.0 and non-dimensional amplitude 0.25, at $Re = 500$ for both sets of data (illustrated in Figure 4.13). The present simulation reached its asymptotic state at $t \approx 75$, and the averages were taken over the time interval $250 \leq t \leq 550$. The frequency of vortex shedding, the forces and pressure on the cylinder, and other flow characteristics correlate well with published data.

	Blackburn et al.	Present Data
\widehat{C}_l	1.776	1.781
$\overline{C_d}$	1.414	1.417
$-\overline{C_{pb}}$	1.377	1.377

Table 4.2: Results for an oscillating cylinder with $\text{Re} = 500$, $F = 1.0$ and $A = 0.25$ in uniform flow compared with results in [1]. Present simulations were performed with 7th-order polynomial approximations, using second-order timestepping and IEXT2 with two iterations per timestep, and the averages were taken over the interval $250 \leq t \leq 550$.

As mentioned above, Koopmann [2] observed certain frequencies and amplitudes of oscillation that enabled ‘lock-in,’ where the vortex shedding frequency is equal to the oscillation frequency of the cylinder. Outside of this lock-in region, the vortex shedding frequency converges to the fixed-cylinder vortex shedding frequency denoted as Strouhal frequency [2, 147]. Lock-in (or non lock-in) behaviour of the vortex shedding is a result of the relationship between the vortex formation near the surface of the cylinder and the changing velocity and acceleration of the cylinder with respect to the fluid. Differing lateral fluid velocities resulting from the different amplitudes and frequencies of oscillation in each case cause portions of the forming vortices to detach at different positions with respect to the cylinder’s cycle. An extensive analysis of this phenomena is found in [145]. Figure 4.14 shows lines that represent the lock-in region established by Koopmann [2], with data from our moving overlapping mesh method showing which simulations exhibited this lock-in behavior and which didn’t, both experiments [2] and simulations used a Reynolds number of 200. Simulations were performed with a non-dimensional amplitude ($A = y_{max}^*/D^*$) of 0.1 and 0.2 with various oscillation frequencies. The present data displays good correlation with the experimental results reported in [2]. Present simulations reached their asymptotic

states at $t \approx 75$ after which the vortex shedding frequency was monitored for ≥ 100 non-dimensional time units to determine if lock-in behavior was exhibited.

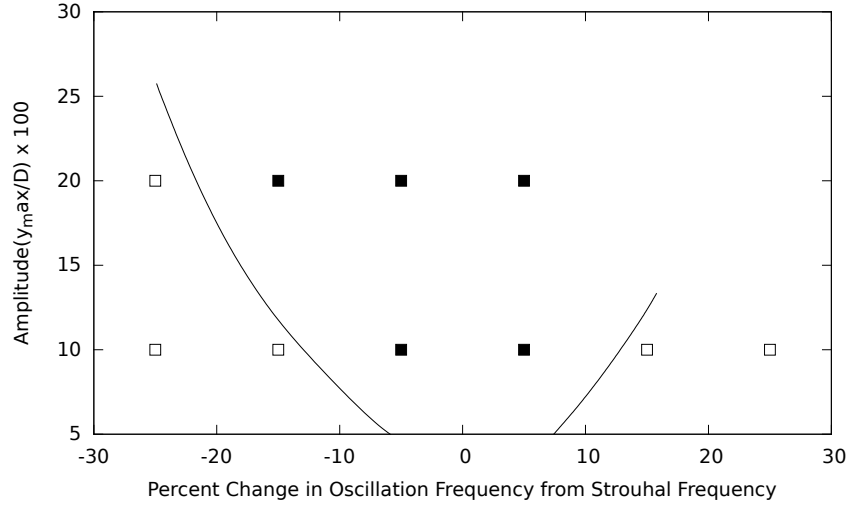


Figure 4.14: 'Lock-In' data compared with experimental data from Koopmann [2]. The region inside the solid lines represents Koopmann's lock-in region, the squares represent simulations performed using the moving overlapping mesh method. The filled squares are those simulations that displayed lock-in, while the hollow squares are those simulations that did not exhibit lock-in behavior. All simulations were performed with 7th-order polynomial approximations, using $\Delta t = 2.5 \times 10^{-3}$ with second-order global temporal scheme and IEXT2 with two iterations per timestep. $Re = 200$ was used for both sets of data.

We conclude that data from the present moving overlapping mesh methodology gives results for two dimensional moving bodies interacting with fluid flow that accurately correlates with other experimental and computational data. The present methodology gives the expected forces and pressure on the cylinder, as well as accurate vortex shedding frequency, and the presented lock-in data matches the experimental findings of Koopman [2].

Although relatively small amplitude regimes documented in this section potentially could be computed with a single deforming mesh, the current two-mesh method can be applied to more extreme cases, such as the one shown in Figure 4.15 with very large amplitude of oscillations, $A=2.5$, that would be difficult to handle with a single

mesh due to extreme mesh distortion. This case, to the author's knowledge, has not been studied before, and data for comparison does not exist.

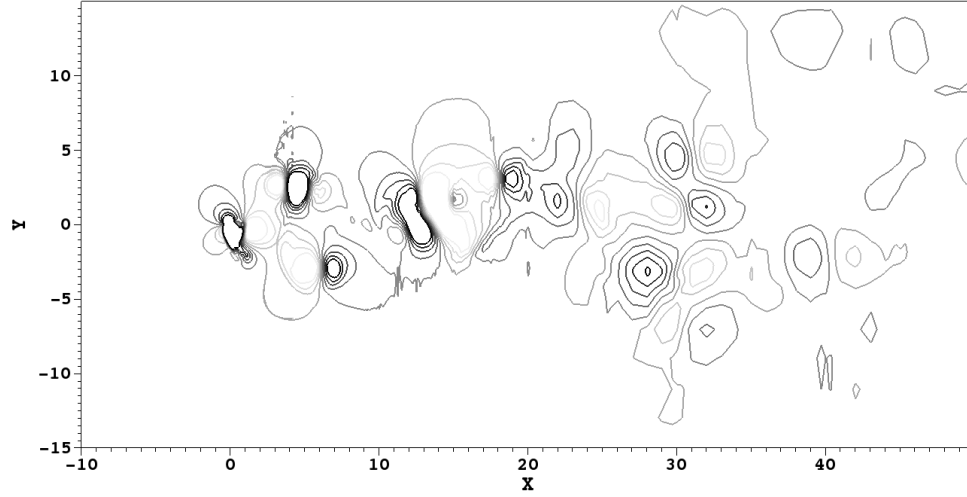


Figure 4.15: Plot of y-direction velocity in the global domain for a cylinder oscillating with $A = 2.5$, $F = 1$, and $\text{Re} = 200$, taken at $t = 125$. The darkest contour lines represent $v = U_\infty$ and the lightest lines $v = -U_\infty$. The simulation used 7th-order polynomial approximations, with $\Delta t = 2.5 \times 10^{-4}$ with second-order global temporal scheme and IEXT2 with two iterations per timestep.

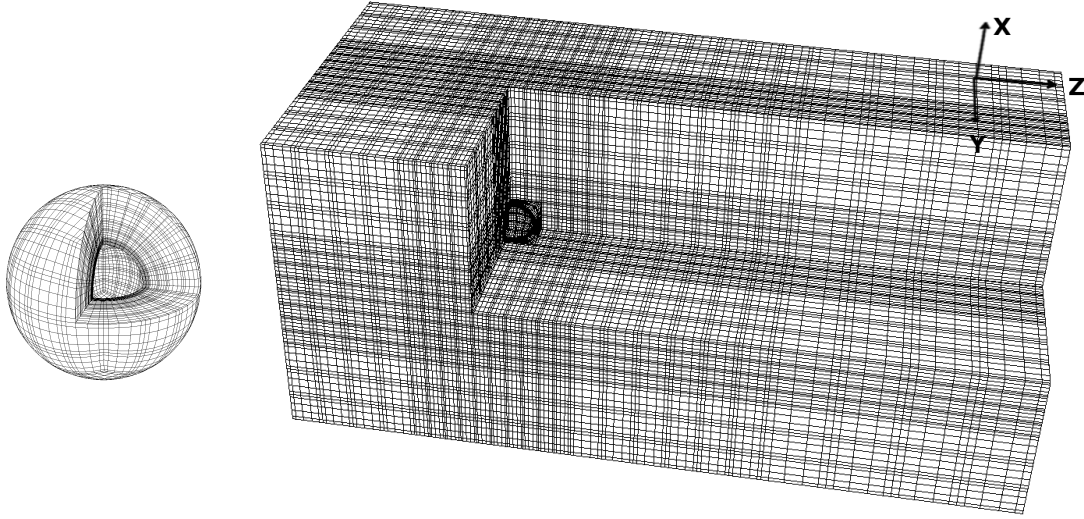
4.3 Rotating Sphere

In this test case, benchmarking of performance of the developed moving overlapping grid method in the presence of moving bodies is extended to a three-dimensional situation. Thus, a problem is considered where an incoming uniform stream interacts with a rotating sphere, as a suitable benchmark. The choice of this particular test problem is, again, spurred by a large amount of previous experimental and computational studies, facilitating a comparison of results.

Some of the earliest research regarding the flow past a rotating sphere was performed by Maccoll in 1928 [148], who experimentally measured drag and lift forces on spheres spinning with various angular velocities. Since that time, several projects, both experimental and computational, have been performed focusing on the subject (see [149, 150, 151, 152, 3, 153, 4, 5]), providing a large collection of data for comparison. The current validation study performs simulations of a transversely rotating sphere (the axis of rotation is perpendicular to the flow) seeing that a large majority of published research has been done regarding a sphere rotating in this manner, as opposed to streamwise rotation. The primary purpose of this test is to illustrate accurate simulation of three-dimensional fluid flow influenced by solid bodies in motion.

As illustrated in Figure 4.16, the global computational domain is decomposed into two subdomains. A spherical mesh is formed around the solid walled sphere, which is positioned to cover a vacancy within a large outer mesh. The inner mesh is created to ensure that at least nine grid points lie within the boundary layer for all the simulations for a sufficient resolution (boundary layer stays laminar in all the simulations for low Reynolds numbers considered here). Steady inflow velocity U_{∞}^* is prescribed at one end of the outer mesh and outflow boundary conditions are set

at the other, while the steady rotation of the sphere is imposed by establishing the inner mesh velocity with the ALE formulation.



(a) Sliced Inner

(b) Sliced Outer Mesh

Mesh

Figure 4.16: Mesh geometry of transversely rotating sphere simulation [Not scaled relative to each other]. The subdomains displayed have sections removed for visualization of the solid sphere and vacancy for the inner and outer meshes respectively.

Note that although the current set-up could have been accomplished through a single stationary mesh by assigning non-zero, rotating Dirichlet velocity conditions at a sphere boundary, as was done in most of the computational simulations of this problem cited above [3, 153, 4, 5], the goal here is to find a suitable and well documented test case for validation of our method. As such, the range of Reynolds numbers and angular velocities investigated corresponds with those of other computational studies of rotating spheres [3, 4, 5], which contain detailed data for comparison that is not found in much of the experimental studies.

Results are presented for several different inner mesh angular velocities, using Reynolds numbers ($\text{Re} = D^*U_\infty^*/\nu^*$) of 100, 250, and 300, which correspond with those reported in a computational study of Giacobello et al. [5] concerning transversely rotating spheres. These Reynolds numbers lie within three different regimes for uniform flow around a *stationary* sphere, of which a detailed investigation and description can be found in the work of Johnson and Patel [154]. The flow around a stationary sphere at $\text{Re} = 20 - 210$ is axisymmetric and steady, while a steady transition regime appears as Reynolds numbers increase beyond 210 and the flow becomes non-symmetric, until $\text{Re} \approx 270$ where the flow becomes unsteady and vortex shedding begins to occur. When the rotation is added to the sphere, the flow patterns change depending on the angular velocity, see discussion below. Thus, vortex shedding is observed for some of the rotating cases with $\text{Re} = 250$ as opposed to a stationary case.

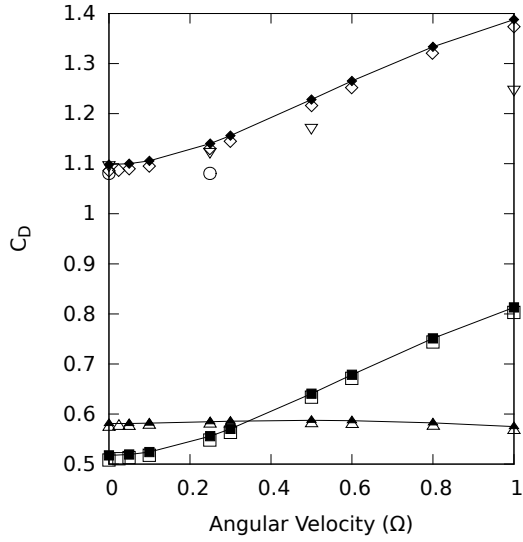
Re = 100 Ω	Regime	Re = 250 Ω	Regime	Re = 300 Ω	Regime
0.0	Steady	0.0	Steady	0.0	Fully Unsteady
0.05	Steady	0.05	Transitional	0.05	Fully Unsteady
0.1	Steady	0.078	Fully Unsteady	0.1	Fully Unsteady
0.25	Steady	0.1	Fully Unsteady	0.25	Fully Unsteady
0.3	Steady	0.2	Fully Unsteady	0.3	Fully Unsteady
0.5	Steady	0.3	Transitional	0.5	Steady
0.6	Steady	0.5	Steady	0.6	Steady
0.8	Steady	0.6	Steady	0.8	Transitional
1.0	Steady	0.8	Steady	1.0	Transitional
		1.0	Steady		

Table 4.3: A tabular listing of regimes for each rotating sphere simulation performed in the present study.

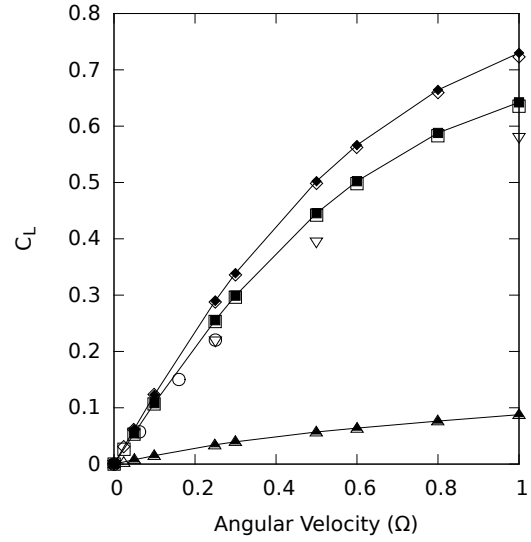
The drag and lift coefficients, determined by forces on the sphere: $C_D = F_D^*/(\frac{1}{8}\rho_f^*U_\infty^{*2}\pi D^{*2})$ and $C_L = F_L^*/(\frac{1}{8}\rho_f^*U_\infty^{*2}\pi D^{*2})$, where ρ_f^* is the fluid density, were used in comparison with results reported in [3, 4, 5] and can be seen in Figure 4.17 where non-dimensional angular velocity is given by $\Omega = \Omega^*D^*/(2U_\infty^*)$. Specific viscous and pressure contributions to the total force are reported for additional comparison. As can be seen from the figure, the forces calculated using our moving overlapping mesh method correlate well with the published results found in [3, 4, 5]. Note that the prescribed rotation of the sphere in the simulations of Kurose and Komori [3] (or the boundary conditions applied at the sphere surface) are different than the rotation prescribed in the present study and in [4, 5] (see [5] for details), which renders the results of Ref. [3] to be slightly different.

For two of the higher Reynolds numbers, $Re = 250$ and $Re = 300$, Giacobello et al. [5] observed three different regimes depending on the angular velocity: steady regime when the vortex shedding is not observed throughout a duration of the simulations; transitional regime when the vortex shedding initially occurs and then dies out; and fully unsteady regime, with the vortex shedding persisting until the end of the simulations (the flow was always in a steady regime for $Re = 100$ for the tested angular velocities). We noticed a similar behavior in our simulations, with corresponding regimes for our cases listed in Table 4.3.

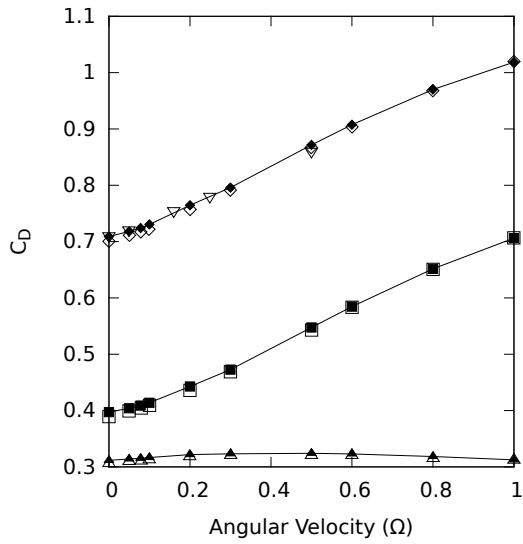
For the Reynolds numbers $Re = 250$ and $Re = 300$, the Strouhal numbers of the vortex shedding were calculated for the tested angular velocities and compared them with the data of [3, 4, 5] in Figure 4.18. Following Giacobello et al. [5], for the transitional cases (for which the vortex shedding initially occurs and then dies out), we present initial and not final vortex shedding frequency (which would be zero). Zero frequency in this plot thus corresponds to steady cases. As can be seen, a comparison of Strouhal numbers for different angular velocities and Reynolds



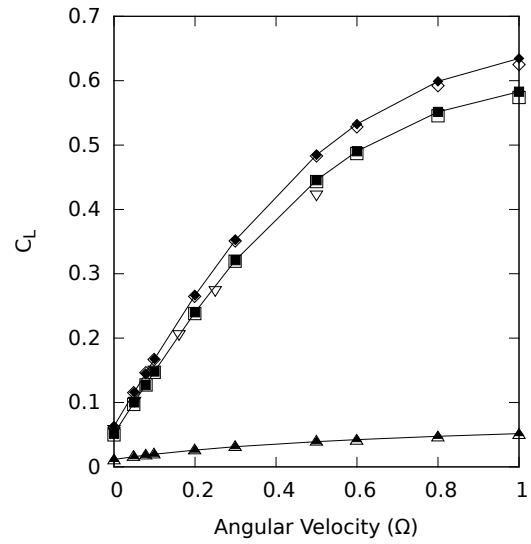
(a) Drag Coefficient, Re=100



(b) Lift Coefficient, Re=100



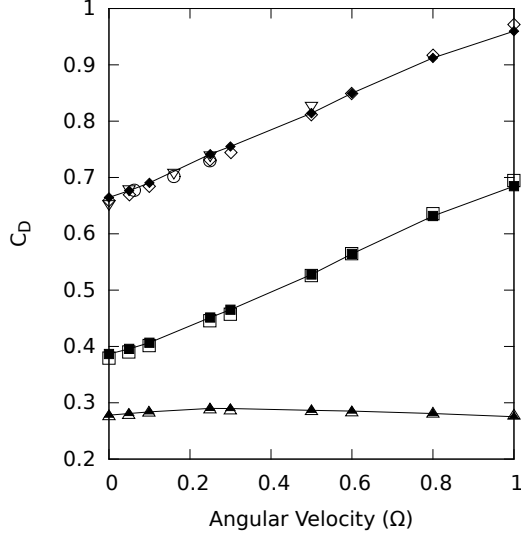
(c) Drag Coefficient, Re=250



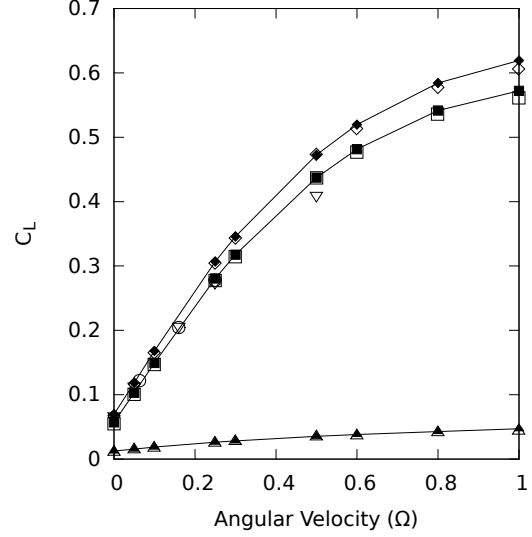
(d) Lift Coefficient, Re=250

: Figure continued on next page

numbers (see Figure 4.18) also shows excellent correlation with the published data, again, apart from a slight divergence from the results in [3], calculated with different rotating conditions.



(e) Drag Coefficient, $Re=300$



(f) Lift Coefficient, $Re=300$

Figure 4.17: Comparison of force coefficients at $Re=100, 250$, and 300 . Closed symbols represent present data, and open symbols represent data reported in previous publications [3, 4, 5]. Closed upright triangles represent the viscous contribution, closed squares represent the pressure contribution, and closed diamonds denote the total force coefficient in the present data. Open diamonds, open inverted triangles, and open circles represent data given for the total force coefficient in [5], [4], and [3] respectively. Open squares denote the pressure contribution, and open upright triangles the viscous contribution, as reported in [5]. Each present simulation was performed using 5^{th} -order polynomial approximations with $\Delta t = 5 \times 10^{-3}$, second-order global temporal scheme and IEXT2 with two iterations per timestep.

Visual comparison of trailing vortices between present results and those of Ref. [5] is performed in Figure 4.19 for $Re = 250$ and 300 for unsteady and transitional regimes (initial vortices). Vortices are visualized using the λ_2 -criteria described in [155] at an iso-value of $\lambda_2 = -8 \times 10^{-4}$. As can be seen, the observed vortical structures reveal excellent resemblance with the figures reported in [5].

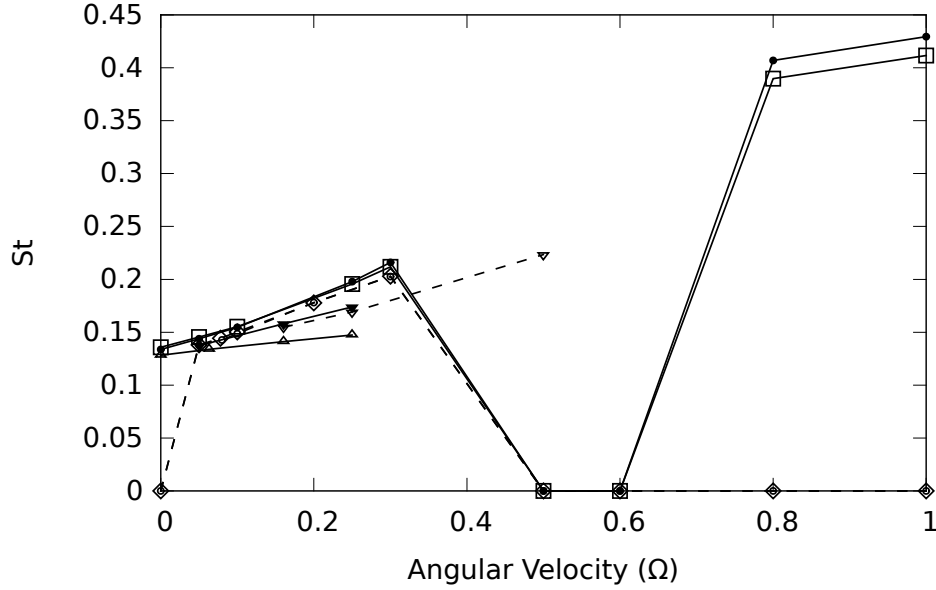
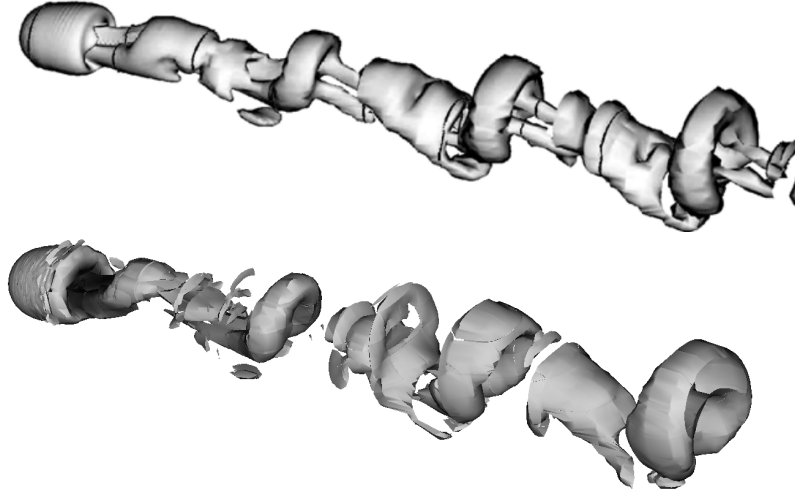
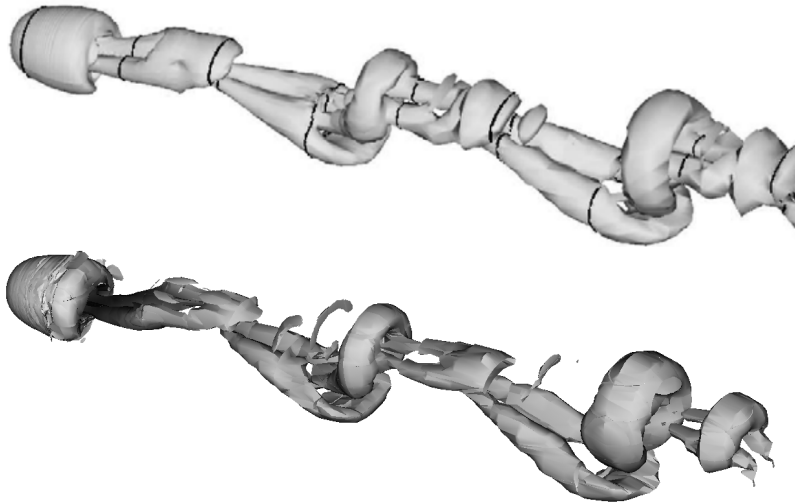


Figure 4.18: Comparison of Strouhal numbers given by the present simulations and data presented in [3, 4, 5]. All data at $Re = 250$ is given with dashed lines, while all data at $Re = 300$ is given with solid lines. Symbols represent data as follows: hollow diamonds - present data with $Re=250$, hollow squares - present data with $Re = 300$, hollow upright triangles - Kurose and Komori [3] with $Re = 300$, hollow inverted triangles - Niazmand and Renksizbulut [4] with $Re = 250$, filled inverted triangles - Niazmand and Renksizbulut [4] with $Re = 300$, hollow circles - Giacobello et al. [5] with $Re = 250$, filled circles - Giacobello et al. [5] with $Re = 300$. All present simulations were performed with 5^{th} -order polynomial approximations and $\Delta t = 5 \times 10^{-3}$, using second order temporal scheme and IEXT2 with two iterations per timestep.

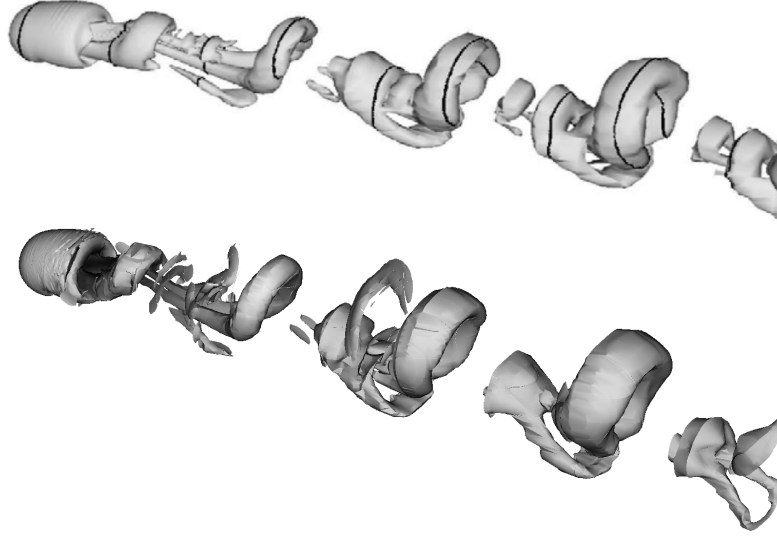


(a) $\text{Re}=250$, $\Omega = 0.2$, $t = 492.5$. Unsteady Regime



(b) $\text{Re}=300$, $\Omega = 0.05$, $t = 350$. Transitional Regime

: Figure continued on next page



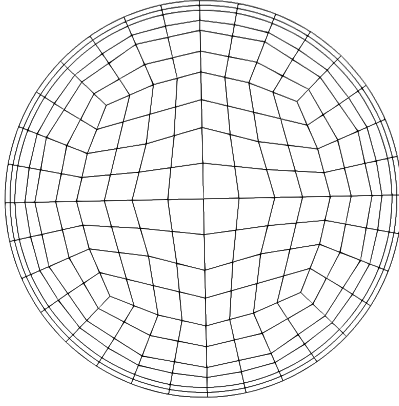
(c) $\text{Re}=300$, $\Omega = 0.3$, $t = 260$. Unsteady Regime

Figure 4.19: Visual comparison of shed vortices as reported in [5] (top visualization in each subfigure) with present results (bottom visualization in each subfigure). Vortex iso-surfaces are presented at $\lambda_2 = -8 \times 10^{-4}$ using the identification method in [155]. The present simulations were performed using 5^{th} -order polynomial approximations and $\Delta t = 5 \times 10^{-3}$, with second order time-stepping scheme and IEXT2 with two iterations per timestep. Visualizations of the wake are displayed up to 16 diameters downstream.

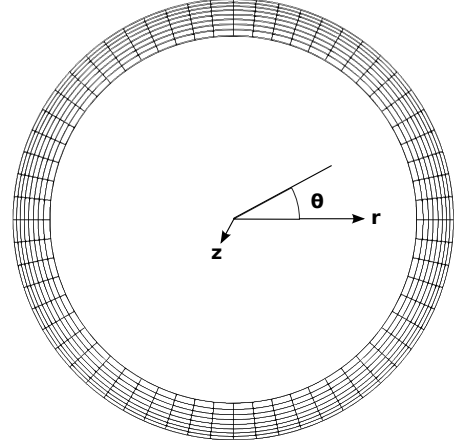
COMPUTATIONAL TIMING AND SCALING

5.1 Stationary Overlapping Meshes

In this section, scalability tests are performed for the developed stationary and moving overlapping grid methods. The previous test case of the turbulent pipe flow simulations (discussed in Section 3.3) is utilized as a platform for the scalability studies. However, the refined computational meshes (see Figure 5.1) were constructed, with higher element counts than in the DNS simulations described above (see label of Figure 5.2) in order to enable testing on high processors counts (up to 1024). The global problem contains a total of 51.2k elements (12.8k in interior mesh, 38.4k in exterior mesh)), with the total number of grid points varied based upon the polynomial order used for solution approximations. The simulations were performed with Reynolds number of 5300 based on the pipe diameter. The refined outside mesh consists of three times the number of elements and collocation points as the inside mesh. This discrepancy in the number of elements allows for testing of two methods for processor allocations: equal allocation (same number of processors for each domain, irrespective of the relative mesh sizes), and proportional allocation (processors allocated proportionally to the number of elements the mesh contains, namely, the outside domain was assigned three times as many processors as the inside domain). The results of the scalability tests, performed on the SDSC Gordon and TACC Stampede clusters, are shown in Figures 5.2 and 5.3 with a refined single mesh pipe simulation for comparison. Figure 5.2 shows the total CPU time for each simulation, divided by the number of time steps and the total number of elements contained in the simula-



(a) Interior Mesh



(b) Exterior Mesh

Figure 5.1: Cross-section of grids for two mesh pipe flow simulations. The displayed meshes extend 6 pipe diameters in the z -direction.

tion, versus the total number of allocated CPUs, and Figure 5.3 contains two plots, the second of which shows a comparison of timing data with respect to the number of gridpoints per core. Figure 5.4 breaks down the time spent per timestep in different aspects of the stationary overlapping mesh coupling procedures. We see that the scaling of the stationary overlapping mesh methodology nearly reaches ideal scaling in each presented case. Line sections that appear to achieve better than ideal scaling are a result of inconsistent communication speeds among the nodes assigned to the jobs, for example, slower communication speeds among nodes assigned to a 128 core test compared with a 256 core test with faster communication speeds among the assigned nodes will appear to have better than ideal scaling. Several simulations were performed for each case, and averages of the timing data is presented. As expected, scaling for proportional CPU allocations yielded a speedup over equal allocations in each of the tested cases, due to more consistent load balancing. However, while the number of elements assigned per core is a good indicator of the work load, there is more to consider when seeking to attain optimal speeds. For instance, the external mesh contains 4800 element faces requiring interpolation from values in the other sub-

domain, while the interior mesh contains only 1600 such faces. In the proportional scaling tests, the outside mesh is assigned three times the number of cores that are assigned to the inside mesh, though during the interpolation step the inside mesh must determine values for three times as many interface points. The fewer processors assigned to the inside mesh have a much larger work load during the overlapping mesh procedures, as is illustrated in the plots of Figure 5.4 by the greater amount of time the inside mesh spends for overlapping mesh procedures. While the proportional allocation is a good starting point for speed optimization of a problem, each simulation must be uniquely tested for additional speed-up.

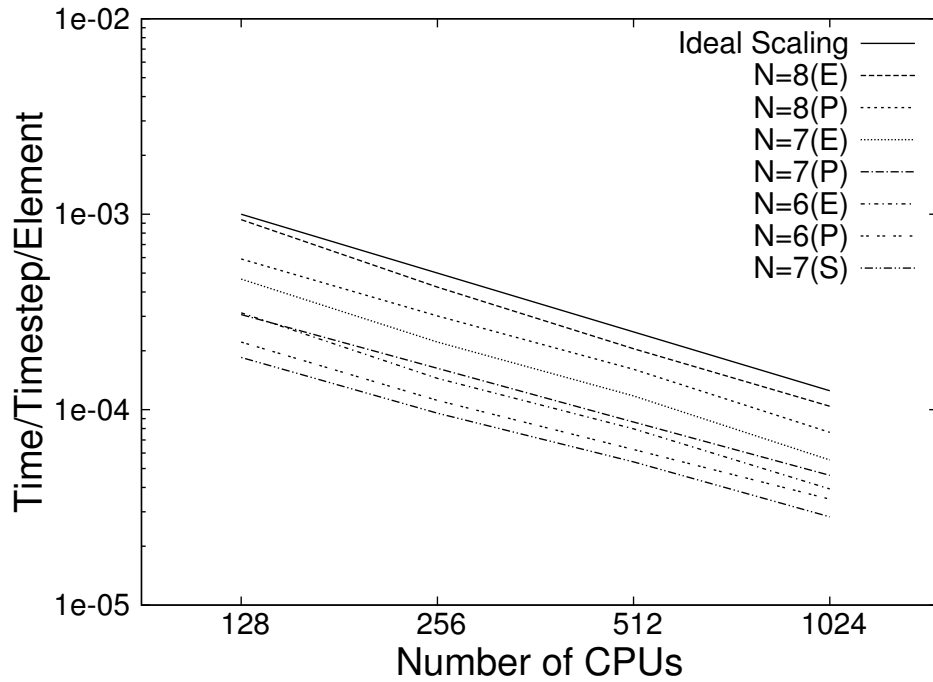
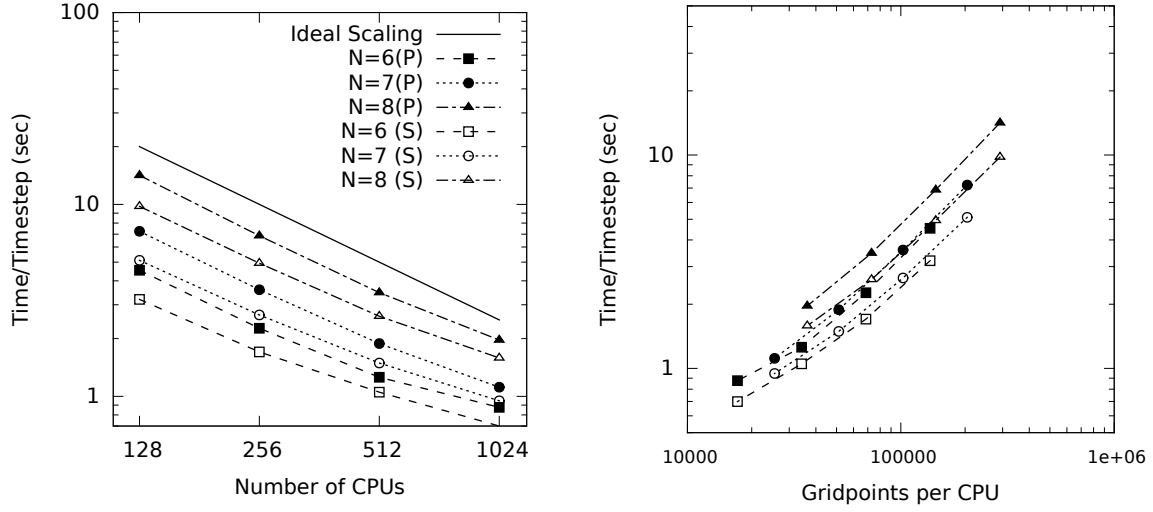


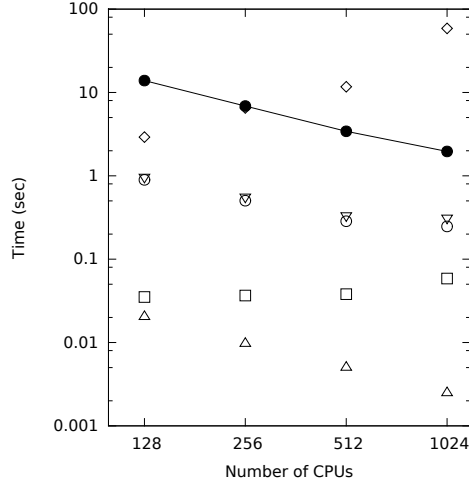
Figure 5.2: Computational time vs number of CPUs for simulations performed on the SDSC Gordon cluster with different polynomial orders of approximation (N). Grids used for double mesh simulations contain a total of 51.2k elements (12.8k in inside mesh, 38.4k in outside mesh), giving $\sim 17.6\text{M}$, 26.2M , and 37.3M total gridpoints for 6th, 7th, and 8th order polynomial approximations respectively. The single mesh simulation consists of $\sim 41\text{k}$ elements giving $\sim 21.0\text{M}$ gridpoints for 7th order polynomial approximations. The 'E' denotes that the simulations were performed with an *equal* number of processors allocated to each mesh, while 'P' designates simulations where processors were assigned to each mesh *proportional* to the number of elements it contains. The single mesh simulation is signified by an 'S'.



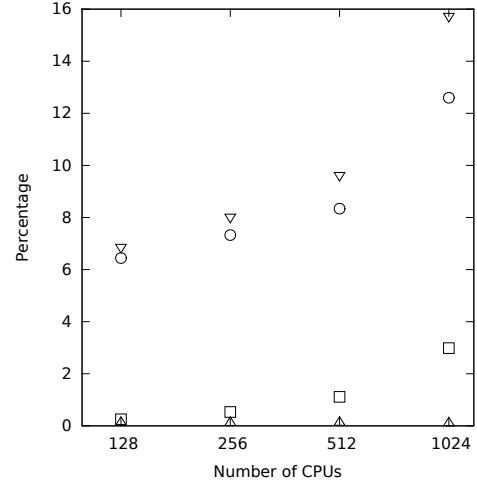
(a) Time vs Number of CPUs

(b) Time vs Gridpoints per CPU

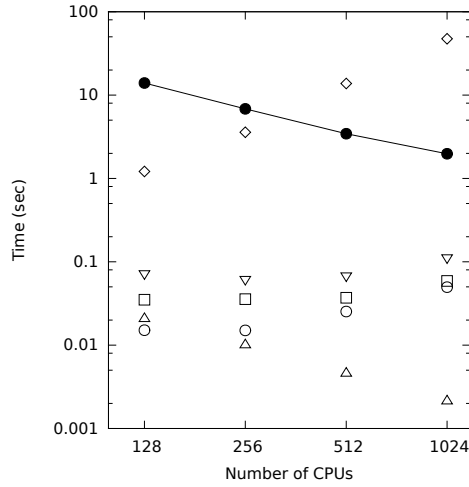
Figure 5.3: Computational scaling data for simulations performed on the TACC Stampede cluster with different polynomial orders of approximation (N), using second order temporal schemes with two iterations. Grids used for double mesh simulations contain a total of 51.2k elements (12.8k in inside mesh, 38.4k in outside mesh), with single mesh simulations (S) also containing a total of 51.2k elements, giving ~ 17.6 M, 26.2M, and 37.3M total gridpoints for 6th, 7th, and 8th order polynomial approximations respectively. Double mesh simulations used proportional (P) CPU allocation.



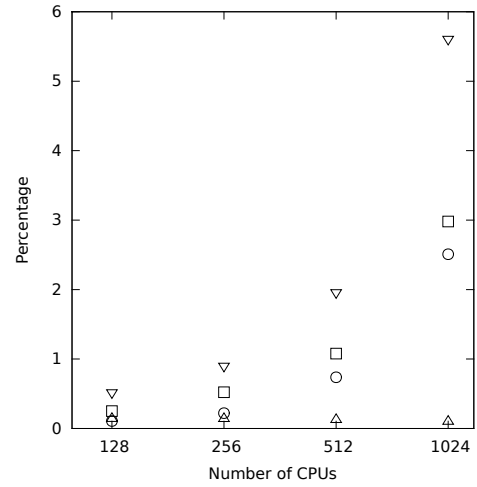
(a) Timing Data, Inside Mesh



(b) Percentage of Time in Overlapping Mesh Procedures, Inside Mesh



(c) Timing Data, Outside Mesh



(d) Percentage of Time in Overlapping Mesh Procedures, Outside Mesh

Figure 5.4: Stationary overlapping mesh timing data for simulations performed on the TACC Stampede cluster. All tests were performed with 8th order polynomials, using second order temporal schemes with two iterations using a proportional allocation of CPUs. The solid lines with filled circles denote the total computational time per timestep. Open diamonds represent the time spent in overlapping mesh initialization (performed only once, before the first timestep), open squares represent the time/timestep used for interpolation in the first iteration, open circles represent the time/timestep used for interpolation in the second iteration, open upright triangles represent the time/timestep for temporal extrapolation, and open inverted triangles represent the total overlapping mesh communication time spent per timestep. Plots (a) and (c) present the timing data, while plots (b) and (d) present the overlapping mesh communication time as a percentage of the total time per timestep.

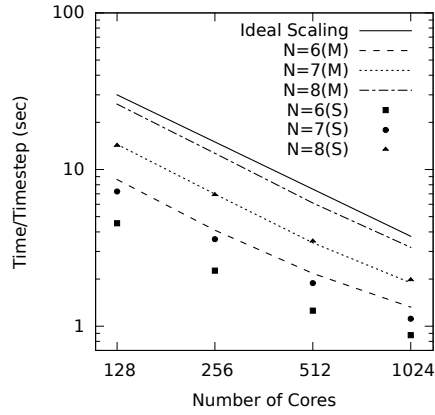
5.2 Moving Overlapping Meshes

Scaling and timing analyses were performed with identical pipe geometry (Figure 5.1) and simulation parameters used in the stationary overlapping mesh tests so that a one-to-one comparison could be made with corresponding data. Tests were performed on Stampede supercomputing resources in conjunction with a computational resources allocation through XSEDE. In the present moving mesh tests, the interior mesh is constrained to rotate with a rotational velocity of $\Omega^* \equiv \Omega D/U_b = 1$, where U_b is the bulk flow rate of fluid in the pipe, which, as in the test case of convecting eddies, represents a “virtual movement” and does not affect the flow. For all tests in this section, each mesh is assigned a number of processors proportional to the number of elements it contained as described in the previous section.

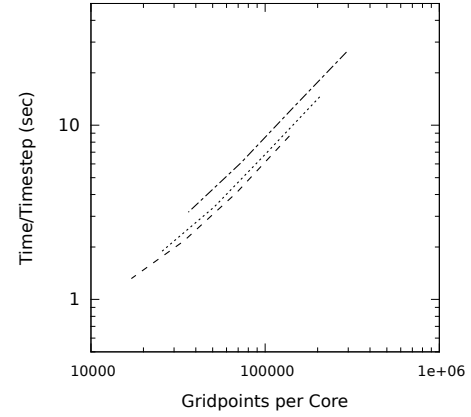
The scaling analyses (Figure 5.5) show near linear strong scaling for both stationary and moving overlapping simulations. The collapse of data is much better when scaled with gridpoints per core using different polynomial orders, indicating robust parallel performance.

Figure 5.6(a) shows a breakdown of the percentage of wall time per timestep spent in components of the moving overlapping mesh code that deal with inter-mesh communications. Notice that the percentage of wall time spent in the interpolation and extrapolation steps is very small while a much more sizable percentage of wall time per timestep is expended in the search step. When comparing the wall time spent in specific components of the code (Figure 5.6(b)), we see that the time spent in routines handling inter-mesh communications remains fairly constant regardless of the number of cores used.

Figure 5.7 illustrates the percent change in wall time per timestep going from the traditional single mesh solver to the moving overlapping mesh solver (of the same



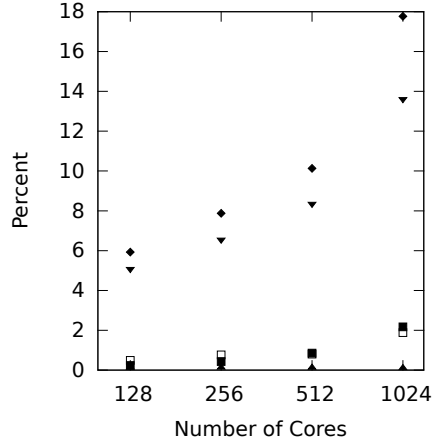
(a) Overlapping Moving Mesh (Lines)
Simulation Scaling Compared with
Overlapping Stationary Mesh (Points)
Scaling



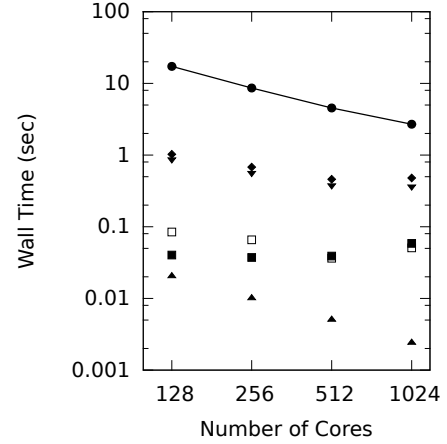
(b) Scaling of Moving Mesh Simulations
with Respect to the Number of Grid-
points Assigned to Each Core. Line
Patterns Correspond to Those Shown in
Figure 5.5(a)

Figure 5.5: Scaling of overlapping mesh simulations using 6th, 7th, and 8th order polynomial approximations. Meshes are identical for moving and stationary mesh cases. The timestepping scheme BFD2 with IEXT2 and two iterations per timestep was used for each simulation.

total element count), and from the stationary overlapping mesh solver to the moving overlapping mesh solver. While still maintaining scalability of the global code, moving overlapping mesh simulations understandably spend more wall time per timestep than the other two methods due to greater amounts of work and additional communications performed during each timestep.



(a) Percentage of Wall Time Per Timestep Used in Different Components Handling Inter-Mesh Communication.



(b) Wall Time Used Per Timestep in Different Components Handling Inter-Mesh Communication.

Figure 5.6: Time spent in different components of the moving overlapping mesh methodology. Filled circles represent total time per timestep, filled diamonds - time spent in all components handling inter-mesh communication, filled inverted triangles - the search step, filled upright triangles - extrapolation, hollow squares - 1st interpolation step, filled squares - 2nd interpolation step. Simulations were performed with 8th order polynomials using BDF2 and IEXT2 with $\Delta t = 1 \times 10^{-3}$.

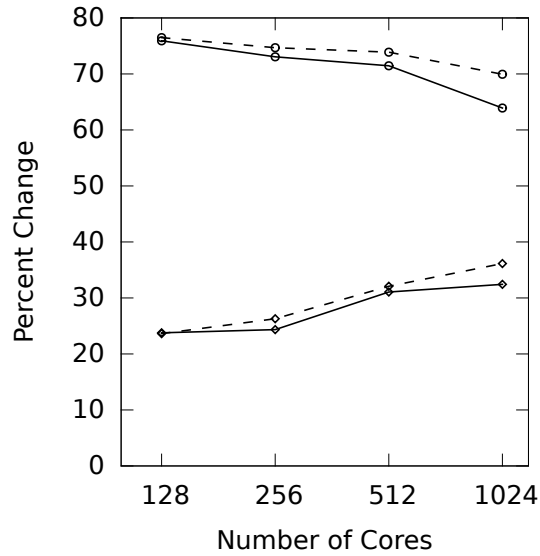


Figure 5.7: Percent change in computation time, comparing the present moving overlapping mesh methodology with the traditional single mesh implementation (hollow circles) and the stationary overlapping mesh methodology (hollow diamonds). Solid lines are used for interior mesh comparisons, and dashed lines for the exterior mesh. For computational parameters see Figure 5.6.

Chapter 6

DYNAMIC STALL

6.1 Problem Introduction

A greater understanding of the aerodynamic forces acting on airfoils with a wide variety of flow conditions is imperative in creating flexible models to improve the fidelity of large scale aerodynamic simulations. While many aspects of airfoil aerodynamics have been examined in depth, several facets have traditionally been difficult to measure or simulate. The flow around pitching airfoils is one field of research that has been studied for several decades, although much is unknown regarding the nature of the flow when unsteady flow conditions are present. The flow around pitching airfoils, especially as it relates to dynamic stall, has ramifications for the efficiency and design of helicopter rotors [156, 157], wind turbines [158], other rotating machinery such as compressors[159], as well as extensions to the maneuverability of fixed wing aircraft [160, 161].

The general characteristics of pitching airfoils and mechanisms of dynamic stall are well understood in the presence of steady inflow conditions, and several experimental and computational projects have examined this topic. A dynamically pitching-upward airfoil with steady inflow will generally maintain lift up to an angle beyond its static stall angle where dynamic stall occurs. As an airfoil pitches upward, a large vortex, referred to as a dynamic stall vortex (DSV), forms at the leading edge of an upward pitching airfoil, creating a low pressure region, which then travels along the suction side of the airfoil toward the trailing edge. The lower pressure on the suction side of the airfoil increases the lift, until the vortex nears the trailing edge where it sepa-

rates and a dramatic decrease in lift, dynamic stall, ensues. Dynamic stall research typically focuses on the effects of altering pitching frequency, pitching amplitude, Reynolds number, and Mach number, as well as possible ways to gain greater control of dynamic stall [162, 163, 164, 165, 166, 167]. However, unsteady inflow conditions can also play a crucial role in the aerodynamics of pitching and plunging airfoils. Blades on wind turbines operating in yawed or unsteady flow conditions, for instance, experience inflow velocity that is periodic with respect to azimuthal angle [168, 169]. Turbulence, gusts, and vortices caused by upstream structures, objects, or atmospheric conditions create unsteady inflow for helicopter rotors, wind turbine blades, and wings on maneuvering aircraft. In order to create a more complete realization of dynamic stall phenomena, the various velocity fluctuations commonly experienced by pitching airfoils in realistic situations need to be included in future research and simulations.

At the present time, some experimental research has been performed to examine the effects of unsteady inflow on pitching airfoils, while a much smaller amount of computational work has been done in this area. In the 1970's Pierce et al. [157] and Kottapalli et al. [170] performed experiments on pitching helicopter blades using a gust generator to produce periodic inflow conditions. Their results show that the varying freestream velocity affects the pitching moment and drag experienced by the blade, and that the magnitudes of the forces and moment change depending on the phase difference between the airfoil oscillation and inflow velocity oscillation. Similar experimental investigations have been performed more recently by Shi and Ming on a pitching delta wing, for improvements in the design of super-maneuverable aircraft [161, 160].

Experimental investigations of turbulence effects on pitching airfoils have been performed by Conger et al. [171], Laneville et al. [172], and Chen et al. [173]. In the

study by Chen et al., a turbulent wake was generated upstream of a ramping airfoil by placing a small cylinder in the flow field. The experiment had a chord based Reynolds number of 80,000. The effects of varied cylinder position with respect to the airfoil were studied to compare the aerodynamic forces and moments with those experienced by a pitching airfoil in steady freestream. Their work showed that dynamic stall events occur at larger angles of attack in the presence of a turbulent wake when compared to steady inflow cases, although different vertical positions of the cylinder had differing effects on the aerodynamic forces.

Little computational research has been performed regarding pitching airfoils with unsteady freestream flow due to the difficulty of traditional computational fluid dynamics (CFD) solvers in handling moving geometries with non-uniform inflow conditions, including the computational cost of such simulations. Recent research published by Gharali et al. [6, 168] investigates two-dimensional pitching airfoils in the presence of periodic inflow velocity using finite volume methods. However, no known computational simulations of pitching airfoils have been performed with turbulent inflow conditions, yet information from such simulations would be pivotal in improving the fidelity of large scale models of systems with realistic flow conditions. These enhanced models lead to improvements in the physical design of aircraft and wind turbines.

The moving overlapping mesh methodology allows us to decompose the global computational domain into a moving mesh and a stationary mesh. By enabling one part of the mesh to move, consistent resolution near the solid boundary of a moving object is easily maintained. A stationary background mesh allows for unsteady boundary conditions to be straightforwardly implemented in the lab reference frame, and for stationary objects to be placed elsewhere in the flow field.

This chapter simulates an extruded NACA0012 airfoil, with prescribed oscillatory pitching motion, and the effects of the airfoil's interactions with a turbulent wake

generated by a small stationary cylinder that disturbs the upstream flow, similar to experiments performed by Chen and Choa [173], are investigated. Aerodynamic forces and the pressure values at the airfoil surface are compared among simulations without the upstream disturbance, and simulations with the disturbance originating at different positions within the upstream flow field. Initial pitching airfoil tests are performed in two- and three-dimensions and results are compared with published findings. Three-dimensional simulations are then performed with an upstream cylinder generating a turbulent wake that is incident upon an oscillating airfoil. The effects of the upstream disturbances are investigated with respect to the oscillating airfoil in a steady flow field, including a comparison of Power Spectral Density functions derived from the aerodynamic coefficients among all cases.

6.2 Validation and Comparison of 2D and 3D Simulations

This section will present DNS of two and three-dimensional pitching airfoil problems that have been performed both for validation and for comparison with each other and with published data. In all present cases the global computational domain is decomposed into two overlapping meshes. While unique mesh geometries are constructed for each simulation, the interior mesh, in each case, is constructed around a NACA0012 airfoil, following the available data for comparison, and is constrained to move with pitching motion. The background mesh is stationary and contains a vacancy which is covered by the interior mesh. Three-dimensional meshes are generated by extruding two-dimensional meshes, and all 3D domains have a spanwise width based on the airfoil chord length, $s/c = 0.2$. A two-dimensional cross-section of the meshes used in sections 6.2.1 and 6.2.2 is illustrated in Figure 6.1

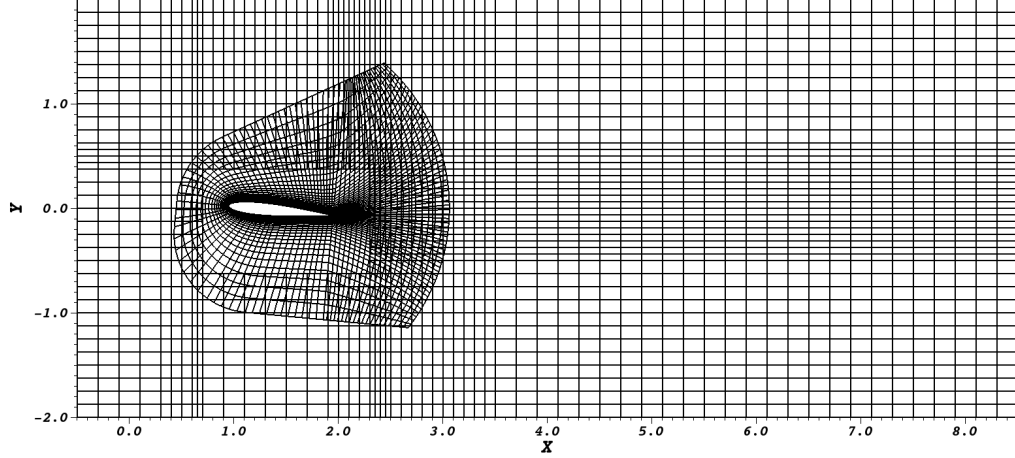


Figure 6.1: A cross section of the global computational domain for steady inflow simulations described in Sections 6.2.1 and 6.2.2, showing element boundaries of the moving airfoil mesh within the stationary background mesh.

In all future problems the lift, drag, and pitching moment coefficients are given by

$$C_L = \frac{L}{\frac{1}{2}\rho U_\infty^2 S} \quad (6.1)$$

$$C_D = \frac{D}{\frac{1}{2}\rho U_\infty^2 S} \quad (6.2)$$

$$C_M = \frac{M}{\frac{1}{2}\rho U_\infty^2 S c}, \quad (6.3)$$

where L is the lift force, D is the drag force, M is the pitching moment (or torque), ρ is density, U_∞ is the inflow velocity, and S is the planform area, and c is the chord length.

6.2.1 Pitching-up Airfoil, $Re=80,000$

The first set of tests simulate the experimental problem described in [173] by Chen and Choa with undisturbed inflow. A NACA0012 airfoil is pitched with constant angular velocity from 0° to 30° with a reduced pitchrate of $k \equiv \Omega c / 2U_\infty = 0.01$, where Ω is the angular velocity and U_∞ is the inflow velocity, and chord based Reynolds

number, $Re = 80,000$. 7^{th} -order polynomial approximations are used for spatial discretization within each element. Variable timestepping is used with the maximum timestep $\Delta t_{max} = 2 \times 10^{-4}$, where the nondimensional time $t = t^* U_\infty^* / c^*$, though the average timestep in the present simulation is $\Delta t_{mean} \approx 4 \times 10^{-5}$. The timestep size is altered when the Courant number (discussed in Section 4.1.4) departs from prescribed upper and lower bound limits, which are chosen for stability and accuracy, but also for efficient use of computational resources. Second order time integration (BDF2) and IEXT2 at interfaces are used with two iterations per timestep. Grid spacings are presented in Table 6.1. Steady inflow is prescribed on the left boundary with

$\Delta s_U / c$	$\Delta s_L / c$	$\Delta n / c$	$\Delta z_{min} / c$	$\Delta z_{max} / c$
5.7×10^{-3}	8.1×10^{-3}	3.4×10^{-5}	1.3×10^{-3}	4.2×10^{-3}

Table 6.1: Airfoil grid parameters for simulations in Sections 6.2.1 and 6.2.2: maximum GL point spacing on the upper ($\Delta s_U / c$) and lower ($\Delta s_L / c$) surfaces of the airfoil, normal spacing on the upper airfoil surface at the mid-chord location ($\Delta n / c$), and the minimum ($\Delta z_{min} / c$) and maximum ($\Delta z_{max} / c$) spanwise GL point spacing. (The spanwise spacing is irrelevant in 2D simulations)

outflow conditions (3.5) at the right boundary. The top and bottom of the domain have symmetry boundary conditions, and in 3-D simulations spanwise boundaries are periodic. The 3-D mesh containing the airfoil is composed of $\sim 32k$ elements ($\sim 17M$ gridpoints with 7^{th} -order polynomials) and the stationary background mesh contains $\sim 23k$ elements ($\sim 12M$ gridpoints). Simulation time is normalized with the pitch-up period, T (t/T), thus at $t = 1$ the angle of attack of the airfoil is $\alpha = 30^\circ$, and for $t/T > 1$ the airfoil held at $\alpha = 30^\circ$.

A comparison of the lift coefficients resulting from 2D and 3D simulations with the experimental data is shown in Figure 6.2. We see that the 2D and 3D computational results match each other quite well for the duration of the simulation, and while lift coefficients match the experimental data during the first portion of the pitch-up cycle,

the simulation results later diverge. Due to the computational cost of the 3D test, only one cycle was performed, while Chen and Choa [173] averaged the results over four separate pitch-up cycles, leading to smoother results. Additionally, Chen and Choa [173] report a maximum of 25% tunnel blockage in their experiments, for which they state no corrections were made, which is likely another cause of the discrepancy between present simulation data and the experiments.

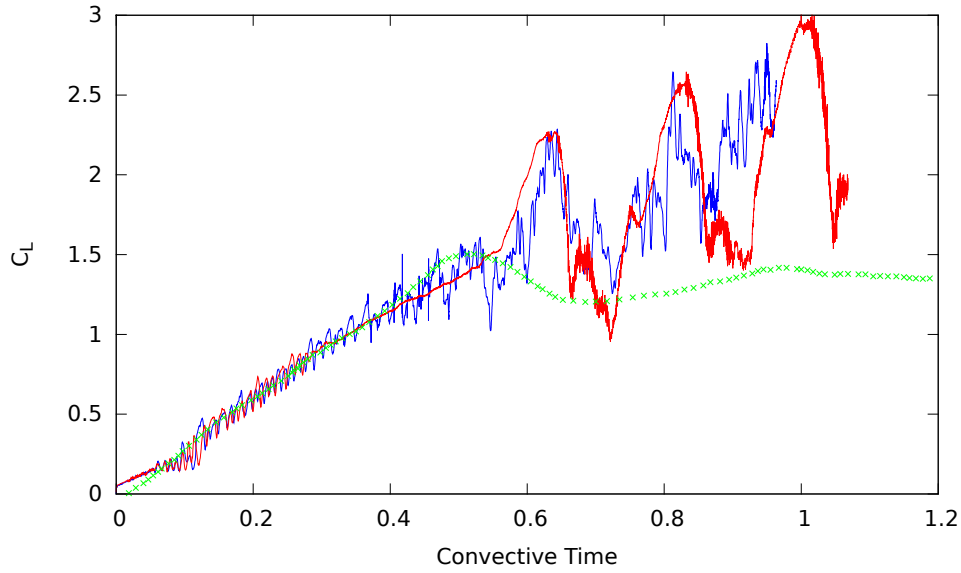


Figure 6.2: Lift coefficient with respect to time of NACA0012 airfoil pitching at reduced pitch rate $k=0.01$. The red curve represent present 3D data, the blue curve represents present 2D data, and green 'x' represent experimental data of Chen et al. [173]. The simulation time is normalized with respect to the pitch-up period (t/T). Thus the airfoil reaches angle of attack of 30° at $t = 26.2$, or $t/T = 1$, and is held stationary for the remainder of the simulation.

Figure 6.3 illustrates the vorticity of the flow just after the Dynamic Stall Vortex detaches from the suction surface of the airfoil. A comparison with the lift coefficient at the same normalized time shows a dramatic decrease in lift.

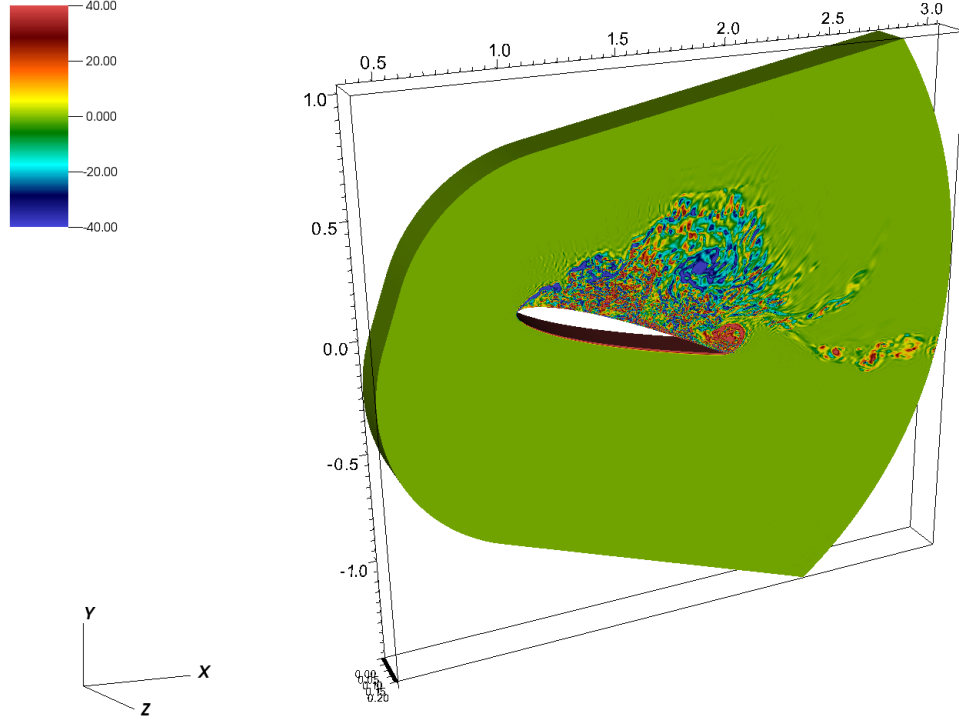


Figure 6.3: 3D spanwise vorticity plot at $t = 18.45$ or $t/T \approx 0.7$. The plot shows only the interior mesh which includes the airfoil.

6.2.2 2D Oscillating Airfoil at $Re = 135,000$

In this test case, a two-dimensional pitching airfoil simulation is performed and compared with the experimental lift results of Lee and Gerontakos [174]. Note that the meshes used for this test case do not achieve sufficient resolution for a fully resolved DNS of the flow with $Re = 135,000$, yet a good comparison of the aerodynamic forces may be expected (See, for example, [175] for a discussion of grid resolution effects in pitching airfoil simulations).

A NACA0012 airfoil is pitched about its quarter-chord axis where the oscillatory angle of attack is prescribed as

$$\alpha = \alpha_{\text{mean}} + \alpha_{\text{amp}} \sin(2\pi ft), \quad (6.4)$$

where, in this case, the mean angle of attack $\alpha_{\text{mean}} = 10^\circ$, the amplitude of oscillation $\alpha_{\text{amp}} = 15^\circ$, and the reduced frequency $k = 0.1$ where

$$k \equiv \pi f c / U_\infty. \quad (6.5)$$

Grid resolution near the airfoil is presented in Table 6.1. 7^{th} -order polynomials are used for spatial discretization and dynamic timestepping (maximum timestep $\Delta t = 1 \times 10^{-4}$) with BDF2/IEXT2 using two iterations is utilized.

Figure 6.4 gives the lift coefficient values resulting from the three full cycles of the present 2D computation compared with the experimental results in [174]. While general values correspond well, we see that the simulation data contains moderate fluctuations during upstroke, and large fluctuations during downstroke, when compared with the experimental data. This is due mainly to the large ensemble average performed on the data in [174], which was averaged over 100 pitching cycles. The larger fluctuations encountered in the lift force during the downstroke phase, in a single cycle, is consistent with published data from other 2D oscillating airfoil simulations [176, 6] (see also Figure 6.5 which was originally published in [6]). The lift coefficients ensemble averaged over the three cycles presented in Figure 6.4, are presented in Figure 6.6 and we see that since the maximum lift peaks in the 2D data do not occur at exactly the same angles of attack for each cycle, averaging such few cycles has the effect of rounding out the maximum lift peaks.

The published two-dimensional computational results, displayed in Figure 6.5, show that the sharp oscillations in the coefficient values are typical of two dimensional data.

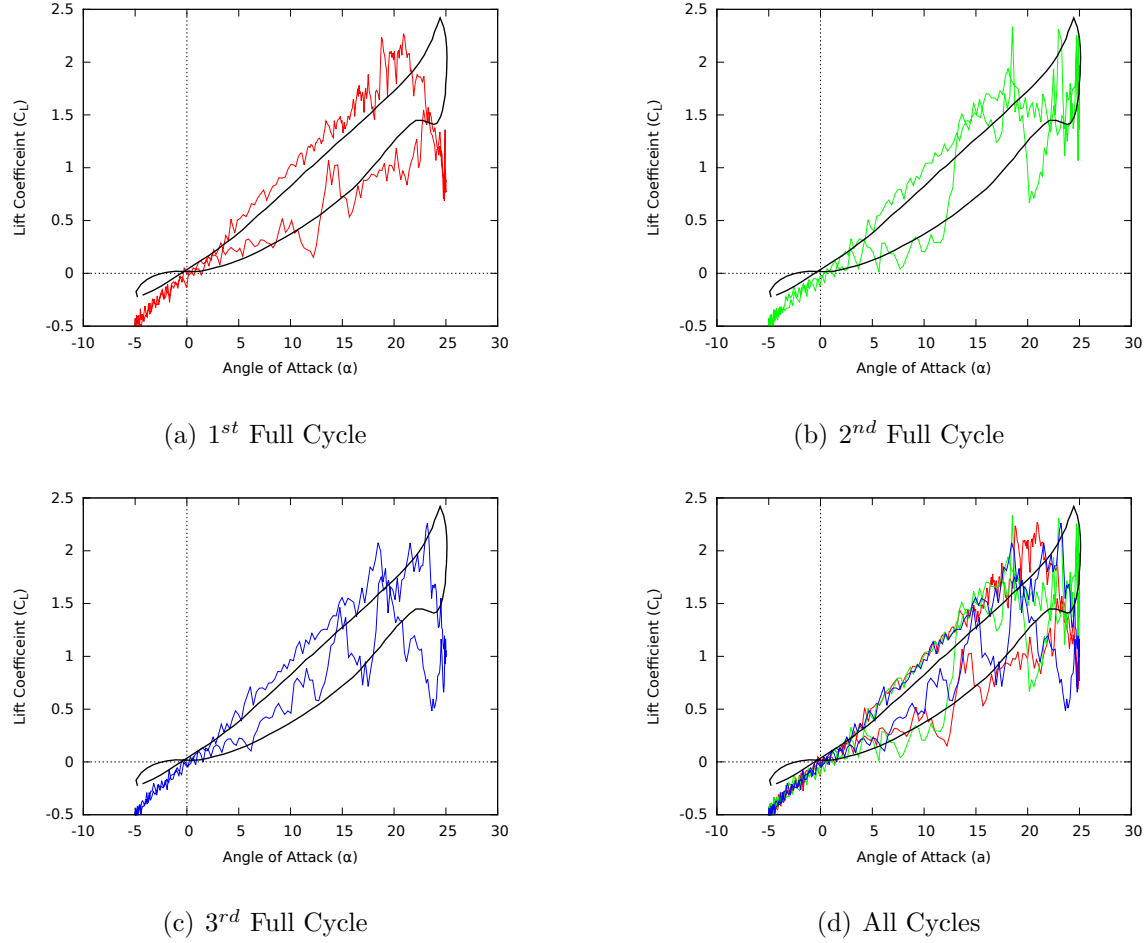


Figure 6.4: Lift coefficients from present 2D tests compared with the experimental data of Lee and Gerontakos (Black curve) [174].

6.2.3 2D and 3D Oscillating Airfoil at $Re = 44,000$

This section compares two and three dimensional data using probelm setup parameters that will also be used to determine the influence of upstream disturbances in future sections. While some 3D simulation results are presented here, a more comprehensive presentation of the data is given in the next section for an in-depth comparison with disturbed freestream flow cases.

A new set of meshes were constructed (Figure 6.7) with finer resolution upstream of the airfoil to ensure that the statistics of the turbulent wake are properly captured.

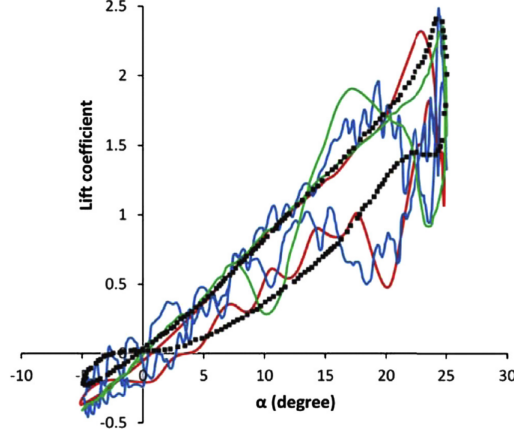


Figure 6.5: Lift coefficient values, as published in [6]. Black squares represent the experimental data in [174], and each of the colored lines represent data from 2D Unsteady Reynolds Averaged Navier-Stokes (URANS) simulations found in: red curve [6], blue curve [176], green curve [177].

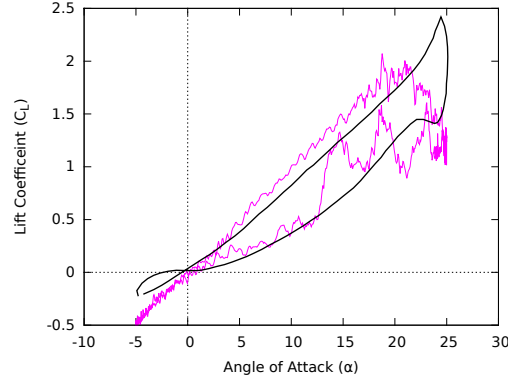


Figure 6.6: Mean lift coefficient for 2D tests phase averaged over the three cycles presented in Figure 6.4, compared with the experimental data of Lee and Gerontakos (Black curve) [174].

The meshes were expanded in the normal and streamwise directions to ensure that boundary condition effects do not negatively alter the flow approximation. Spatial discretization is performed with 6th-order polynomials, and dynamic timestepping is used with $\Delta t_{max} = 2 \times 10^{-4}$ and BDF2/IEXT2 with two iterations per timestep. Grid spacings at the airfoil surface are given in Table 6.2.

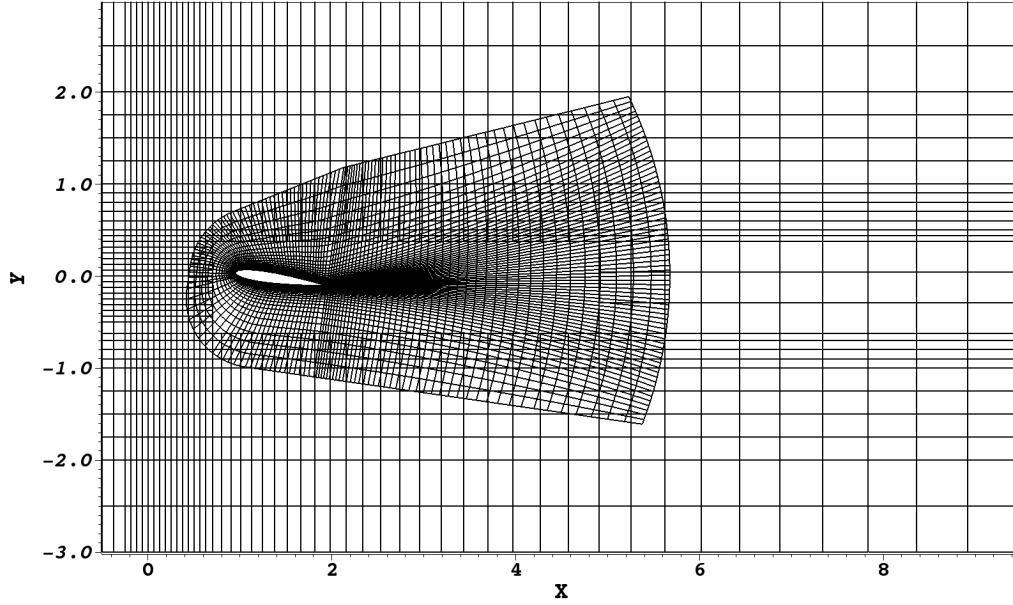


Figure 6.7: A cross section of the global computational domain for steady inflow simulations described in Section 6.2.3, showing element boundaries of the moving airfoil mesh within the stationary background mesh.

$\Delta s_U/c$	$\Delta s_L/c$	$\Delta n/c$	$\Delta z_{min}/c$	$\Delta z_{max}/c$
6.4×10^{-3}	9.0×10^{-3}	4.5×10^{-5}	1.3×10^{-3}	4.2×10^{-3}

Table 6.2: Airfoil grid parameters for simulations in Section 6.2.3. Refer to the caption of Table 6.1 for description of column headings.

The prescribed angle of attack for the NACA0012 airfoil pitching about its quarter-chord axis is prescribed as

$$\alpha = 15.3^\circ - 9.7^\circ \cos(2\pi ft), \quad (6.6)$$

where the reduced frequency (6.5) is $k = 0.16$.

Here, and in Section 6.3, simulation results are compared with the experimental results of Panda and Zaman [7] for the validation, who used probes in the wake of an oscillating airfoil to collect velocity and vorticity data of the flow in the wake. The unsteady lift force, reported in [7], is divided into a *non-circulatory* component and a *circulatory* component [178], where Panda and Zaman used the experimental

wake data to approximate only the circulatory component of the lift force acting on the airfoil. They argued that, while the addition of the non-circulatory component, which they express as

$$C_{l,NC}(t) = \pi\alpha_a \left(k\cos(2\pi ft) - \frac{1}{2}k^2\sin(2\pi ft) \right), \quad (6.7)$$

where α_a is the amplitude of the angle of attack oscillation, would change the values of the total lift force, the difference is fairly minor, and even negligible at small reduced frequencies [7]. This is illustrated in Figure 6.8 which shows circulatory lift data measured in [7] at $k = 0.16$ and the calculated non-circulatory component of lift. In the present simulations, the total lift coefficient values are compared with only the circulatory component of lift that is published in [7].

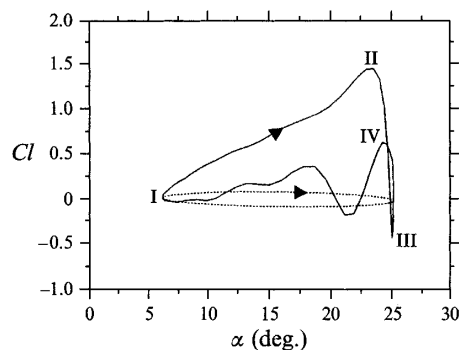


Figure 6.8: The measured circulatory component of lift for an airfoil oscillating at $k=0.16$ (solid line), and the calculated non-circulatory lift component (dotted line) as presented in [7]

Three methods are described in [7] for approximating the circulatory component of lift by integration of the phase averaged measured vorticity data. The authors state that an unknown steady contribution to the lift approximation exists due to vortices shed before the start of data collection, and thus the lift is assumed to be zero at the minimum angle of attack in their lift coefficient plots. In the plots below, data from [7] is shifted to give a lift value that matches present simulations at the

minimum angle of attack. The experimental data collected in the wake was phase averaged over 80 cycles [7].

In 2D simulations the average lift coefficient at 5.6° is $C_L|_{\alpha=5.6^\circ} \approx 0.5$, thus the lift coefficient values approximated and reported in [7] are shifted by a constant value of 0.5 in Figure 6.9. We again see that present lift approximations during upstroke correlate quite well with the experimental lift approximations of [7], though with more fluctuations, while the downstroke values do not match as well. The average of the last three cycles is displayed in Figure 6.9 (d), and the fluctuations are not as strong, though the downstroke data is still quite different from the experimental approximations.

Figure 6.10 presents the lift coefficient values for 3D simulations compared with the experimental approximations [7]. Performing phase averages for simulations over a large number of cycles, as was done for the experimental data, is not feasible considering the computational resources it would require, thus the instantaneous simulation data is compared with the phase averaged experimental approximations. The 3D simulation data correlates better with the experimental data than the 2D simulation data, especially in the downstroke phase. We see that while the general shapes of the curves are the same, the simulation data reports higher lift values than the experimental approximation, and though lift values during downstroke do not exactly follow the experimental approximations, the same features are present, namely a local minimum near $\alpha = 25^\circ$ followed by an increase in lift, then another major stall event. Following the major stall event during the downstroke, the lift values oscillate, in both simulation and experimental data, until the airfoil reaches 5.6° , when it begins the next cycle. It is expected that averaging 3D simulation data over many cycles would produce better correlation with the experimental results.

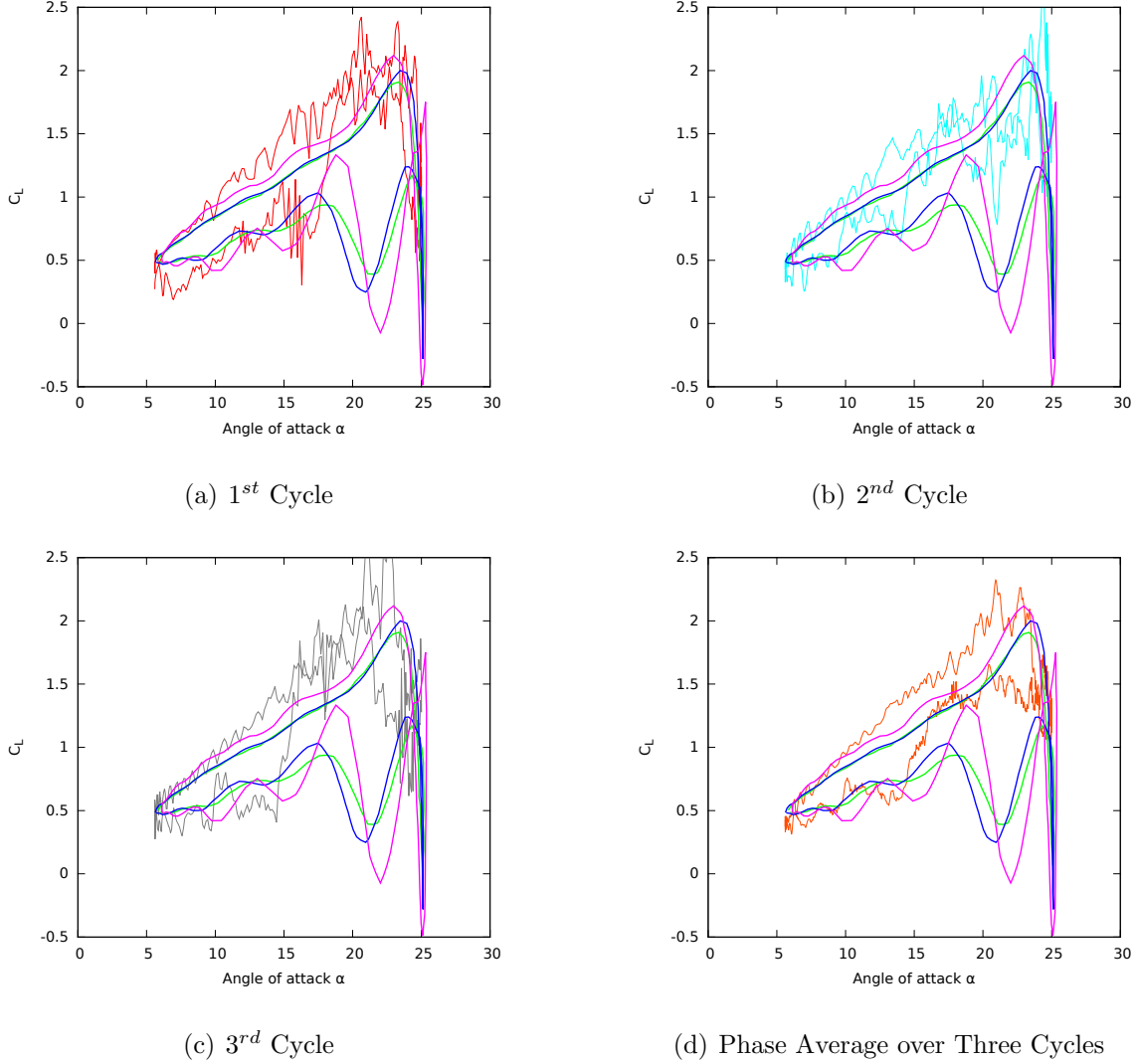


Figure 6.9: Lift coefficients from present 2D tests compared with approximations derived from experimental wake surveys by Panda and Zaman [7]. The green, blue, and magenta lines represent the circulatory component of the lift coefficient approximated from experimental wake data, which was phase averaged over 80 cycles, using methods 1, 2, and 3, respectively as published in [7].

A comparison of 2D and 3D simulations (Figure 6.11) shows that both simulations report similar maximum and minimum lift coefficient values, and display the same general features, though occurring at slightly different times. The main difference between the two sets of data is the much more dominant and sharp fluctuations seen in the two-dimensional values. Greater details will be given in the next section

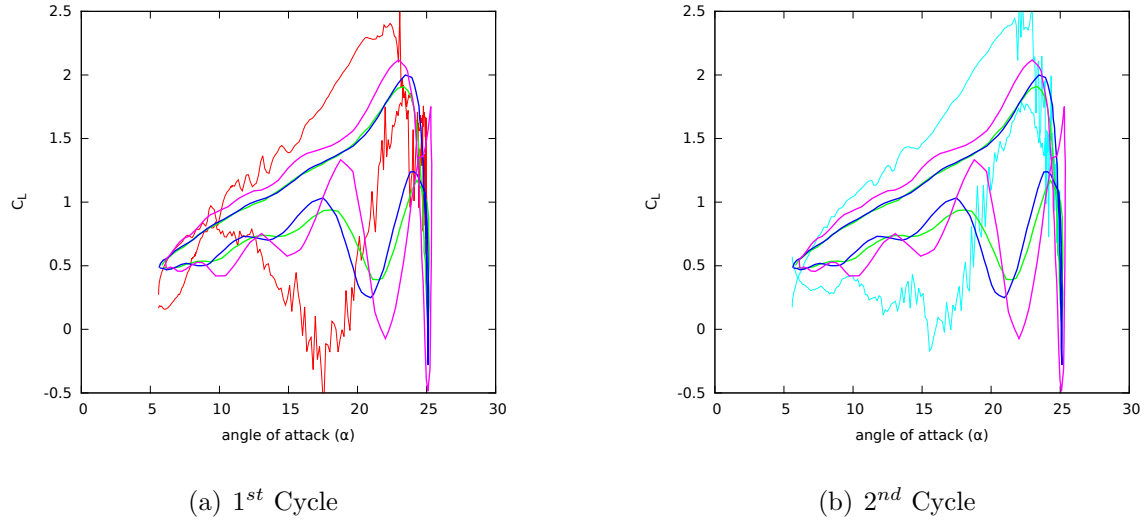


Figure 6.10: Lift coefficients from present 3D tests compared with approximations derived from experimental wake surveys by Panda and Zaman [7]. See Figure 6.9 caption for description of curves representing experimental approximations.

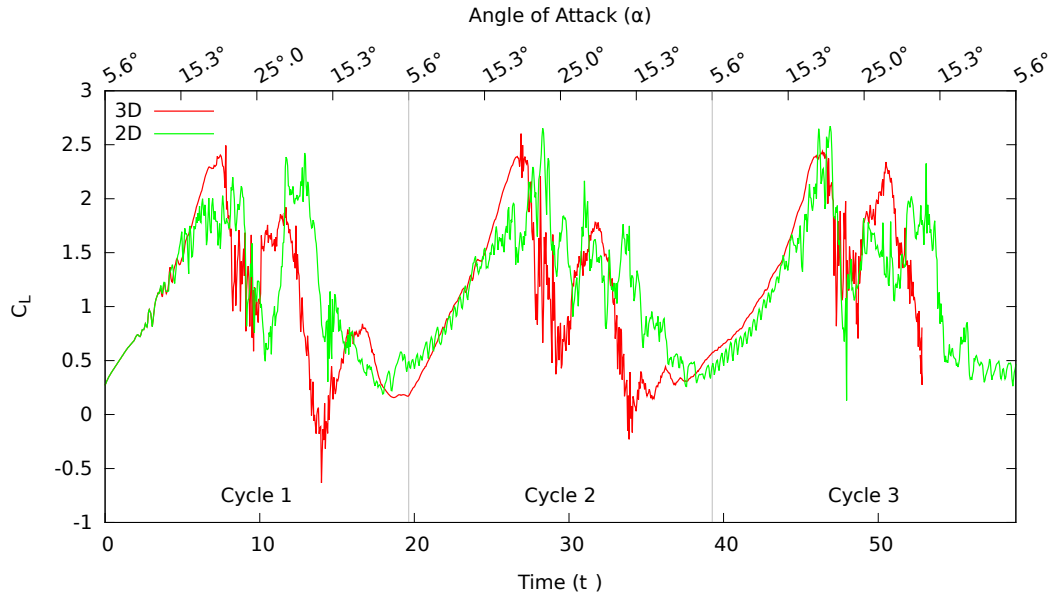


Figure 6.11: Comparison of lift coefficient values reported by 2D and 3D simulations.

regarding the 3D case for comparison with disturbed flow cases.

6.3 Influence of Upstream Disturbances

6.3.1 Problem Description and Setup

The present project simulates a NACA0012 airfoil with oscillatory pitching motion and chord based Reynolds number $Re_c = 44,000$. Its interactions with a turbulent wake generated by a small stationary cylinder, with diameter based Reynolds number $Re_D = 3900$, that disturbs the upstream flow are investigated. Three cases are investigated and compared as presented in Table 6.3. Case I corresponds to steady inflow (without a cylinder wake) which will serve as the baseline case and provide opportunities for validation with the previous experimental results [7]. Cases II and III represent two different scenarios for investigating the effects of upstream disturbances on the pitching airfoil, featuring different vertical positioning of the upstream cylinder with respect to the airfoil. In Case II, the cylinder is positioned at the same level as the airfoil pitching axis, and in Case III the cylinder's position is shifted vertically down. Thus, in Case II, disturbances impact directly on the leading edge of the airfoil for most of the pitching cycle, and in Case III, the effect of disturbances is mostly confined to the pressure side of the airfoil. Case II can be categorized as a stronger disturbance with respect to a steady inflow as compared to Case III.

Two meshes are used in the simulations: an inner airfoil mesh prescribed to move with the airfoil pitching motion, and an outer background stationary mesh, that either contains or does not contain a fixed solid cylinder inside. Overlapping meshes used to perform the simulations of Cases I, II and III are shown in Figure 6.12.

In all three cases, the global domain has a streamwise, vertical and spanwise lengths of $L_x/c = 10$, $L_y/c = 6$ and $L_z/c = 0.2$, respectively, where c is the airfoil chord length. The inner airfoil mesh is the same for all the cases, with grid spacings at the airfoil surface given in Table 6.4, and contains $\sim 54k$ elements ($\sim 19M$ gridpoints).

For the disturbed airfoil inflow cases, an outer background mesh contains an upstream cylinder with the diameter of $D \approx c/11.25$ triggering the disturbances. The horizontal distance from the center of the cylinder to the pitching axis of the airfoil located at the quarter-chord position is $L_x = 1.2c$, or, in terms of cylinder diameters, $L_x \approx 13.5D$. In Case II, the center of the upstream cylinder is on the same horizontal plane as the quarter-chord axis of the airfoil, and in Case III the cylinder is shifted down by $0.15c$, or $1.6875D$. The background mesh contains $\sim 16k$ elements ($\sim 3.4M$ gridpoints) for Case I and $\sim 21k$ elements ($\sim 4.5M$ gridpoints) for Cases II and III.

Dynamic timestepping procedure was developed for the moving multidomain simulations to ensure that sufficient temporal resolution is achieved while allowing efficient use of computational resources. Our dynamic timestepping method takes the mesh motion into account when calculating the Courant number for ALE simulations:

$$C_{ALE} = \max_i |C_{x_i,ALE} + C_{y_i,ALE} + C_{z_i,ALE}| \quad (6.8)$$

where i enumerates all GL collocation points in the domain. The Courant number for each coordinate direction is calculated

$$C_{x_i,ALE} = \frac{u_{i,rel} \cdot \Delta t}{\Delta x_i} \quad (6.9)$$

and the relative velocity $u_{i,rel}$ is calculated using the corresponding fluid velocity (u_i) and mesh velocity ($u_{i,mesh}$) components at a gridpoint i , with Δx_i being the characteristic stencil size in the corresponding direction,

$$u_{i,rel} = |u_i - u_{i,mesh}|. \quad (6.10)$$

Since the simulations are comprised of two subdomains, the maximum Courant number between the subdomains is determined and communicated at each timestep. When the maximum Courant number goes above or below certain limits, the timestep

is modified to keep the Courant number within the bounds, to yield computational efficiency while providing stability. This procedure ensures that both subdomains time-advance with the same timestep size.

Pitching airfoil simulations are performed using 6th-order polynomial approximations for spatial discretization within each element and dynamic timestepping with a maximum timestep of $\Delta t_{max} U_{\infty}/c = 2 \times 10^{-4}$. The average timestep in the simulations was approximately $\Delta t_{avg} U_{\infty}/c \approx 5 \times 10^{-5}$. Simulations use second order time integration with IEXT2 at interface boundaries and two intergrid iterations per time step. For all the cases I, II and III, the simulations were run for four airfoil oscillation cycles and took about 700 hours to complete on 1000 processors on SDSC Comet cluster.

The NACA0012 airfoil was constrained to move with an oscillatory pitching motion about its quarter-chord axis prescribed as

$$\alpha = \alpha_m - \alpha_a \cos(2\pi ft), \quad (6.11)$$

where α is the angle of attack, $\alpha_m = 15.3^\circ$ is the mean value, $\alpha_a = 9.7^\circ$ is the oscillation amplitude, and the frequency, f , is expressed in terms of the nondimensional reduced frequency, $k \equiv \pi fc/U_{\infty} = 0.16$. Note that the airfoil oscillates between 5.6° and 25.0° angles of attack in the current motion.

For the global (outer) domain, steady uniform velocity U_{∞} is prescribed on the far left boundary, outflow conditions on the right boundary, and symmetry conditions on the top and bottom boundaries. Periodic conditions are prescribed on the spanwise boundaries. The left, top and bottom boundaries of the inner (airfoil) mesh use interface conditions to pass values between subdomains, as described in the next section, while the right boundary is prescribed outflow conditions for a smooth propagation

of shed vortices across the interface. Moving wall boundary conditions with fluid velocity matching the airfoil velocity are prescribed at the surface of the airfoil.

Case	Position of Cylinder ($\frac{x}{c}, \frac{y}{c}$)	Position of Airfoil Quarter-Chord ($\frac{x}{c}, \frac{y}{c}$)
I	No Cylinder	(1.2, 0)
II	(0, 0)	(1.2, 0)
III	(0,-0.15)	(1.2, 0)

Table 6.3: Pitching airfoil cases (c is the airfoil chord-length).

$\Delta s_U/c$	$\Delta s_L/c$	$\Delta n/c$	$\Delta z_{min}/c$	$\Delta z_{max}/c$
6.4×10^{-3}	9.0×10^{-3}	4.5×10^{-5}	1.3×10^{-3}	4.2×10^{-3}

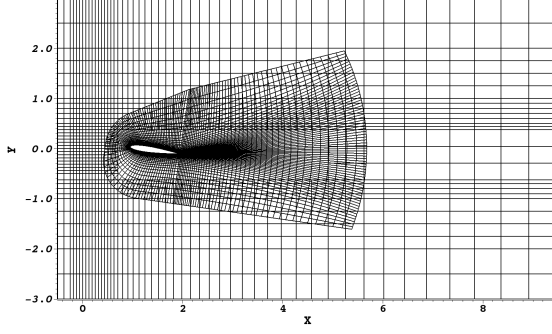
Table 6.4: Airfoil grid parameters for present pitching airfoil simulations: maximum streamwise collocation point spacing on the upper ($\Delta s_U/c$) and lower ($\Delta s_L/c$) surfaces of the airfoil, normal spacing on the upper airfoil surface at the mid-chord location ($\Delta n/c$), and the minimum ($\Delta z_{min}/c$) and maximum ($\Delta z_{max}/c$) spanwise collocation point spacing.

6.3.2 Oscillating Airfoil with Steady Inflow

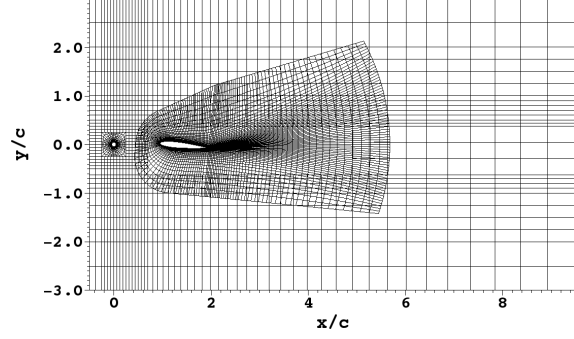
In this section, the results of Case I are discussed, which are used as a baseline for comparison with the two unsteady airfoil inflow cases. While some results for this case were presented in Section 6.2.3, more comprehensive results are presented here for comparison with Cases II and III.

Validation

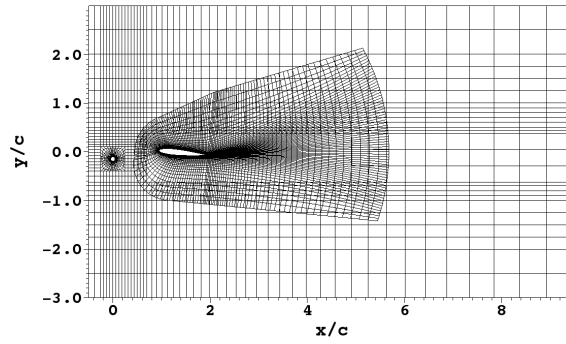
Simulation results are compared with the experimental results of Panda and Zaman [7] as discussed in Section 6.2.3. In the plots below, data from [7] is shifted (refer to Section 6.2.3) to give a lift value that matches the average lift in present simulations at the minimum angle of attack.



(a) Steady airfoil inflow (Case I)



(b) Disturbed airfoil inflow with an upstream cylinder (Case II)



(c) Disturbed airfoil inflow with an upstream cylinder shifted vertically (Case III)

Figure 6.12: Mesh configurations used in the current simulations of an oscillating airfoil. Only element boundaries are shown.

Figure 6.13 presents the lift coefficient values for present simulations compared with the experimental approximations [7]. The simulation was performed for a total of four cycles, while only the last three are shown in Figure 6.13. Phase averaging the simulation data over 80 cycles, as was done for the experimental data, is not feasible considering the computational resources it would require, thus the instantaneous simulation data is compared with the phase averaged experimental approximations that explains the prevalence of spikes in the simulation cases. We see that while the general shapes of the curves are the same, the present simulation data reports higher lift values than the experimental approximation, and though lift values during

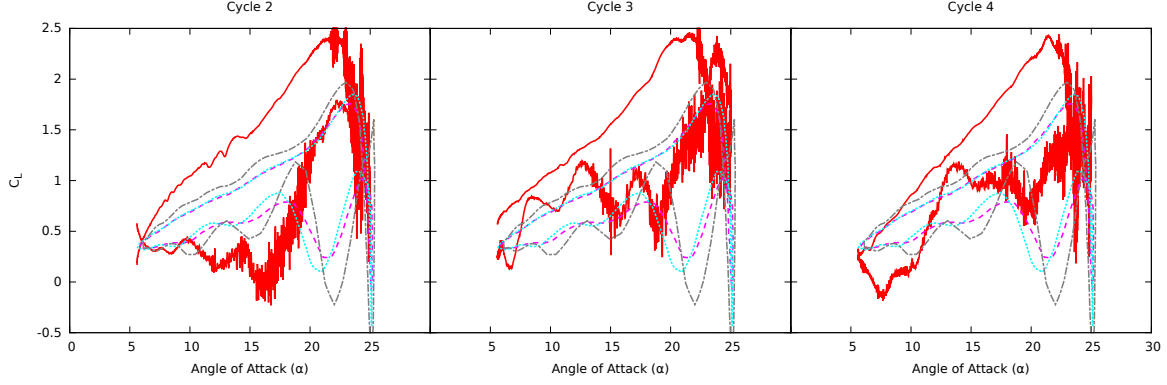


Figure 6.13: Lift coefficients from present tests compared with approximations derived from experimental wake surveys by Panda and Zaman [7]. The red solid lines represent present simulation data for Case I, while the magenta (dashed), cyan (dotted), and gray (dash dot) lines represent the circulatory component of the lift coefficient approximated from experimental wake data, which was phase averaged over 80 cycles, using methods 1, 2, and 3, respectively as published in Panda and Zaman [7].

downstroke (the lower curves) do not exactly follow the experimental approximations, the same features are present, namely a local minimum near the maximum angle of attack $\alpha = 25^\circ$ followed by an increase in lift at the downstroke, then another stall event. Following the major stall event during the downstroke, the lift values oscillate, in both simulation and experimental data, until the airfoil reaches its minimum angle of attack, $\alpha = 5.6^\circ$, after which it begins the next cycle. It is expected that averaging simulation data over many cycles would produce smoother curves and better correlation with the experimental results.

Physics of Dynamic Stall

Here, the physics of dynamic stall is investigated further, by performing flow visualizations and correlating the vortex dynamics observed in the flowfield with the pressure coefficient values, at different angles of attack, to be later compared with the disturbed airfoil inflow cases.

Visualizations of the the spanwise averaged streamwise velocity (Figure 6.14) and spanwise vorticity (Figure 6.15) at different angles of attack show the clockwise ro-

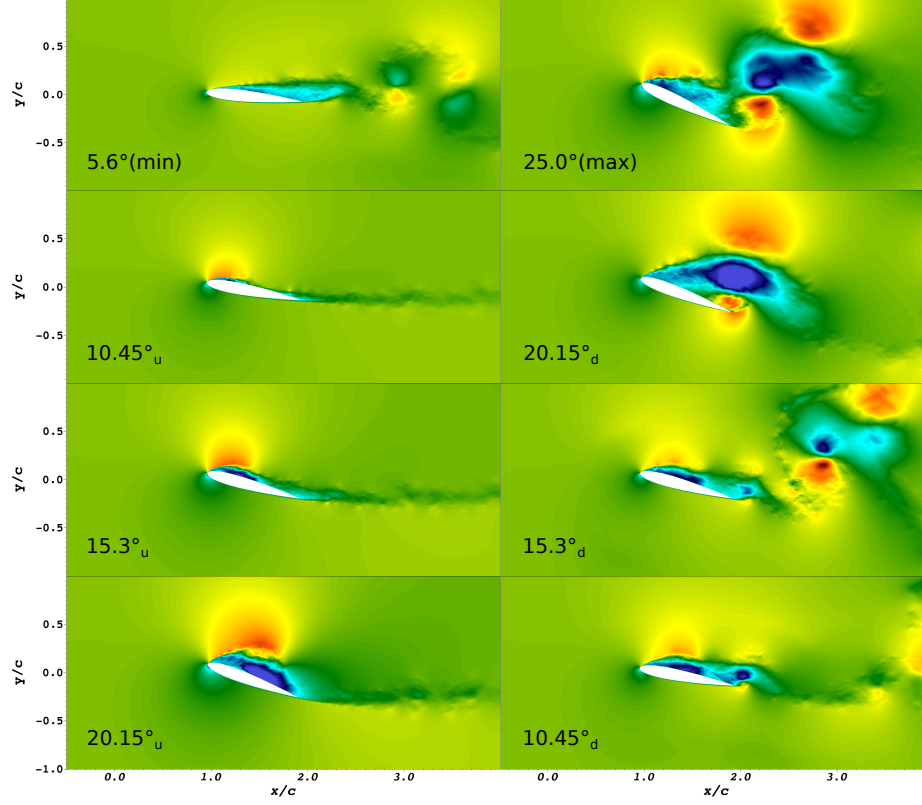


Figure 6.14: Zoomed-in spanwise averaged streamwise velocity plots during the third cycle, at angles of attack posted in bottom left corner of each snapshot. The subscript 'u' denotes upstroke and 'd' downstroke. Darkest blue represents streamwise velocity less than $u/U_\infty = -1$, and darkest red greater than $u/U_\infty = 3$

tating dynamic stall vortex (DSV) forming and traveling toward the trailing edge as the airfoil pitches upward. In the $\alpha = 15.3^\circ_u$ frame we find evidence of the boundary layer beginning to separate at the leading edge of the airfoil, which leads to the formation of a DSV seen in the $\alpha = 15.3^\circ_u$ frame. As the DSV nears the trailing edge, a counter-clockwise trailing edge vortex (TEV) forms, which appears in the $\alpha = 25^\circ$ frame. When the airfoil begins its downstroke, another clockwise vortex forms at the leading edge, which is smaller than the original DSV, and travels along the upper surface of the airfoil until it detaches, creating a secondary TEV and causing another stall event.

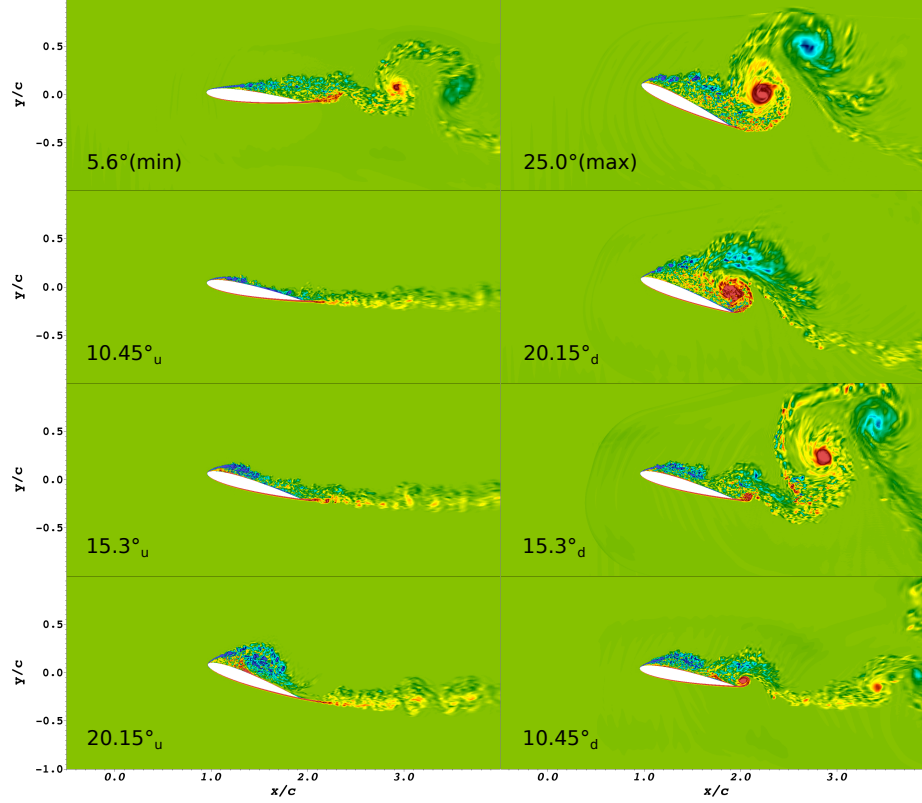


Figure 6.15: Zoomed-in spanwise averaged spanwise vorticity plots during the third cycle, at angles of attack posted in bottom left corner of each plot. The subscript 'u' denotes upstroke and 'd' downstroke. Darkest blue represents spanwise vorticity of less than $\omega_z U_\infty / c = -40$ (corresponding with clockwise fluid motion), and darkest red greater than $\omega_z U_\infty / c = 40$ (corresponding with counter-clockwise fluid motion).

By comparing the pressure coefficient values along the upper surface of the airfoil (Figure 6.16) with the streamwise velocity (Figure 6.14) and spanwise vorticity (Figure 6.15) plots we see that the position of low pressure valleys correlate with the location of vortex centers. Evidence of the forming DSV is found in the $\alpha = 10.45^\circ_u$ curve by the low pressure at the leading edge of the airfoil. We see that the low pressure region becomes stronger and propagates toward the trailing edge of the airfoil in the $\alpha = 15.3^\circ_u$ and $\alpha = 20.15^\circ_u$ curves. When the airfoil reaches its maximum angle of attack, the DSV and subsequent TEV have already detached. The $\alpha = 20.15^\circ_d$ curve shows evidence of a TEV resulting from the second major vortex system. After the second major vortex detaches, the pressure values along the suction side of the airfoil

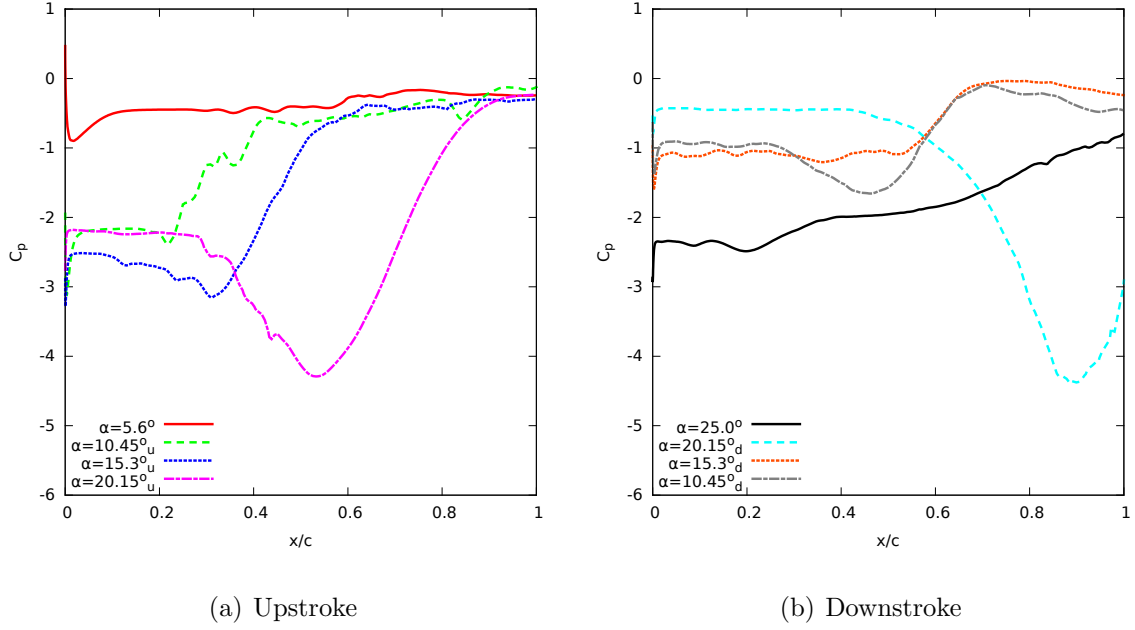


Figure 6.16: Pressure coefficient values along the suction surface of the airfoil at posted angles of attack during the third pitching cycle. The pressure coefficient values are averaged in the spanwise direction. The subscript 'u' denotes upstroke and 'd' downstroke.

begin to normalize to a relatively constant value as the boundary layer re-attaches at lower angles of attack.

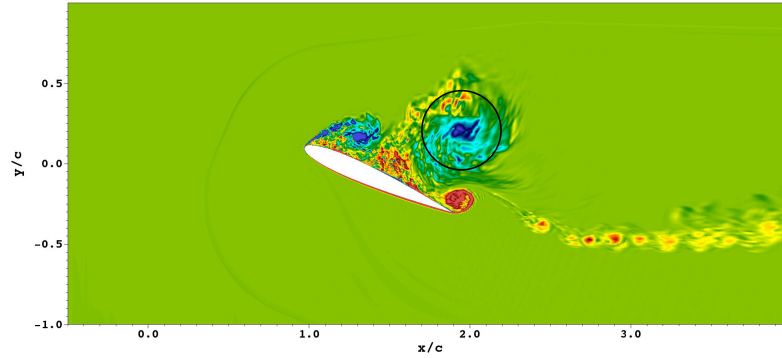


Figure 6.17: Spanwise averaged spanwise vorticity plot when the center of the DSV during the fourth cycle reaches $x/c = 1.95$ ($tU_\infty/c = 27.9$, $\alpha = 23.83_u^\circ$) for Case I. The circle outlines the volume over which the integral of vorticity is taken.

An indication of the strength of the DSV is obtained by performing a volume average of the spanwise vorticity over a cylindrical volume centered at the vortex

center:

$$\Gamma \equiv \frac{\left| \int_{V_{cyl}} \omega_z dV_{cyl} \right|}{V_{cyl}}, \quad (6.12)$$

where V_{cyl} is the volume of the cylinder being considered, and the non-dimensional spanwise vorticity, $\omega_z \equiv \omega_z^* U_\infty^* / c^*$. The center of the vortex is located by comparing the spanwise vorticity, streamwise velocity, and normal velocity components in the spanwise averaged fields. Ideally, a vertical line going through center of a vortex would show normal (vertical) velocities of zero, streamwise velocities that change sign at the center of the vortex which would correspond with a minimum or maximum spanwise vorticity. However, since the flow is turbulent, the best case was determined by careful examination. The strength of the DSV (Γ) is determined as the center reaches the $x/c = 1.95$ position in the flow field, and integration is performed over a cylindrical volume with the radius of $r/c = 0.25$, as illustrated in Figure 6.17, and the width, $l_z/c = 0.2$, which corresponds to the span of the domain. Table 6.5 presents properties of the DSV during the third and fourth pitching cycles.

	Third Cycle	Fourth Cycle
Vortex Location ($\frac{y}{c}$)	0.240	0.208
Time ($t U_\infty / c$)	47.8	67.1
Angle of Attack (α)	24.19_u°	23.72_u°
Vortex Strength (Γ)	10.271	9.899

Table 6.5: Properties of the DSV during the third and fourth pitching cycles of Case I as the center of the vortex reaches the $x/c = 1.95$ position in the flow field.

6.3.3 Cylinder Wake Validation

Before the results of the oscillating airfoil subject to impingement of a turbulent wake are presented, the wake turbulence data generated by an upstream solid circular cylinder will be validated, in the absence of the airfoil. Direct Numerical Simulations of a 3D circular cylinder is performed with diameter based Reynolds number $Re_D = 3900$, the same Reynolds number as in the coupled cylinder-airfoil simulation cases described below. The validation is done to ensure that good turbulent statistics are achieved for oscillating airfoil cases which are to be performed in the presence of a turbulent wake generated by an upstream cylinder.

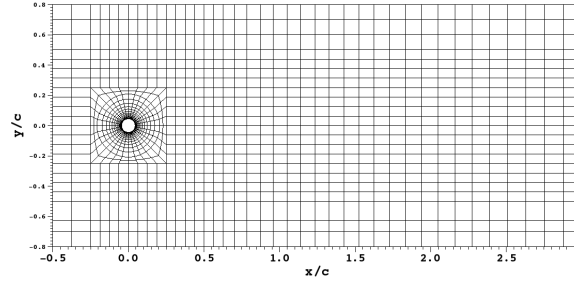
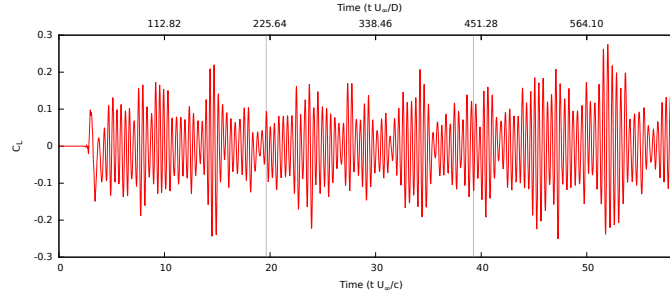


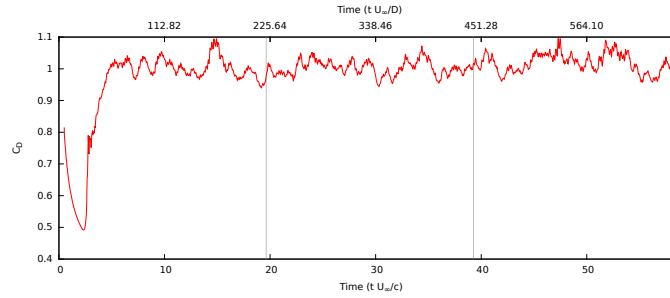
Figure 6.18: A cross-section view of the mesh used for simulations of the 3D cylinder wake. Only element boundaries are shown.

DNS simulations of a single cylinder are performed using a mesh shown in Figure 6.18 and are conducted with 6th-order polynomial approximations and a fixed time step $\Delta t = 5 \times 10^{-5}$. The mesh dimensions based on a chord length are $3c \times 1.6c \times 0.2c$ in streamwise, vertical and spanwise directions, respectively, which corresponds to $33.8D \times 18D \times 2.26D$ in terms of cylinder diameter. The normal distance from the surface of the cylinder to the nearest GL point is $\Delta n/D = 1.4 \times 10^{-3}$. The grid contains 15K elements (~ 3.2 M gridpoints). The mesh resolution, time step and numerical parameters of the 3D cylinder simulation are similar to the parameters that are used in oscillating airfoil simulations. This is done to ensure that the wake studied

here possesses the same properties as the wake incident upon the airfoil in simulations discussed in the next section.



(a) Lift Coefficient



(b) Drag Coefficient

Figure 6.19: Time history of force coefficients on a cylinder at $Re_D = 3900$. The bottom axis displays time nondimensionalized with the airfoil chord length and the top axis with the cylinder diameter. Grey lines denote the times at which new oscillation cycles begin in oscillating airfoil simulations.

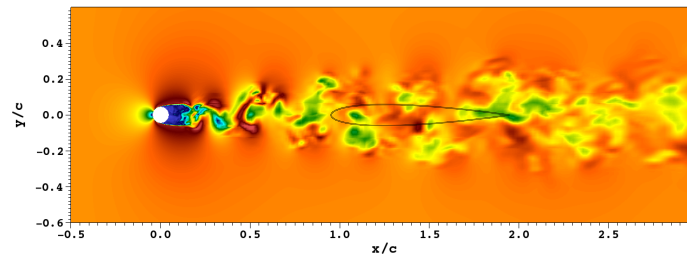


Figure 6.20: Velocity magnitude plot of a spanwise cross-section of the turbulent wake behind the cylinder at $t U_\infty/c = 19.6$ ($t U_\infty/D \approx 220.5$). An outline of the NACA0012 airfoil has been superimposed on the image to illustrate the location of the airfoil in the cylinder-airfoil simulations. Blue regions represent a value of 0 and red regions a value of $1.5 U_\infty$.

In pitching airfoil simulations, the airfoil begins pitching upward at $t = 0$, the same time as the cylinder simulations start from steady initial conditions, and thus the first cycle will serve to give the turbulent wake time to fully develop. The time history of the lift and drag forces acting on the cylinder is illustrated in Figure 6.19. The unsteady effects on the cylinder are fully developed before $tU_\infty/c = 10$ ($tU_\infty/D \approx 112.82$), allowing a sufficient time for the unsteady effects to propagate into the very far wake of the cylinder, where the airfoil will be located, before the second cycle begins. Figure 6.20 illustrates the velocity magnitude of the wake at $tU_\infty/c = 19.6$ ($tU_\infty/D \approx 220.5$), which is the time when the second pitching cycle of the airfoil will begin. An outline of the airfoil is superimposed upon the image to illustrate where the airfoil will be located in the cylinder-airfoil simulations. We see that in the time it would take the airfoil to complete its first cycle, the turbulence in the very far wake of the cylinder is fully developed.

The turbulent statistics for the wake behind the cylinder (without an airfoil) are displayed in Figures 6.21, 6.22, and 6.23 for different streamwise locations. Statistics were averaged over non-dimensional time of $tU_\infty/D \approx 100$ and were collected after the flow transitioned to fully turbulent.

Good correlation exists between present turbulent statistics and published data. The vortex shedding frequency of the present simulation gives a Strouhal number, $St \equiv fD/U_\infty = 0.216$, agreeing well with the experimental data of the same case presented by Cardell [185] which finds a Strouhal number of, $St = 0.215 \pm 0.005$.

While turbulent statistics are not generally collected in the very far wake ($x/D > 10$), it is the region of particular interest in this case, since it is the location in the wake where the airfoil will reside. Due to wake meandering in the very far wake, the statistics were averaged over non-dimensional time of $tU_\infty/D \approx 350$ to ensure smooth statistics. For the NACA0012 airfoil at $\alpha = 0^\circ$, the distances from the center of

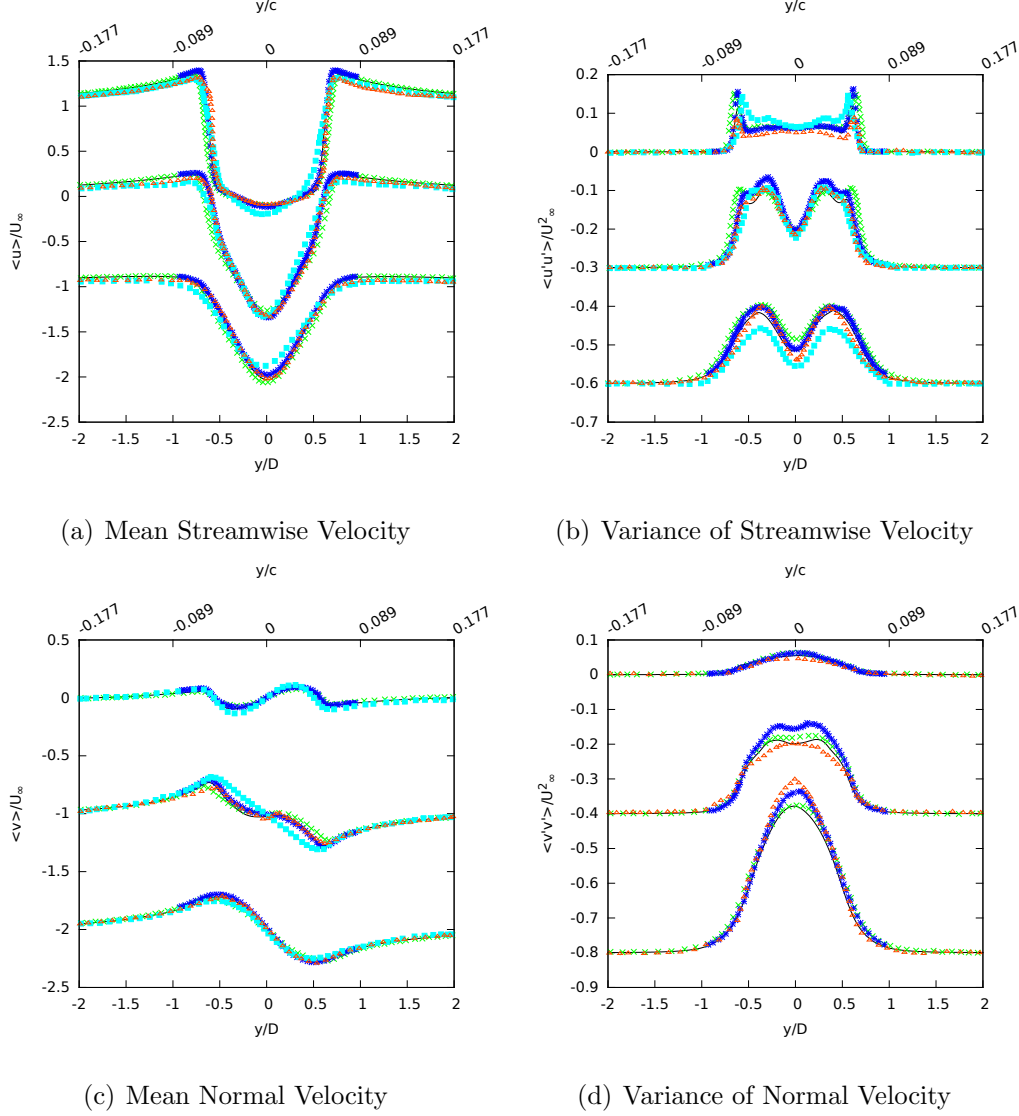
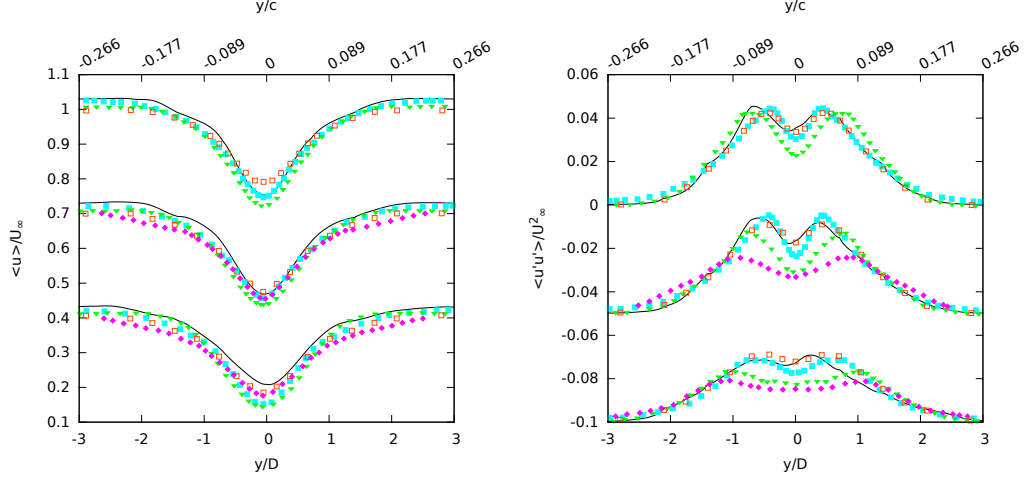


Figure 6.21: Turbulent wake statistics of the flow behind a cylinder at $Re_D = 3900$ at streamwise locations $x/D=1.06$ (top curve), 1.54 (middle curve), 2.02 (bottom curve). The middle and bottom curves have been shifted according to the conventions in [179]. Lines represent present data, green 'x' represent data from LES studies of Parnaudeau et al. [180], blue '*' represent data from PIV investigations of Parnaudeau et al. [180], light blue squares represent data from the B-Spline simulations of Kravchenko and Moin [179], and red triangles represent DNS data from simulations of Ma et al. [181]

the cylinder to the leading edge, quarter-chord, mid-chord, three-quarter-chord, and trailing edge locations on the airfoil are $x/D = 10.7, 13.5, 16.4, 19.2,$ and 22.0 , respectively. Figure 6.23 displays the mean streamwise velocity and variance of streamwise velocity at these locations in the wake. The wake width, defined by width of the



(a) Mean Streamwise Velocity

(b) Variance of Streamwise Velocity

Figure 6.22: Turbulent wake statistics of the flow behind a cylinder at $Re_D = 3900$ at streamwise locations $x/D=6.0$ (top curve), 7.0 (middle curve), 10.0 (bottom curve). The middle and bottom curves have been shifted according to the conventions in [179]. Lines represent present data, light blue squares represent data from the B-Spline simulations of Kravchenko and Moin [179], red open squares represent data from experiments of Ong and Wallace [182], magenta diamonds represent data from upwind finite difference simulations of Beaudan and Moin [183], and inverted green triangles represent data from central finite difference simulations of Mittal and Moin [184]

wake where the mean streamwise velocity difference is half of the difference at the centerline ($U_\infty - \langle u \rangle = \frac{1}{2}(U_\infty - \langle u \rangle_0)$), at the $x/D = 10.7$ location is about $0.13D$ or $0.012c$.

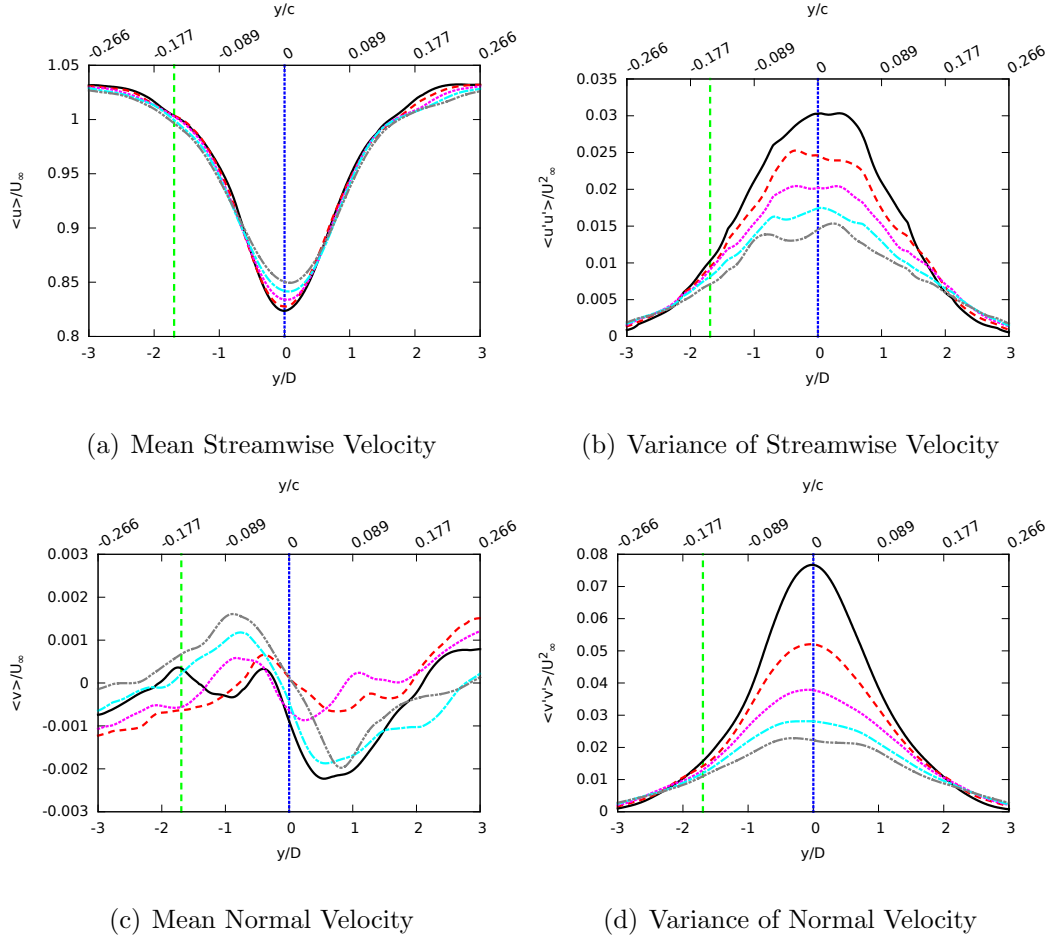


Figure 6.23: Turbulent wake statistics of the flow behind a cylinder at $Re_D = 3900$ at streamwise locations $x/D=10.7$ (black solid), 13.5 (red dashed), 16.4 (maroon dotted), 19.2 (light blue dash-dot), and 22.0 (grey dash-dot-dot). The blue (dotted) and green (dashed) vertical lines represent the position of the pitching axis of the airfoil relative to the wake for Case II and III respectively.

6.3.4 Pitching Airfoil in a Turbulent Wake

In this section Cases II and III that correspond to disturbed airfoil inflow cases are discussed. In Case II, the center of the upstream cylinder generating the freestream disturbances is placed on the same horizontal plane as the quarter-chord axis of the airfoil, and in Case III the center of the cylinder is $0.15c$ lower than the airfoil quarter-chord axis. The meshes for these simulations are presented in Figures 6.12(b) and 6.12(c), and the general description of the problem setup is done in Section 6.3.1.

In the coupled cylinder-airfoil DNS, the airfoil pitching simulations and the cylinder wake simulations are done concurrently (to ensure proper modification of the wake by the downstream airfoil), and the airfoil begins its first cycle at initially steady flow at time zero, before the cylinder wake has developed. It was demonstrated in the previous section that the cylinder turbulent wake is fully developed for the entire region wherein the airfoil resides by the beginning of the second cycle. Thus, the ensuing discussion will focus on the second, third, and fourth cycles of the airfoil motion (although the first cycle is often shown for completeness).

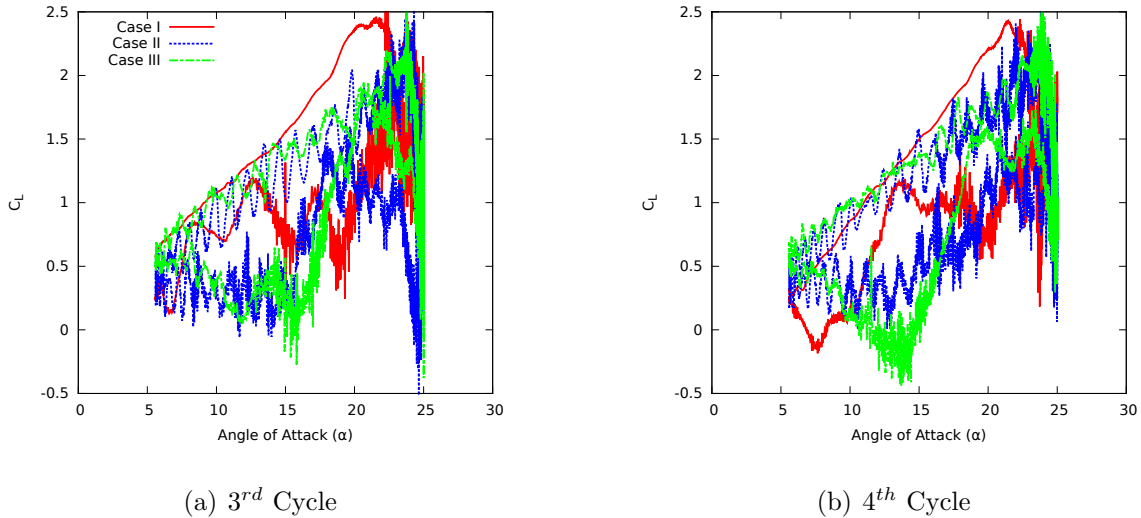
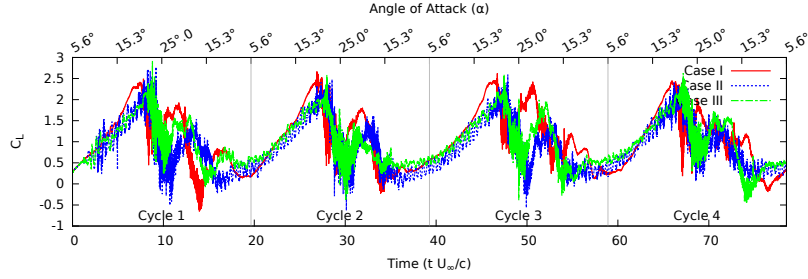


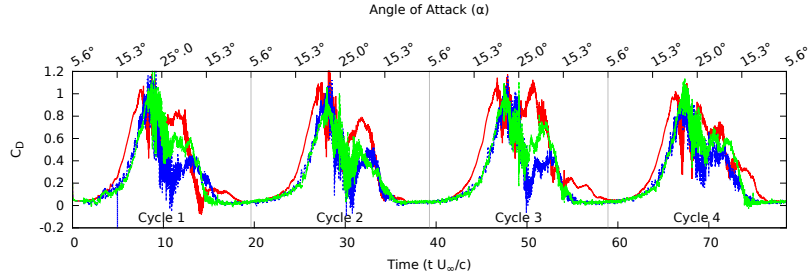
Figure 6.24: Lift coefficient results compared among all three cases for the third and fourth cycles.

A comparison of the lift coefficient loops among the three cases for the third and fourth cycles (Figure 6.24) shows stark differences in the lift that the airfoils experience in each case. The lift acting on the airfoil in Cases II and III during upward pitching (top curves) are quite oscillatory due to the incoming turbulent vortices shed from the upstream cylinder. However, note that the oscillations of lift values in Case III are not as large as those displayed by Case II, since in Case III the vortices are not directly incident upon the leading edge of the airfoil, but slightly below it. We see that in Case I, the lift increases at a slightly steeper rate starting at $\alpha \approx 15_u^\circ$, which corresponds to the formation of the DSV and its path along the upper surface of the airfoil. In Cases II and III we do not find evidence of a steeper rate of lift increase, and we find that in the high-lift regions, the lift in Case I is generally greater than the lift in Cases II and III, both at upstroke and downstroke. Judging by the position of the maximum lift peaks and a subsequent drop in lift, we see that dynamic stall occurs earlier (at smaller angles of attack) in Case I than for the airfoils with turbulent inflow. Details on the timing of the dynamic stall in each case will be discussed further in future sections.

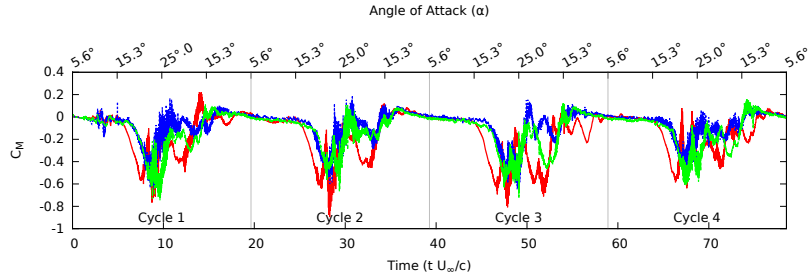
Figure 6.25 displays the aerodynamic forces and moments acting on the airfoil in Cases I, II, and III with respect to time. A comparison of lift coefficient maxima during the third cycle is presented in Table 6.6. We see that $C_{L,max}$ due to the DSV is comparable among the three cases presented, although the maxima are achieved at different times during the cycle. The secondary maxima of C_L , caused by the secondary vortex formed during downstroke, are noticeably different with Case I reaching the largest value, followed by Case III, then Case II. This pattern seems repeatable and is found in three out of four cycles during the simulation time (refer back to Figure 6.25). For both the DSV and secondary vortex, $C_{L,max}$ is reached earliest in Case I, next in Case III, and last in Case II, and this pattern is also



(a) Lift Coefficient



(b) Drag Coefficient



(c) Pitching Moment Coefficient

Figure 6.25: A comparison of aerodynamic forces and moments among Cases I, II, and III for all four cycles. The time ($t U_\infty/c$) is labeled along the bottom axis while the corresponding angle of attack is along the top axis.

found in the other cycles presented in Figure 6.25. We see that incident disturbances have the effect of delaying stall, similar to the experimental findings of Chen and Choa [173] who showed that stall of a constant rate pitching upward airfoil occurs at a later time when upstream disturbances are present. Additionally, it appears that larger disturbances incident upon the airfoil's leading edge (such as with in-line versus shifted vertical position of the upstream cylinder) have the effect of delaying the stall

Third Cycle			
	Case I	Case II	Case III
Maximum C_L	2.623	2.596	2.576
Time of $C_{L,max}$ ($t U_\infty/c$)	46.735	47.890	47.485
α of $C_{L,max}$	22.4_u°	24.3_u°	23.8_u°
Secondary Maximum C_L	2.426	1.593	1.947
Time of C_{L,max_2} ($t U_\infty/c$)	50.630	52.540	52.204
α of C_{L,max_2}	23.8_d°	19.7_d°	20.6_d°

Table 6.6: Comparison of lift coefficient maxima during the third cycle among all three cases

to a greater extent, as well as reducing the value of the secondary lift maximum ($C_{L,max}$). These effects can perhaps be explained by the fact that the incoming turbulence weakens the “native” vortical systems formed by a pitching airfoil motion, and this interaction is more pronounced for stronger incident disturbances, as in Case II.

Evidence of stall is seen in the drag values as well, with a steep decrease in drag when stall occurs. After initial stall occurs in Case I, a steep increase in drag soon ensues reaching drag values comparable to the first peak. This second drag peak correlates with the formation of the TEV, and a sharp decrease in drag is again witnessed as the TEV detaches. Cases II and III do not show clear signs of this large drag peak due to the formation of the TEV. While the drag acting on the airfoil in the turbulent inflow cases nearly reach the same global maximum values as those in Case I (though at a later time), the drag in nearly all other portions of the cycle is greatly reduced by upstream disturbances, with the larger incident disturbances of Case II creating a greater drag reduction, than Case III. We see particular evidence of this when looking at the drag peak due to the second major vortex formed during

the airfoil downstroke in the third cycle, where Case I reports a local drag maximum of $C_D \approx 1.125$ at $\alpha \approx 23.8^\circ$, Case II: $C_D \approx 0.512$ at $\alpha \approx 19.5^\circ$, and Case III: $C_D \approx 0.811$ at $\alpha \approx 22.6^\circ$. Others cycles of the simulations show similar patterns for the drag coefficient.

The *magnitude* of the pitching moment of the airfoil remains small until a large vortex begins to form on the leading edge of the airfoil, where a sharp increase in the *magnitude* of the pitching moment is seen and continues to increase until the vortex detaches from the surface of the airfoil. In Case I, large fluctuations in the pitching moments are seen for the majority of the cycle, while in Cases II and III, we see a large fluctuation in the pitching moment as a result of the DSV (though not as large as in Case I), but the pitching moment remains small for the remainder of the cycle. Again, we see that the larger incident disturbances in Case II diminish the magnitude of pitching moment more so than the off-set disturbances of Case III. A comparison of the phase averaged coefficient data (Figure 6.26) shows trends for lift, drag and pitching moments that are in agreement with the previous discussion.

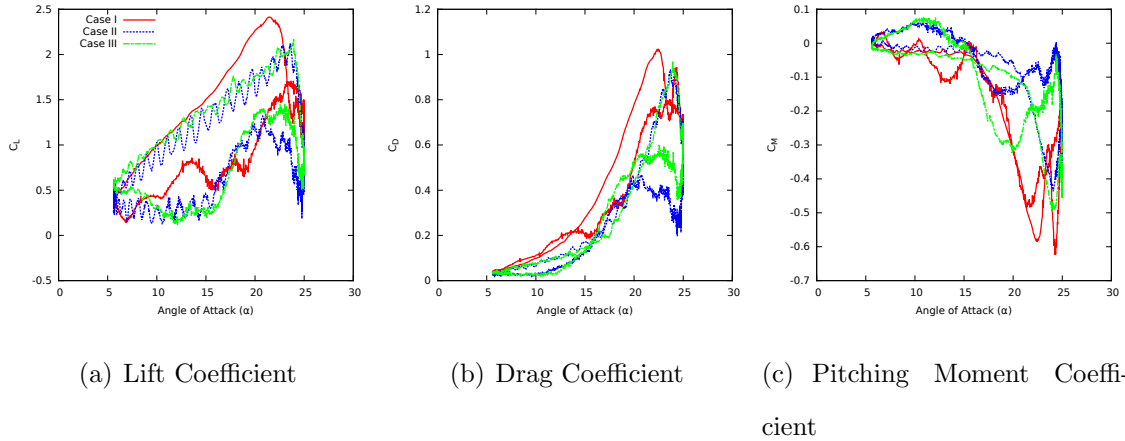


Figure 6.26: A comparison of aerodynamic forces and moments phase averaged over the last three cycles for Cases I, II, and III.

Visual inspection of the spanwise averaged streamwise velocity (Figures 6.27 and 6.28) and spanwise vorticity (Figures 6.29 and 6.30) for Cases II and III shows evidence

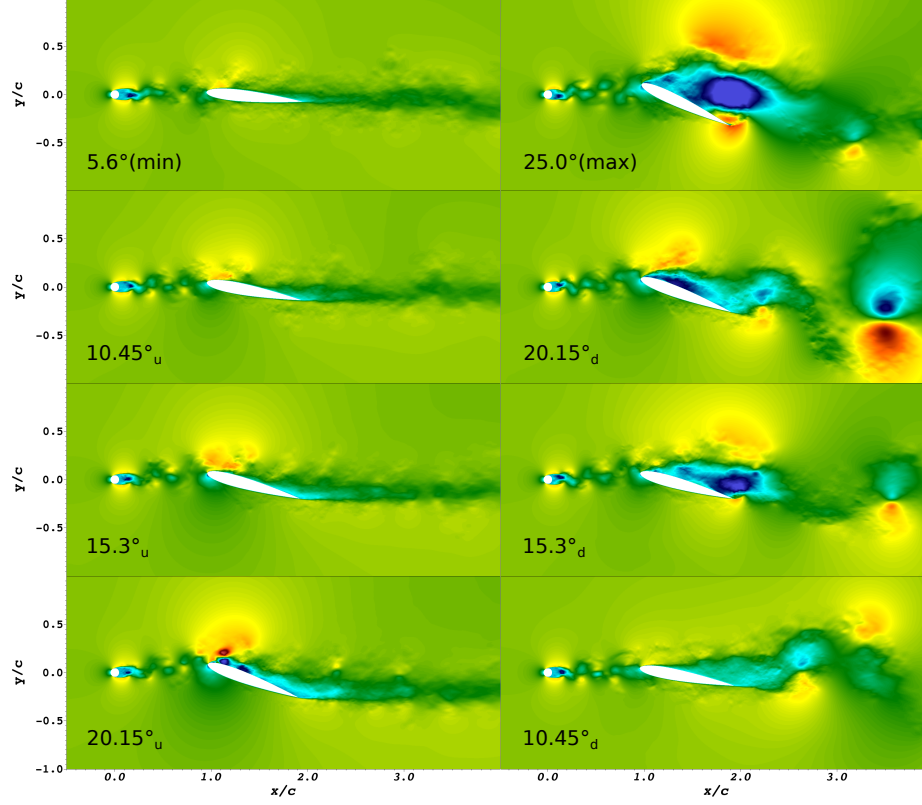


Figure 6.27: Zoomed-in spanwise averaged streamwise velocity plots of Case II during the third pitching cycle, at angles of attack posted in bottom left corner of each snapshot. The subscript 'u' denotes upstroke and 'd' downstroke. Darkest blue represents streamwise velocity less than $u/U_\infty = -1$, and darkest red greater than $u/U_\infty = 3$

of a forming dynamic stall vortex in the 20.15°_u frames, though not as large as the DSV seen in the 20.15°_u frames of Case I (Figures 6.14 and 6.15). However, note that the forming DSV is larger in this frame for Case III than in Case II. Thus we see that the formation of the DSV occurs earliest for the steady freestream case resulting in the earliest detachment of the vortex and subsequent dynamic stall among the three cases, as also evidenced in Figures 6.24 and 6.25. Again, since the disturbances in Case II are larger upon the airfoil due to the cylinder's vertical alignment, the formation of the DSV occurs the latest among the three cases, leading to the latest occurrence of dynamic stall.

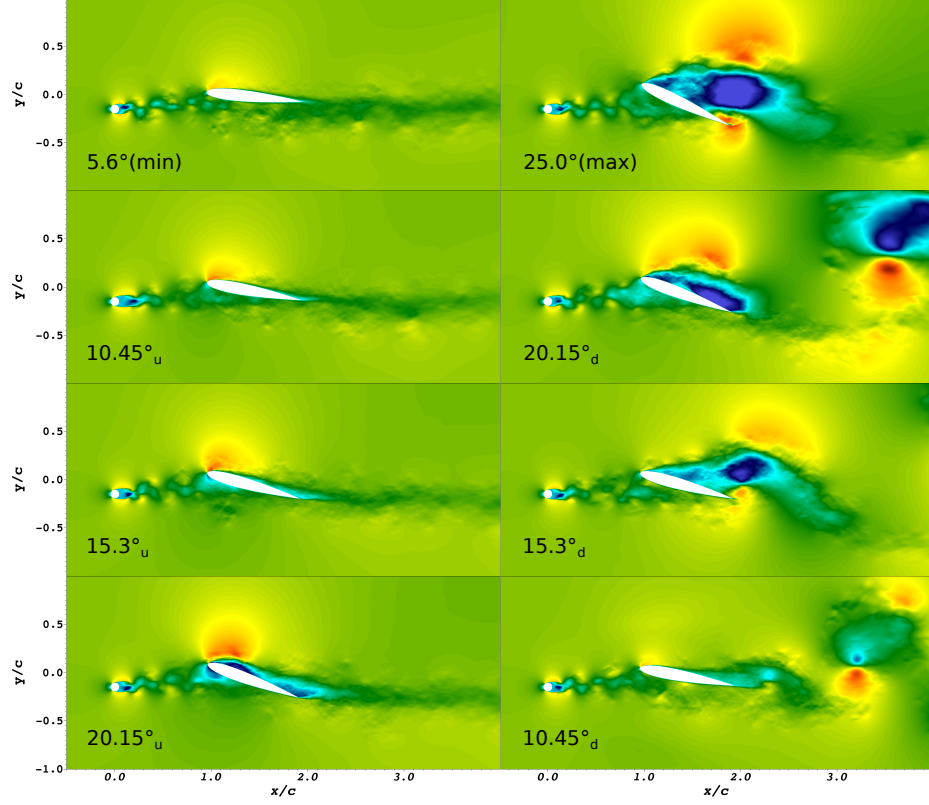


Figure 6.28: Zoomed-in spanwise averaged streamwise velocity plots of Case III during the third pitching cycle. Refer to the caption of Figure 6.27 for further details.

Averages of the aerodynamic coefficient values over the duration of the simulations are given in Table 6.7. We see that smallest aerodynamic coefficients are found in Case II followed by Case III and Case I, aligning with the previous analysis of aerodynamic forces where it is found that the larger incident disturbances given in Case II have the largest effect in reducing the force coefficient values, followed by Case III and then Case I.

A comparison of the pressure coefficients along the suction side of the airfoils in all three cases (Figure 6.31), again gives evidence of the timing of the DSV formation. The $\alpha = 20.15_u^\circ$ pressure profile plot shows that the DSV in Case I is roughly at the half chord distance (where the center of the vortex is determined by the lowest pressure values), the DSV is forming at the leading edge of the airfoil in Case II,

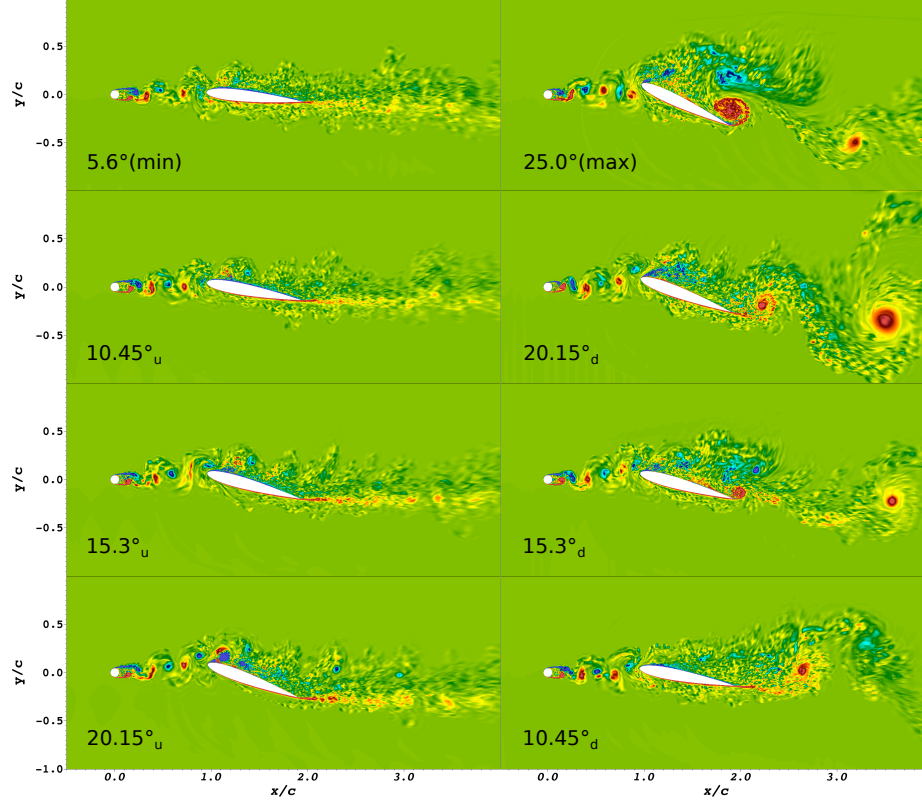


Figure 6.29: Zoomed-in spanwise averaged spanwise vorticity plots of Case II during the third cycle, at angles of attack posted in bottom left corner of each plot. The subscript 'u' denotes upstroke and 'd' downstroke. Darkest blue represents spanwise vorticity of less than $\omega_z U_\infty / c = -40$ (corresponding with clockwise fluid motion), and darkest red greater than $\omega_z U_\infty / c = 40$ (corresponding with counter-clockwise fluid motion).

and the DSV is roughly at the quarter-chord location in Case III. When the pitching airfoil begins its downstroke a second major vortex is formed at the leading edge. The $\alpha = 20.15^\circ_d$ plots shows evidence of the TEV in Case I resulting from the detachment of the second major vortex, while the pressure values in Case II show that the second major vortex is fairly weak and its center is located in front of the half chord location, and in Case III, the center of the second major vortex is at about the three-quarter chord location and appears to be much stronger than the vortex in Case II.

The strength of the DSV in Cases II and III is calculated with Eq. (6.12) as the center of the vortex crosses the $x/c = 1.95$ position in the flow field (Figure 6.32), the

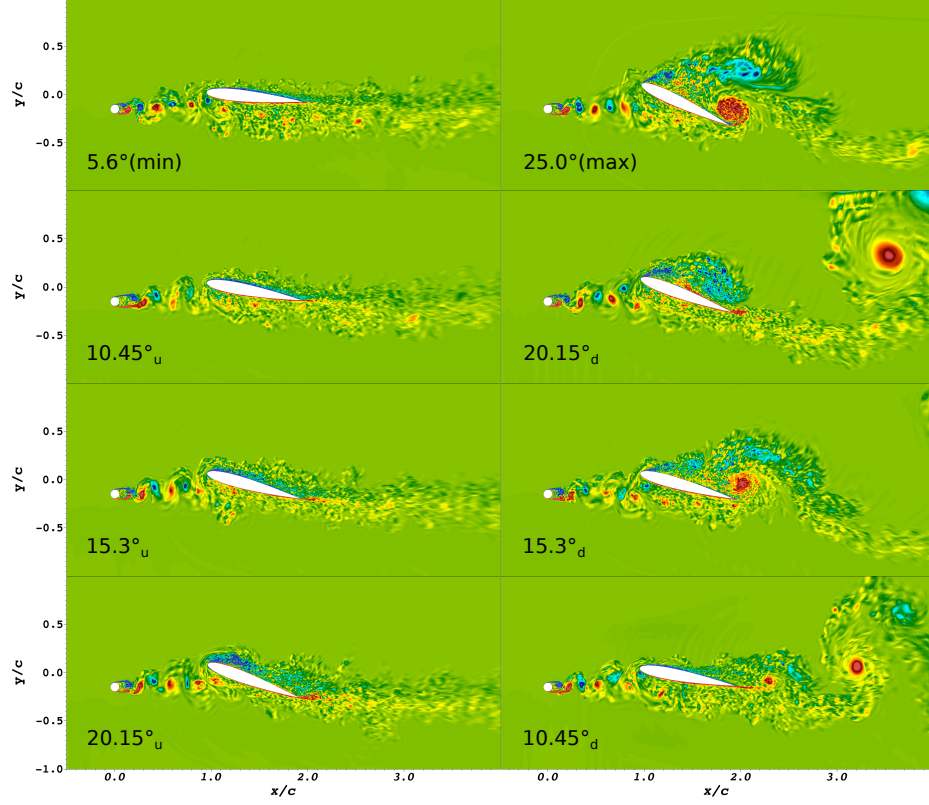


Figure 6.30: Zoomed-in spanwise averaged spanwise vorticity plots of Case III during the third cycle. Refer the caption of Figure 6.29 for further details.

Case I	C_L	1.0656
	C_D	0.3793
	C_M	-0.1606
Case II	C_L	0.8721
	C_D	0.2559
	C_M	$-6.8568 \cdot 10^{-2}$
Case III	C_L	0.9726
	C_D	0.2839
	C_M	-0.1108

Table 6.7: Time averaged values of the aerodynamic force and moment coefficients. The averaging was performed over the last three cycles of the simulation.

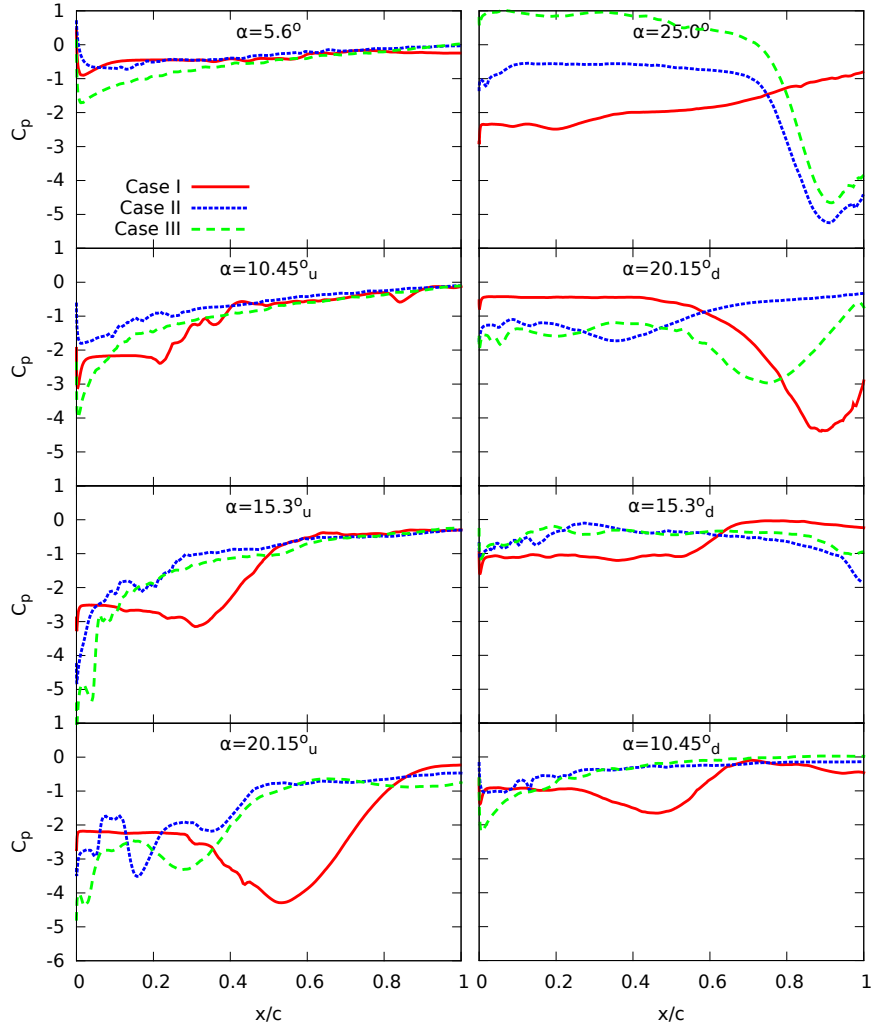


Figure 6.31: Spanwise averaged pressure coefficients along the suction surface of the airfoil for Cases I, II and III during the third cycle at angles of attack posted in the top center of each plot. The subscript 'u' denotes upstroke and 'd' downstroke.

	Case I (3rd Cycle)	Case I (4th Cycle)	Case II (3rd Cycle)	Case II (4th Cycle)	Case III (3rd Cycle)	Case III (4th Cycle)
Vertical position ($\frac{y}{c}$)	0.240	0.208	0.146	0.187	0.168	0.199
Time (tU_∞/c)	47.8	67.1	48.4	68.0	48.7	68.4
Angle of Attack (α)	24.19°_u	23.72°_u	24.77°_u	24.74°_u	24.93°_u	24.95°_u
Vortex Strength (Γ)	10.271	9.899	8.297	7.105	9.879	9.232

Table 6.8: Properties of the DSV during the third and fourth pitching cycles for all three cases as the center of the vortex reaches the $x/c = 1.95$ position in the flow field.

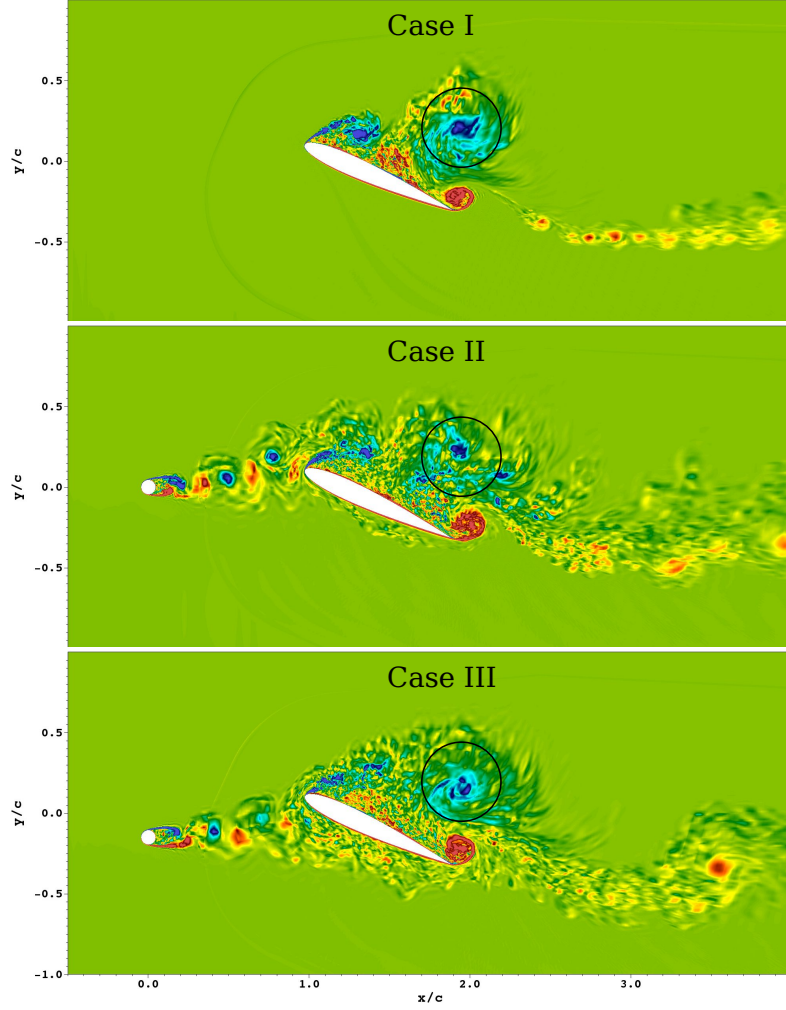


Figure 6.32: Spanwise averaged spanwise vorticity plots when the center of the DSV in the fourth cycle reaches the $x/c = 1.95$ position in the flow field for all three cases. The circle outlines the volume over which the integral of vorticity is taken.

calculation parameters are the same as in Case I in Section 6.3.2. A comparison of all the cases (see Table 6.8) shows that the DSV is strongest in Case I, next strongest in Case III, and weakest in Case II, aligning with the pattern seen in most of the other results presented here.

Recall that previous results suggested that the DSV forms earlier in the pitching cycle for Case III compared to Case II, yet note that the DSV in Case III crosses the $x/c = 1.95$ position at a later time than the DSV in Case II. A possible explanation to

this effect is as follows. In Case III, a strong counter-clockwise rotating trailing edge vortex forms early. As this TEV propagates upward, it creates a region with slower streamwise velocity in the downstream path of the DSV. This slower moving fluid decreases the propagation speed of the DSV. This effect is also witnessed in the other cases, though the timing of the DSV and TEV formation do not produce a response that is as impactful. These timing differences are presumably due to the turbulent energy being concentrated in different regions of the flow field. For example, in the 25° frame of Figure 6.29 it appears that the disturbances created by the cylinder in Case II play a larger role in the dynamics on the suction side of the airfoil, while in Figure 6.30 we see that the turbulent interactions are concentrated on the pressure side of the airfoil at its maximum angle of attack.

6.3.5 Power Spectral Densities of Aerodynamic Coefficients

The power spectral density (PSD) functions of the lift, drag, and pitching moment coefficients are computed by importing the aerodynamic data into the periodogram power spectral density estimate function in MATLAB. The aerodynamic coefficients were calculated within the moving overlapping mesh simulation at nondimensional time intervals of $\Delta t U_\infty / c = 0.05$, and PSD functions were calculated from data collected in pitching cycles two, three, and four, with cycle one being excluded since the turbulent wake is not fully developed at the beginning of the simulation.

Figures 6.33, 6.34, and 6.35 give a side-by-side comparison of the PSD function plots for lift, drag, and pitching moment coefficients respectively. As expected, the dominant frequency peak in every case corresponds with the pitching frequency of the airfoil, which shows that most of energy that is transferred to the airfoil occurs near this frequency that was imposed upon the system. In the lift plots (Figure 6.33), both Case II and Case III show another power peak corresponding with the vortex

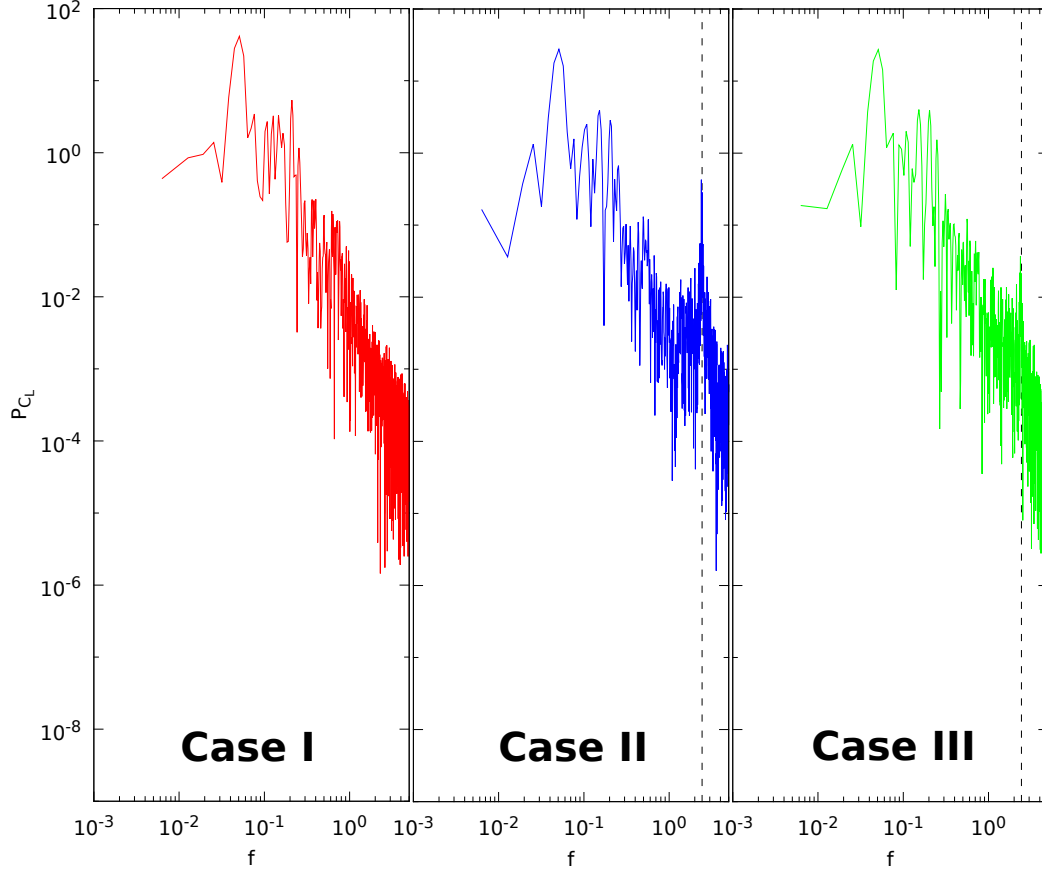


Figure 6.33: PSD functions of the lift coefficients for the three cases. The dashed lines in the Case II and III plots designate the vortex shedding frequency of the upstream cylinder.

shedding frequency of the upstream cylinder (indicated by dashed lines), suggesting that a moderate amount of energy affecting the lift acting on the airfoil is transferred by the turbulent wake. Note, however, that the peak at the cylinder vortex shedding frequency is not as prominent in Case III, likely due to the fact that the turbulent wake is not directly incident upon the airfoil's leading edge. Each of the Case II PSD function plots display this secondary peak, though in the case of the pitching moment coefficient it is not quite as pronounced, indicating that the higher frequency energy contributions from the upstream cylinder's wake are not as crucial to the pitching moments experienced by the airfoil as the forced pitching frequency imposed upon it.

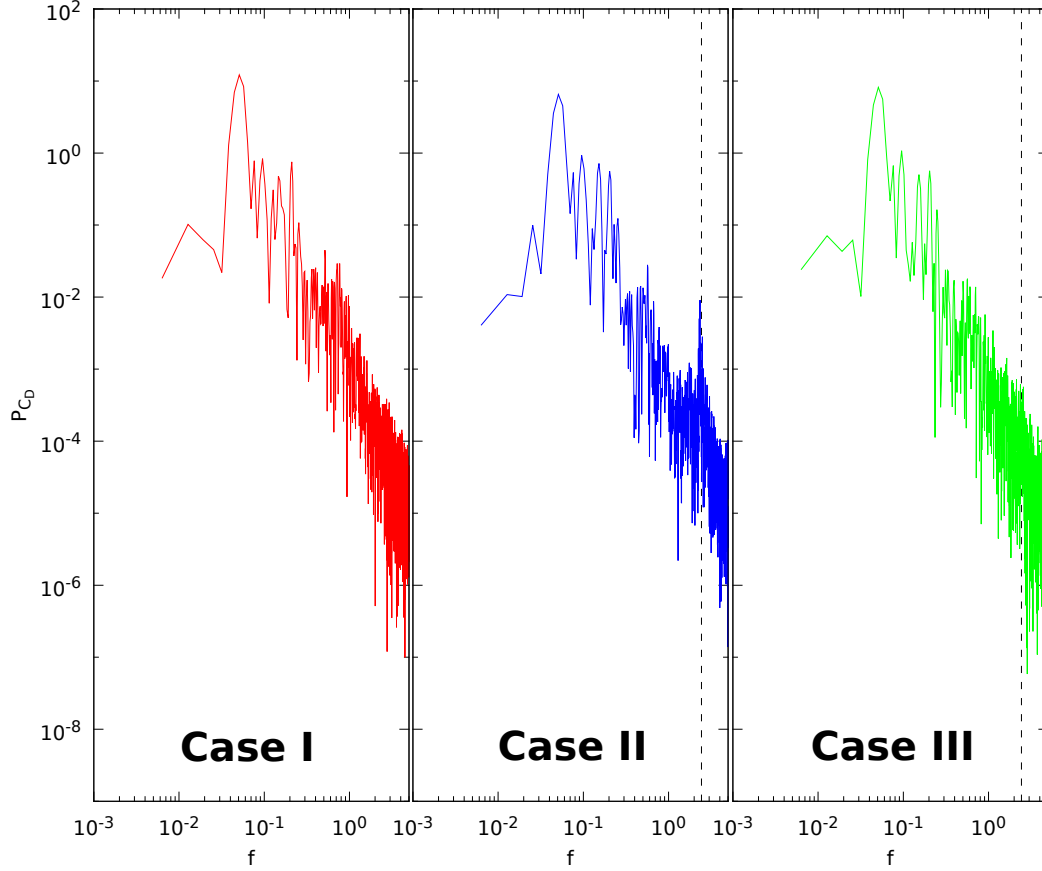


Figure 6.34: PSD functions of the drag coefficients for the three cases. The dashed lines in the Case II and III plots designate the vortex shedding frequency of the upstream cylinder.

In contrast, there are no noticeable peaks resulting from the upstream cylinder's wake in the drag and pitching moment PSD function plots for Case III. This suggests that the energy transferred to the airfoil due to the pitching frequency plays a much more important role in the determination of the drag force and pitching moment experienced by the airfoil in Case III. All of this additionally indicates that a turbulent wake incident upon the leading edge of a pitching airfoil is likely to play a larger role in the forces and moments acting on the airfoil than in the case with the same turbulent wake but incident upon the pressure side of the pitching airfoil. This is consistent with the previous discussion showing stronger interaction of incoming turbulence with the airfoil vortices in Case II.

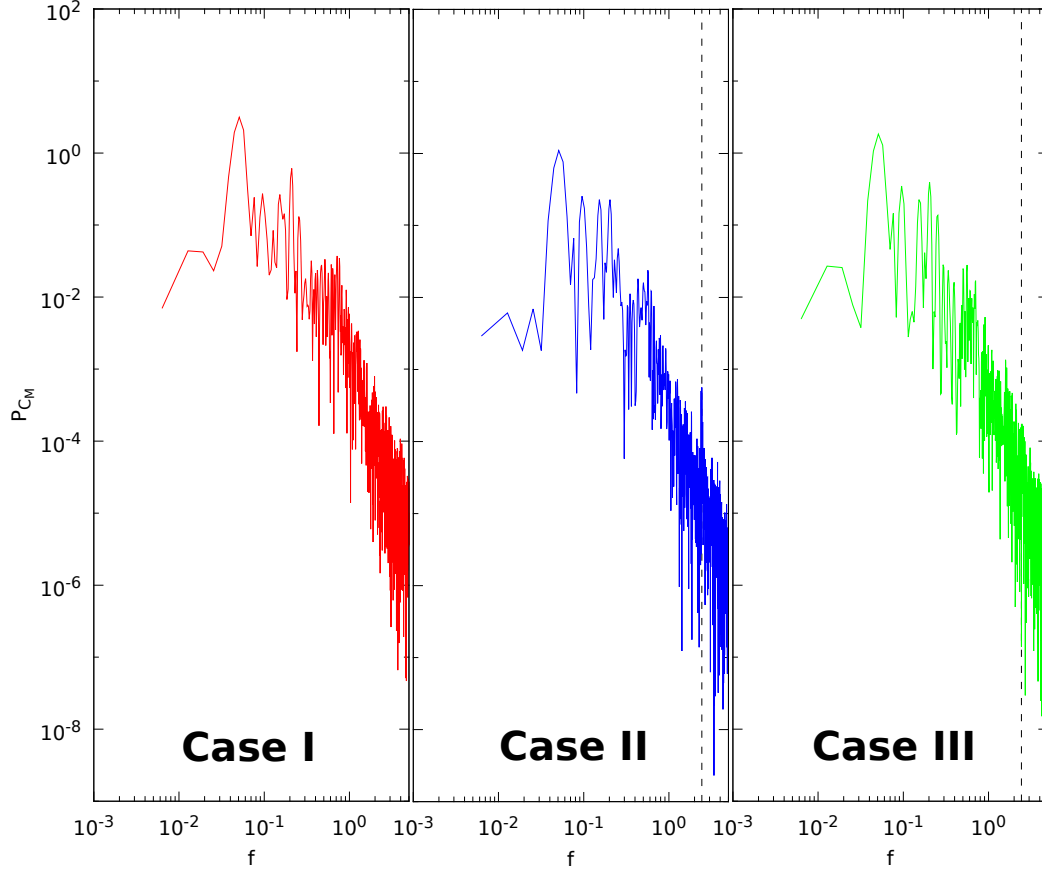
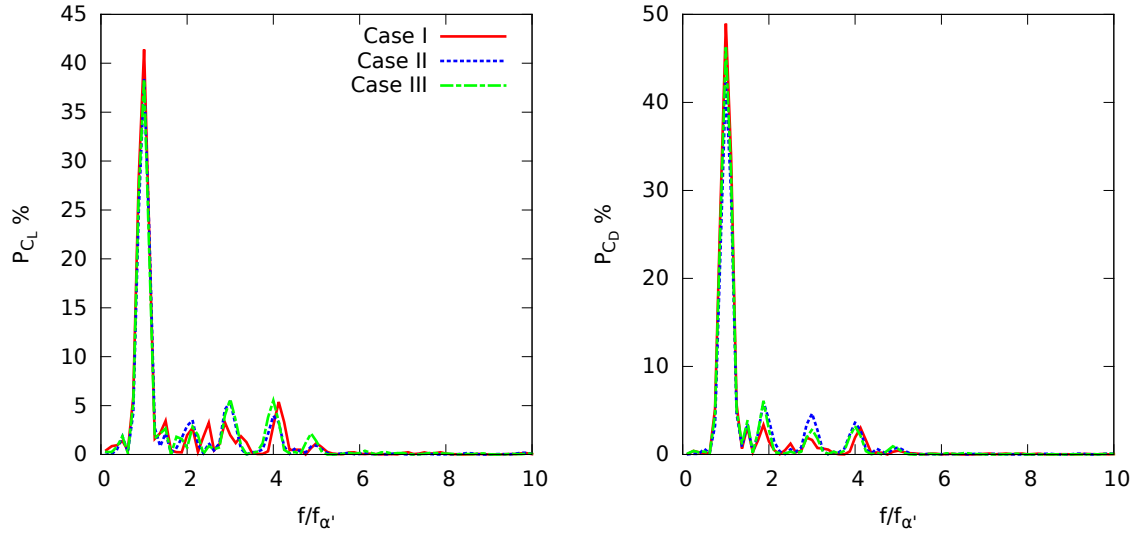


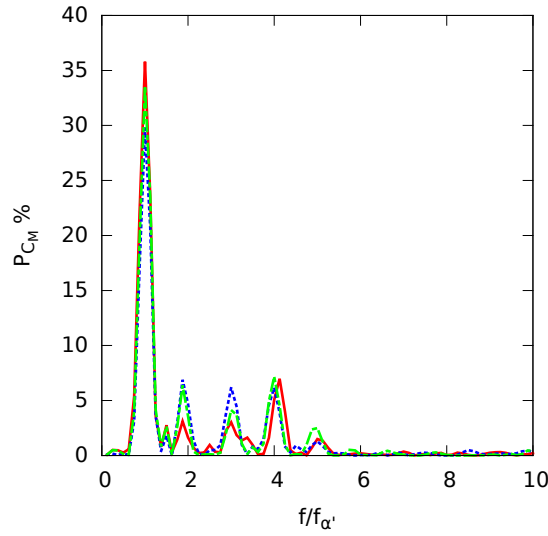
Figure 6.35: PSD functions of the pitching moment coefficients for the three cases. The dashed lines in the Case II and III plots designate the vortex shedding frequency of the upstream cylinder.

Following the normalization procedure employed in [186], I now focus on the lower frequency region of the PSD function plots. In Figure 6.36, the frequency is normalized by the pitching frequency of the airfoil ($f = f/f_{\alpha'}$), so that “1” now represents the pitching frequency. Additionally, the PSD function is normalized by the total power in the full range of frequencies computed ($P/\int_0^\infty P(f)df$), allowing for more direct comparison of PSD functions among the three cases. The frequencies lower than $f/f_{\alpha'} = 10$ are displayed in Figure 6.36, where we clearly see higher-order harmonics in every case which are prominent until about the fifth harmonic. In general, the peaks at the pitching frequency and at the harmonics correspond quite nicely among the three cases, though slight differences are apparent. In all of the



(a) PSD of C_L

(b) PSD of C_D



(c) PSD of C_M

Figure 6.36: PSD functions of aerodynamic coefficients for all three cases. The power is normalized by the total power of the function within the frequency range examined, and the frequency is normalized by the pitching frequency of the airfoil.

plots, the peaks at the harmonics are generally larger in Cases II and III than in Case I. The harmonics, as discussed in [186], indicate that there is a non-linear interaction between the pitching airfoil and the fluid which transfers energy from the pitching

frequency to the harmonics, while larger harmonics denote more prevalent non-linear interactions.

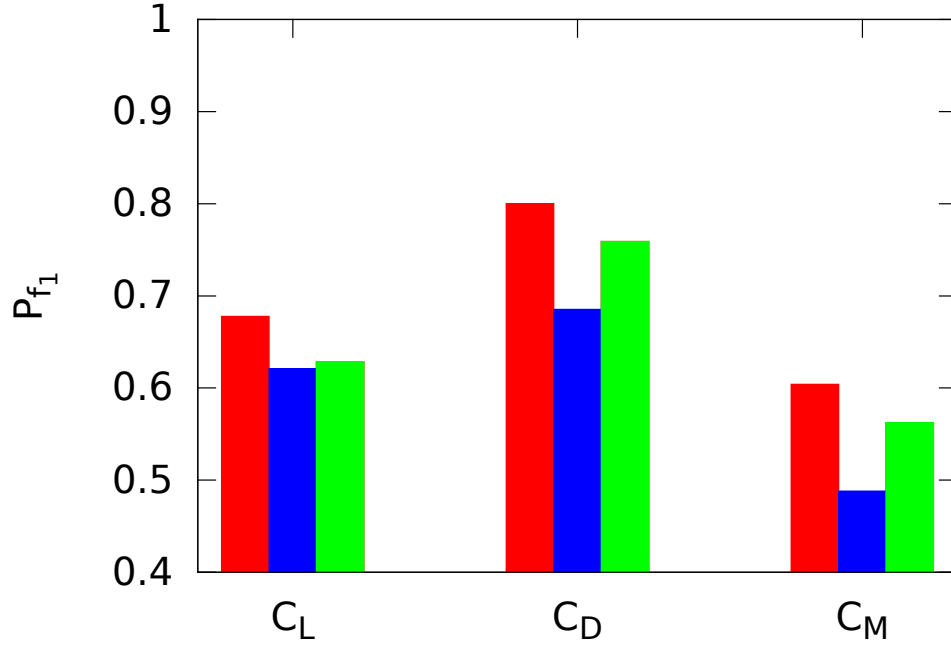


Figure 6.37: The energy partition derived from the aerodynamic coefficient PSD functions at the airfoil pitching frequency for Cases I (red, leftmost bars), II (blue, center bars), and III (green, rightmost bars).

In each of the plots, the PSD function at the airfoil pitching frequency is largest for Case I. The main peaks of Cases II and III in the PSD plot of C_L lie almost directly on top of each other, while careful inspection of the C_D and C_M PSD plots shows that the main peak for Case III is larger than that of Case II. The energy partition at the pitching frequency is calculated to achieve a more clear indication of the prominence of non-linear interactions in each case [186]. The energy partition at the pitching frequency is calculated by

$$P_{f_1} = \frac{\int_{0.5}^{1.5} P(f)df}{\int_0^{\infty} P(f)df}, \quad (6.13)$$

where a value of “1” would indicate the absence of non-linear interactions, or that all of the energy lies at the fundamental frequency. Thus we should expect that nonlinear

interactions would be greater in a system that introduces upstream turbulence as we do in Cases II and III, indicated by lower energy partition values at the pitching frequency. Figure 6.37 displays the energy partition values at the airfoil pitching frequency for the forces and moments in all three cases. We, in fact, see that the values at this energy partition are greater for Case I than for the other cases, and Case III is greater than Case II in every case, though only slightly for C_L . This implies that the greatest non-linear interactions occur in Case II where the turbulence is more concentrated on the suction side of the airfoil (where the DSV vortices form), followed by Case III where the turbulence is more prominent on the pressure side of the airfoil, with non-linear interactions playing the smallest role in Case I performed in the absence of an upstream wake. Also note that non-linear interactions appear to play the largest role in the moments experienced by the airfoil (indicated by the lowest energy partition values), which corresponds well with the results displayed in the C_M plot of Figure 6.25 that shows major reductions in the magnitude of pitching moments throughout nearly the entire cycle when the pitching airfoil is in the presence of a turbulent wake.

Chapter 7

CONCLUSION

7.1 Summary

A moving overlapping grid methodology for two and three dimensional applications in a Spectral Element Method incompressible flow solver has been developed and validated. Spectral accuracy is maintained for spatial discretization due to a spectral interpolation at interface boundaries using solution approximations in a N th-order polynomial space on GL points. The global temporal accuracy of the flow solver is also maintained with few, or no, iterations using solutions at previous timesteps to form explicit extrapolation approximations at subdomain interfaces.

The stationary and moving overlapping grid methodologies have been validated with several two- and three-dimensional simulations, comparing results to exact solutions and experimental data. Results from stationary, rotating, and sliding mesh simulations of two dimensional convecting eddies were compared with exact solutions. As the polynomial order for solution approximations is increased, spectral convergence of the global solution is observed. As the size of timestep is decreased, keeping the polynomial order constant, the temporal accuracy of the global scheme is maintained, as long as the interface extrapolation order is consistent with the global temporal accuracy of the method. The errors of solution approximations in moving subdomains is shown to increase quadratically/linearly with increasing mesh velocity using a second/first order temporal scheme, respectively, though the spatial and temporal convergence rates are unaffected.

Stationary overlapping mesh simulations of an incompressible Taylor vortex convecting through a long two-dimensional global domain helped form a more complete understanding of errors as flow features cross subdomain boundaries, and where disturbances due to outflow boundary conditions are present downstream. It was observed that as the vortex crossed a subdomain boundary, overall errors remained small in the presence of outflow disturbances. The three dimensional fully-developed turbulent pipe flow simulation on two overlapping meshes showed good agreement with other physical and computational experiments. Turbulent statistics closely matched those published in [142], including Re_c , Re_τ , mean velocity profile, and u_{rms}^+ profile data.

Moving overlapping mesh simulations of an oscillating two-dimensional cylinder produced results that compare well with experimental and other computational studies. Approximated aerodynamic forces were in line with published computational data, and simulations showed lock-in behavior of vortex shedding matching experimental findings [2]. Three-dimensional rotating sphere tests give lift, drag and vortex shedding results that are in agreement with published data.

Parallel scaling and timing analyses show that both methodologies achieve near linear strong scaling up to moderately high processor counts.

A turbulent wake incident upon a pitching airfoil largely affects the forces and moments acting on the airfoil, and the flow structure of the fluid, including vortex formation. Dynamic stall of a pitching airfoil occurs at a later time (and larger angle of attack) when in the presence of a turbulent wake, due to the delayed formation, and thus detachment, of the dynamic stall vortex. While the lift in a high-lift region of a pitching cycle is slightly reduced with upstream disturbances, the drag and magnitude of pitching moments are significantly reduced for most of the pitching cycle. This implies that upstream disturbances can have both negative and positive effects

on the performance of engineering systems, depending on the goals. For high-lift and maneuvering aircrafts, the decrease in peak lift means reduced maneuverability and implies the need to take it into account during pilot training and operation, as well as in control systems development. However, the reduction in mean drag leads to higher efficiencies of cyclic operations, such as in rotary-wing flight and wind turbine power production, implying that formation flight and wind turbines in an array can actually lead to an increase in performance with well designed positioning and control. Lower pitching moments additionally imply a decrease in the structural strain on dynamically pitching blades and wings, potentially improving the life span of these mechanical systems. Weaker dynamic stall vortices shed from the airfoil in disturbed upstream flow could lead to less intrusive vortex interactions with downstream structures, as in the case of blade-vortex interaction.

7.2 Future Directions

Future development efforts will improve the moving mesh methodology for greater flexibility and ease-of-use through dynamic hole cutting procedures to automate the process of determining vacancies in the background mesh in real-time as the interior mesh moves. The methodology will be enhanced to allow for global domain decomposition into more than two overlapping subdomains, increasing flexibility to simulate flows around even more complex stationary or moving geometries. Extensions to other SEM formulations will be made, such as to the unstaggered grid formulation with pressure approximations continuous at element boundaries (known as P_n - P_n), thus requiring interpolation of pressure at interfaces. Future developments will establish 6 degree-of-freedom (6-DOF) modules for rigid body motion determined by flow forces, rather than prescribed rigid body motion. This work will eventually provide a framework for fluid-structure interaction problems allowing mesh deformations.

The moving overlapping mesh methodology, in its present form, enables the investigation of many interesting and complex problems and applications. Hence, future work will utilize the moving overlapping mesh methodology to perform additional dynamic stall simulations with fully-developed turbulent inflow and with other types of airfoil motion, such as plunging. Additionally, dynamic stall at higher Reynolds numbers will be investigated for better application to conditions seen in flight and wind energy. The methodology will be used in rotating wind turbine simulations to investigate the influence of atmospheric turbulence and the impact of other wind turbines on the flow. Additional applications can be made to turbomachinery and rotor-stator interactions, maneuvering aircraft, the seakeeping ability of watercraft, dynamic interface problems such as aircraft interaction with carrier ships for sea-based aviation, and to many biological flows such as fish locomotion and artificial heart valves.

REFERENCES

- [1] H.M. Blackburn and R.D. Henderson. A study of two-dimensional flow past an oscillating cylinder. *Journal of Fluid Mechanics*, 385:255–286, 1999.
- [2] G.H. Koopman. The vortex wakes of vibrating cylinders at low reynolds numbers. *Journal of Fluid Mechanics*, 28(3):501–512, 1967.
- [3] R. Kurose and S. Komori. Drag and lift forces on a rotating sphere in a linear shear flow. *Journal of Fluid Mechanics*, 384:183–206, 1999.
- [4] H. Niazmand and M. Renksizbulut. Surface effects on transient three-dimensional flows around rotating spheres at moderate Reynolds numbers. *Computers & Fluids*, 32(10):1405–1433, 2003.
- [5] M. Giacobello, A. Ooi, and S. Balachandar. Wake structure of a transversely rotating sphere at moderate Reynolds numbers. *Journal of Fluid Mechanics*, 621:103–130, 2009.
- [6] K. Gharali and D.A. Johnson. Dynamic stall simulation of a pitching airfoil under unsteady freestream velocity. *Journal of Fluids and Structures*, 42:228–244, 2013.
- [7] J. Panda and K.B.M.Q. Zaman. Experimental investigation of the flow field of an oscillating airfoil and estimation of lift from wake surveys. *Journal of Fluid Mechanics*, 265:65–95, 1994.
- [8] T.F. Chan and T.P. Mathew. Domain decomposition algorithms. *Acta Numerica*, 3:61–143, 1994.
- [9] B.F. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 1996.
- [10] P.F. Fischer. An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 133(1):84–101, 1997.
- [11] W. Henshaw and D. Schwendeman. Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement. *Journal of Computational Physics*, 227(16):7469–7502, 2008.
- [12] J.L. Steger and J.A. Benek. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(1):301–320, 1987.
- [13] G. Chesshire and W.D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64, 1990.

- [14] C. Bernardi, Y. Maday, and A. Patera. Domain decomposition by the mortar element method. In *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, pages 269–286. Springer, 1993.
- [15] R.L. Meakin. Composite overset structured grids. *Handbook of Grid Generation*, pages 1–20, 1999.
- [16] C. Dawson, Q. Du, and T. Dupont. A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Mathematics of Computation*, 57(195):63–71, 1991.
- [17] J.U. Schlüter, X. Wu, S. Kim, S. Shankaran, J.J. Alonso, and H. Pitsch. A framework for coupling Reynolds-averaged with large-eddy simulations for gas turbine applications. *Journal of Fluids Engineering*, 127(4):806–815, 2005.
- [18] S. Hahn, K. Duraisamy, G. Iaccarino, S. Nagarajan, J. Sitaraman, X. Wu, J.J. Alonso, J.D. Baeder, S.K. Lele, P. Moin, and F. Schmitz. Coupled high-fidelity URANS simulation for helicopter applications. *Center for Turbulence Research, Annual Research Briefs*, 2006.
- [19] Y. Peet and S. Lele. Computational framework for coupling compressible and low mach number codes. *AIAA Journal*, 46(8):1990–2001, 2008.
- [20] H.A. Schwarz. Ueber einige abbildungsaufgaben. *Journal für die Reine und Angewandte Mathematik*, 70:105–120, 1869.
- [21] E.A. Volkov. The method of composite meshes for finite and infinite regions with piecewise smooth boundary. *Trudy Matematicheskogo Instituta im. VA Steklova*, 96:117–148, 1968.
- [22] P.-L. Lions. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. Paris, France, SIAM, Philadelphia, 1988.
- [23] Z.J. Wang. A fully conservative interface algorithm for overlapped grids. *Journal of Computational Physics*, 122(1):96–106, 1995.
- [24] W. Henshaw and D. Schwendeman. An adaptive numerical scheme for high-speed reactive flow on overlapping grids. *Journal of Computational Physics*, 191(2):420–447, 2003.
- [25] Y. Peet and P. Fischer. Heat transfer LES simulations in application to wire-wrapped fuel pins. *Proceedings 10th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, Chicago, IL*, (AIAA 2010-4318), 2010.
- [26] P.-L. Lions. On the Schwarz alternating method. III: A variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, volume 6, pages 202–223. SIAM, Philadelphia, PA, 1990.

- [27] M.H. Carpenter, J. Nordström, and D. Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148(2):341–365, 1999.
- [28] Y. Maday, C. Mavriplis, and A. Patera. *Nonconforming mortar element methods: Application to spectral discretizations*. Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, 1988.
- [29] M. Manna, A. Vacca, and M.O. Deville. Preconditioned spectral multi-domain discretization of the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 201(1):204–223, 2004.
- [30] W. Henshaw. A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids. *Journal of Computational Physics*, 113(1):13–25, 1994.
- [31] G. Desquesnes, M. Terracol, E. Manoha, and P. Sagaut. On the use of a high order overlapping grid method for coupling in CFD/CAA. *Journal of Computational Physics*, 220(1):355–382, 2006.
- [32] L.F. Pavarino and O.B. Widlund. A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions. *SIAM Journal on Numerical Analysis*, 33(4):1303–1335, 1996.
- [33] T.M. Burton and J.K. Eaton. Analysis of a fractional-step method on overset grids. *Journal of Computational Physics*, 177(2):336–364, 2002.
- [34] S.E. Sherer and J.N. Scott. High-order compact finite-difference methods on general overset grids. *Journal of Computational Physics*, 210:459–496, 2005.
- [35] P.K. Moore and J.E. Flaherty. Adaptive local overlapping grid methods for parabolic systems in two space dimensions. *Journal of Computational Physics*, 98(1):54–63, 1992.
- [36] J.C. Strikwerda and C.D. Scarbnick. A domain decomposition method for incompressible viscous flow. *SIAM Journal on Scientific Computing*, 14(1):49–67, 1993.
- [37] Y. Zang and R.L. Street. A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *International Journal for Numerical Methods in Fluids*, 20(5):341–361, 1995.
- [38] H.S. Tang, S.C. Jones, and F. Sotiropoulos. An overset-grid method for 3D unsteady incompressible flows. *Journal of Computational Physics*, 191(2):567–600, 2003.
- [39] P. Bjorstad and W. Gropp. *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 2004.

- [40] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Number CMCS-BOOK-2009-019. Oxford University Press, 1999.
- [41] A. Toselli and O. Widlund. *Domain decomposition methods: Algorithms and theory*, volume 3. Springer, 2005.
- [42] G. Starius. Composite mesh difference methods for elliptic boundary value problems. *Numerische Mathematik*, 28(2):243–258, 1977.
- [43] G. Starius. Constructing orthogonal curvilinear meshes by solving initial value problems. *Numerische Mathematik*, 28(1):25–48, 1977.
- [44] G. Starius. On composite mesh difference methods for hyperbolic differential equations. *Numerische Mathematik*, 35(3):241–255, 1980.
- [45] B. Kreiss. Construction of a curvilinear grid. *SIAM Journal on Scientific and Statistical Computing*, 4(2):270–279, 1983.
- [46] J.L. Steger, F.C. Dougherty, and J. Benek. A Chimera grid scheme. *Advances in Grid Generation, ASME Fluids Engineering Conference, Houston, TX*, 1983.
- [47] P.G. Buningt. A 3-D Chimera grid embedding technique. *Proceedings 7th AIAA Computational Physics Conference, Cincinnati, OH*, (AIAA 1985-1523), 1985.
- [48] F.C. Dougherty and J.-H. Kuan. Transonic store separation using a three-dimensional Chimera grid scheme. *27th AIAA Aerospace Sciences Meeting, Reno, NV*, (AIAA 1989-637), 1989.
- [49] P.G. Buning. Navier-Stokes simulation of the Space Shuttle launch vehicle flight transonic flow field using a large scale Chimera grid system. *12th AIAA Applied Aerodynamics Conference, Colorado Springs, CO*, (AIAA 1994-1860), 1994.
- [50] K.A. Hennesius and T.H. Pulliam. A zonal approach to solution of the Euler equations. *3rd AIAA Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, MO*, 82(AIAA 82-0969):0969, 1982.
- [51] M.M. Rai, K.A. Hennesius, and S.R. Chakravarthy. Metric-discontinuous zonal grid calculations using the Osher scheme. *Computers & Fluids*, 12(3):161–175, 1984.
- [52] K. Hennesius and M. Rai. Applications of a conservative zonal scheme to transient and geometrically complex problems. *Computers & Fluids*, 14(1):43–58, 1986.
- [53] M.M. Rai. A conservative treatment of zonal boundaries for Euler equation calculations. *Journal of Computational Physics*, 62(2):472–503, 1986.
- [54] R. Chen and T. Hagstrom. p-Adaptive Hermite methods for initial value problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(03):545–557, 2012.

- [55] D.A. Kopriva. A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations. *Journal of Computational Physics*, 143(1):125–158, 1998.
- [56] A.T. Patera. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, 54:468–488, 1984.
- [57] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-order methods for incompressible fluid flow*, volume 9. Cambridge University Press, 2002.
- [58] N.C. Prewitt, D.M. Belk, and W. Shyy. Parallel computing of overset grids for aerodynamic problems with moving objects. *Progress in Aerospace Sciences*, 36:117–172, 2000.
- [59] Z.J. Wang, V. Parthasarathy, and N. Hariharan. A fully automated Chimera methodology for multiple moving body problems. *International Journal for Numerical Methods in Fluids*, 33(7):919–938, 2000.
- [60] G. Houzeaux and R. Codina. A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(31):3343–3377, 2003.
- [61] W.D. Henshaw and D.W. Schwendeman. Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow. *Journal of Computational Physics*, 216(2):744–779, 2006.
- [62] W.D. Henshaw. A high-order accurate parallel solver for Maxwell’s equations on overlapping grids. *SIAM Journal on Scientific Computing*, 28(5):1730–1765, 2006.
- [63] C.K.W. Tam and K.A. Kurbatskii. A wavenumber based extrapolation and interpolation method for use in conjunction with high-order finite difference schemes. *Journal of Computational Physics*, 157(2):588–617, 2000.
- [64] A.K. Chaniotis and D. Poulikakos. High order interpolation and differentiation using B-splines. *Journal of Computational Physics*, 197(1):253–274, 2004.
- [65] R.E. Gordnier and M.R. Visbal. Implicit LES computation of a vortical-gust/wing interaction for transitional flow. *53rd AIAA Aerospace Sciences Meeting*, (AIAA 2015-1067), 2015.
- [66] B. Sjögren and H.C. Yee. Variable high order multiblock overlapping grid methods for mixed steady and unsteady multiscale viscous flows. *Communication in Computational Physics*, 5:730–744, 2009.
- [67] M.J. Brazell, J. Sitaraman, and D.J. Mavriplis. An overset mesh approach for 3D mixed element high order discretizations. *53rd AIAA Aerospace Sciences Meeting*, (AIAA 2015-1739), 2015.

- [68] T. Holst, S.D. Thomas, U. Kaynak, K.L. Gundy, J. Flores, and N.M. Chaderjian. Computational aspects of zonal algorithms for solving the compressible Navier-Stokes equations in three dimensions. *NASA Ames Research Center; Moffett Field, CA, United States*, (19860003828), 1985.
- [69] Ü. Kaynak, T. Holst, and B. Cantwell. Computation of transonic separated wing flows using an Euler/Navier-Stokes zonal approach. *NASA STI/Recon Technical Report N*, 88:11089, 1986.
- [70] M. Hinatsu and J.H. Ferziger. Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique. *International Journal for Numerical Methods in Fluids*, 13(8):971–997, 1991.
- [71] H.S. Tang and T. Zhou. On nonconservative algorithms for grid interfaces. *SIAM Journal on Numerical Analysis*, 37(1):173–193, 1999.
- [72] X.C. Cai. Additive Schwarz algorithms for parabolic convection diffusion equations. *Numerical Mathematics*, 60:41–61, 1991.
- [73] H. Rui. Multiplicative Schwarz methods for parabolic problems. *Applied Mathematics and Computation*, 136(2):593–610, 2003.
- [74] W.D. Henshaw. On multigrid for overlapping grids. *SIAM Journal on Scientific Computing*, 26(5):1547–1572, 2005.
- [75] Y. Peet and P.F. Fischer. Stability analysis of interface temporal discretization in grid overlapping methods. *SIAM Journal on Numerical Analysis*, 50(6):3375–3401, 2012.
- [76] C.S. Peskin. *Flow Patterns Around Heart Valves: A Digital Computer Method for Solving the Equations of Motion*. PhD thesis, Bronx, NY: Albert Einstein College of Medicine, 1972.
- [77] C.S. Peskin. The fluid dynamics of heart valves: Experimental, theoretical, and computational methods. *Annual Review of Fluid Mechanics*, 14(1):235–259, 1982.
- [78] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
- [79] S. Haeri and J.S. Shrimpton. On the application of immersed boundary, fictitious domain and body-conformal mesh methods to many particle multiphase flows. *International Journal of Multiphase Flow*, 40:38–55, 2012.
- [80] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3):283–303, 1994.
- [81] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 112(1):133–148, 1994.

- [82] R. Glowinski, T.-W. Pan, and J. Periaux. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 151(1):181–194, 1998.
- [83] M.-C. Lai and C.S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000.
- [84] B.E. Griffith, R.D. Hornung, D.M. McQueen, and C.S. Peskin. An adaptive, formally second order accurate version of the immersed boundary method. *Journal of Computational Physics*, 223(1):10–49, 2007.
- [85] K. Taira and T. Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.
- [86] A. Mark and B.G.M. van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660–6680, 2008.
- [87] S.V. Apte, M. Martin, and N.A. Patankar. A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows. *Journal of Computational Physics*, 228(8):2712–2738, 2009.
- [88] M.J. Berger and J. Olinger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984.
- [89] A.M. Roma. *A multilevel self adaptive version of the immersed boundary method*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, 1996.
- [90] A.M. Roma, C.S. Peskin, and M.J. Berger. An adaptive version of the immersed boundary method. *Journal of Computational Physics*, 153(2):509–534, 1999.
- [91] T.E. Tezduyar, J. Liou, and M. Behr. A new strategy for finite element computations involving moving boundaries and interfaces—the DSD/ST procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 1992.
- [92] T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94(3):353–371, 1992.
- [93] R. Torii, M. Oshima, T. Kobayashi, K. Takagi, and T.E. Tezduyar. Computer modeling of cardiovascular fluid–structure interactions with the deforming-spatial-domain/stabilized space–time formulation. *Computer Methods in Applied Mechanics and Engineering*, 195(13):1885–1895, 2006.

- [94] W.F. Noh. CEL: A time-dependent, two-space-dimensional, coupled Eulerian–Lagrange code. Technical report, Lawrence Radiation Lab., Univ. of California, Livermore, 1963.
- [95] R.M. Franck and R.B. Lazarus. Mixed Eulerian–Lagrangian method. *Methods in Computational Physics*, 3:47–67, 1964.
- [96] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.
- [97] W.E. Pracht. Calculating three-dimensional fluid flows at all speeds with an Eulerian–Lagrangian computing mesh. *Journal of Computational Physics*, 17(2):132–159, 1975.
- [98] J. Donéa, P. Fasoli-Stella, and S. Giuliani. Lagrangian and Eulerian finite element techniques for transient fluid–structure interaction problems. In *Structural Mechanics in Reactor Technology*. 1977.
- [99] T. Belytschko, J.M. Kennedy, and D.F. Schoeberle. Quasi-Eulerian finite element formulation for fluid–structure interaction. In *American Society of Mechanical Engineers and Canadian Society for Mechanical Engineering, Pressure Vessels and Piping Conference, Montreal, Canada*, 1978.
- [100] T. Belytschko, J.M. Kennedy, and D.F. Schoeberle. Quasi-Eulerian finite element formulation for fluid–structure interaction. *Journal of Pressure Vessel Technology*, 102(1):62–69, 1980.
- [101] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329–349, 1981.
- [102] B. Ramaswamy and M. Kawahara. Arbitrary Lagrangian–Eulerian finite element method for unsteady, convective, incompressible viscous free surface fluid flow. *International Journal for Numerical Methods in Fluids*, 7(10):1053–1075, 1987.
- [103] L.-W. Ho and A.T. Patera. A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows. *Computer Methods in Applied Mechanics and Engineering*, 80(1):355–366, 1990.
- [104] L.-W. Ho and A.T. Patera. Variational formulation of three-dimensional viscous free-surface flows: Natural imposition of surface tension boundary conditions. *International Journal for Numerical Methods in Fluids*, 13(6):691–698, 1991.
- [105] J.Y. Murthy, S.R. Mathur, and D. Choudhury. CFD simulation of flows in stirred tank reactors using a sliding mesh technique. In *Institution of Chemical Engineers Symposium Series*, volume 136, pages 341–341. Hemisphere Publishing Corporation, 1994.

- [106] A. Bakker, R.D. LaRoche, M. Wang, and R.V. Calabrese. Sliding mesh simulation of laminar flow in stirred reactors. *Chemical Engineering Research and Design*, 75(1):42–44, 1997.
- [107] K. Ng, N.J. Fentiman, K.C. Lee, and M. Yianneskis. Assessment of sliding mesh CFD predictions and LDA measurements of the flow in a tank stirred by a rushton impeller. *Chemical Engineering Research and Design*, 76(6):737–747, 1998.
- [108] J.M. Bujalski, Z. Jaworski, W. Bujalski, and A.W. Nienow. The influence of the addition position of a tracer on CFD simulated mixing times in a vessel agitated by a rushton turbine. *Chemical Engineering Research and Design*, 80(8):824–831, 2002.
- [109] C.A. Rivera, M. Heniche, F. Bertrand, R. Glowinski, and P.A. Tanguy. A parallel finite element sliding mesh technique for the simulation of viscous flows in agitated tanks. *International Journal for Numerical Methods in Fluids*, 69(3):653–670, 2012.
- [110] Y.M. Park and O.J. Kwon. Simulation of unsteady rotor flow field using unstructured adaptive sliding meshes. *Journal of the American Helicopter Society*, 49(4):391–400, 2004.
- [111] G. Barakos, R. Steijl, K. Badcock, and A. Brocklehurst. Development of CFD capability for full helicopter engineering analysis. In *31st European Rotorcraft Forum, Florence, Italy, Sept*, 2005.
- [112] H.J. Nam, Y.M. Park, and O.J. Kwon. Simulation of unsteady rotor-fuselage aerodynamic interaction using unstructured adaptive meshes. *Journal of the American Helicopter Society*, 51(2):141–149, 2006.
- [113] R. Steijl and G. Barakos. Sliding mesh algorithm for CFD analysis of helicopter rotor-fuselage aerodynamics. *International Journal for Numerical Methods in Fluids*, 58(5):527–549, 2008.
- [114] A. Buffa, Y. Maday, and F. Rapetti. A sliding mesh-mortar method for a two dimensional eddy currents model of electric engines. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 35(2):191–228, 2001.
- [115] R.L. Meakin. Moving body overset grid methods for complete aircraft tiltrotor simulations. *AIAA Paper*, 3350:1993, 1993.
- [116] K. Pahlke and B. Van Der Wall. Calculation of multibladed rotors in high-speed forward flight with weak fluid-structure-coupling. In *27th European Rotorcraft Forum*, 2001.
- [117] T. Renaud, C. Benoit, J.C. Boniface, and P. Gardarein. Navier-Stokes computations of a complete helicopter configuration accounting for main and tail rotors effects. *Office National d Etudes et de Recherches Aerospatiales Onera-Publications-TP*, (142), 2003.

- [118] H. Pomin and S. Wagner. Aeroelastic analysis of helicopter rotor blades on deformable chimera grids. *Journal of Aircraft*, 41(3):577–584, 2004.
- [119] M. Potsdam, H. Yeo, and W. Johnson. Rotor airloads prediction using loose aerodynamic/structural coupling. *Journal of Aircraft*, 43(3):732–742, 2006.
- [120] D. Chandar, J. Sitaraman, and D. Mavriplis. Dynamic overset grid computations for CFD applications on graphics processing units. In *Proceedings of the Seventh International Conference on Computational Fluid Dynamics*, 2012.
- [121] J.W. Banks, W.D. Henshaw, and D.W. Schwendeman. Deforming composite grids for solving fluid structure problems. *Journal of Computational Physics*, 231(9):3518–3547, 2012.
- [122] C. Kiris, D. Kwak, S. Rogers, and I.D. Chang. Computational approach for probing the flow through artificial heart devices. *Journal of Biomechanical Engineering*, 119(4):452–460, 1997.
- [123] L. Ge, H.-L. Leo, F. Sotiropoulos, and A.P. Yoganathan. Flow in a mechanical bileaflet heart valve at laminar and near-peak systole flow rates: CFD simulations and experiments. *Journal of Biomechanical Engineering*, 127(5):782–797, 2005.
- [124] L.E. Lijewski and N.E. Suhs. Chimera-eagle store separation. In *1992 AIAA Atmospheric Flight Mechanics Conference*, 1992.
- [125] R.L. Meakin. Computations of the unsteady flow about a generic wing/pylon/finned-store configuration. *AIAA Paper*, 4568:1992, 1992.
- [126] N.C. Prewitt, D.M. Belk, and W. Shyy. Parallel computing of overset grids for aerodynamic problems with moving objects. *Progress in Aerospace Sciences*, 36(2):117–172, 2000.
- [127] R.W. Noack, D.A. Boger, R.F. Kunz, and P.M. Carrica. Suggar++: An improved general overset grid assembly capability. *Proceedings of the 47th AIAA Aerospace Science and Exhibit*, pages 22–25, 2009.
- [128] P.M. Carrica, R.V. Wilson, R.W. Noack, and F. Stern. Ship motions using single-phase level set with dynamic overset grids. *Computers & Fluids*, 36(9):1415–1433, 2007.
- [129] P.M. Carrica, J. Huang, R. Noack, D. Kaushik, B. Smith, and F. Stern. Large-scale DES computations of the forward speed diffraction and pitch and heave problems for a surface combatant. *Computers & Fluids*, 39(7):1095–1111, 2010.
- [130] M.J. Brazell, J. Sitaraman, and D.J. Mavriplis. An overset mesh approach for 3D mixed element high order discretizations. *53rd AIAA Aerospace Sciences Meeting*, (AIAA 2015-1739), 2015.

- [131] J. Donea, A. Huerta, J-Ph. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian–Eulerian methods. *Encyclopedia of Computational Mechanics*, 2004.
- [132] C. Truesdell and R. Toupin. *The classical field theories*. Springer, 1960.
- [133] L.W. Ho. *A Legendre spectral element method for simulation of incompressible unsteady viscous free-surface flows*. PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 1989.
- [134] G.E. Karniadakis, M. Israeli, and S.A. Orszag. High-order splitting methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 97(2):414–443, 1991.
- [135] B.E. Merrill, Y.T. Peet, and P.F. Fischer. A spectrally accurate method for overlapping grid solution of incompressible Navier–Stokes equations. *Journal of Computational Physics (in review)*, 2015.
- [136] P. Fischer, J. Lottes, D. Pointer, and A. Siegel. Petascale algorithms for reactor hydrodynamics. In *Journal of Physics: Conference Series*, volume 125, page 012076. IOP Publishing, 2008.
- [137] O. Walsh. Eddy solutions of the Navier–Stokes equations. In *The Navier-Stokes Equations II - Theory and Numerical Methods*, pages 306–309. Springer, 1992.
- [138] C. Schneidesch and M. Deville. Chebyshev collocation method and multi-domain decomposition for Navier–Stokes equations in complex curved geometries. *Journal of Computational Physics*, 106:234–257, 1993.
- [139] Y. Peet and P. F. Fischer. Legendre spectral element method with nearly incompressible materials. *European Journal of Mechanics - A/Solids*, 44:91–103, 2013.
- [140] T. Colonius, S.K. Lele, and P. Moin. The free compressible viscous vortex. *Journal of Fluid Mechanics*, 230:45–73, 1991.
- [141] G.I. Taylor. On the dissipation of eddies. *The Scientific Papers of Sir Geoffrey Ingram Taylor*, 2:96–101, 1918.
- [142] J.G.M. Eggels, F. Unger, M.H. Weiss, J. Westerweel, R.J. Adrian, R. Friedrich, and F.T.M. Nieuwstadt. Fully developed turbulent pipe flow: A comparison between direct numerical simulation and experiment. *Journal of Fluid Mechanics*, 268:175–210, 1994.
- [143] G. Magnus. On the deflection of a projectile. *Abhandlungen der Akademie der Wissenschaften, Berlin, Germany*, 1852.
- [144] R.E.D. Bishop and A.Y. Hassan. The lift and drag forces on a circular cylinder oscillating in a flowing fluid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 277(1368):51–75, 1964.

- [145] C.H.K. Williamson and A. Roshko. Vortex formation in the wake of an oscillating cylinder. *Journal of Fluids and Structures*, 2(4):355–381, 1988.
- [146] C.H.K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.
- [147] H.S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174(1):345 – 380, 2001.
- [148] J.W. Maccoll. Aerodynamics of a spinning sphere. *Journal of the Royal Aeronautical Society*, 28:777–798, 1928.
- [149] H.M. Barkla and L.J. Auchterlonie. The Magnus or Robins effect on rotating spheres. *Journal of Fluid Mechanics*, 47(03):437–447, 1971.
- [150] Y. Tsuji, Y. Morikawa, and O. Mizuno. Experimental measurement of the Magnus force on a rotating sphere at low Reynolds numbers. *Journal of fluids engineering*, 107(4):484–488, 1985.
- [151] R.G. Watts and R. Ferrer. The lateral force on a spinning sphere: Aerodynamics of a curveball. *Am. J. Phys*, 55(1):40–44, 1987.
- [152] B. Oesterle and T.B. Dinh. Experiments on the lift of a spinning sphere in a range of intermediate Reynolds numbers. *Experiments in Fluids*, 25(1):16–22, 1998.
- [153] D. Kim and H. Choi. Laminar flow past a sphere rotating in the streamwise direction. *Journal of Fluid Mechanics*, 461:365–386, 2002.
- [154] T.A. Johnson and V.C. Patel. Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics*, 378:19–70, 1999.
- [155] J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.
- [156] N.D. Ham and M.S. Garelick. Dynamic stall considerations in helicopter rotors. *Journal of the American Helicopter Society*, 13(2):49–55, 1968.
- [157] G.A. Pierce, D.L. Kunz, and J.B. Malone. The effect of varying freestream velocity on airfoil dynamic stall characteristics. *Journal of the American Helicopter Society*, 23(2):27–33, 1978.
- [158] J.W. Larsen, S.R.K. Nielsen, and S. Krenk. Dynamic stall model for wind turbine airfoils. *Journal of Fluids and Structures*, 23(7):959–982, 2007.
- [159] W.G. Brady and G.R. Ludwig. Unsteady stall of axial flow compressors. Technical report, DTIC Document, 1963.
- [160] Z.-W. Shi and X. Ming. Effects of unsteady freestream on aerodynamic characteristics of a pitching delta wing. *Journal of Aircraft*, 45(6):2182–2185, 2008.

- [161] Z. Shi and X. Ming. Experimental investigation on a pitching motion delta wing in unsteady free stream. *Modern Physics Letters B*, 23(03):409–412, 2009.
- [162] L.W. Carr, K.W. McAlister, and W.J. McCroskey. Analysis of the development of dynamic stall based on oscillating airfoil experiments. Technical report, NASA Technichal Note D-8382, 1977.
- [163] W.J. McCroskey. The phenomenon of dynamic stall. Technical report, DTIC Document, 1981.
- [164] L.E. Ericsson and J.P. Reding. Fluid mechanics of dynamic stall part i. unsteady flow concepts. *Journal of fluids and structures*, 2(1):1–33, 1988.
- [165] P.G. Choudhuri and D.D. Knight. Effects of compressibility, pitch rate, and reynolds number on unsteady incipient leading-edge boundary layer separation over a pitching airfoil. *Journal of Fluid Mechanics*, 308:195–217, 1996.
- [166] M.R. Visbal. Numerical investigation of deep dynamic stall of a plunging airfoil. *AIAA journal*, 49(10):2152–2170, 2011.
- [167] D.J. Garmann and M.R. Visbal. Numerical investigation of transitional flow over a rapidly pitching plate. *Physics of Fluids (1994-present)*, 23(9):094106, 2011.
- [168] K. Gharali and D.A. Johnson. Effects of nonuniform incident velocity on a dynamic wind turbine airfoil. *Wind Energy*, 2014.
- [169] L.W. Carr. Dynamic stall progress in analysis and prediction. *Proceedings AIAA 12th Atmospheric Mechanics Conference, Snowmass, CO*, (AIAA paper 1985-1769), 1985.
- [170] S.B.R. Kottapalli and G.A. Pierce. Drag on an oscillating airfoil in a fluctuating free stream. *Journal of Fluids Engineering*, 101(3):391–399, 1979.
- [171] R.N. Conger and B.R. Ramaprian. Pressure measurements on a pitching airfoil in a water channel. *AIAA journal*, 32(1):108–115, 1994.
- [172] A. Laneville and P. Vittecoq. Effect of turbulence on dynamic stall. In *Wind Turbine Aerodynamics Seminar*. Sandia National Labs, 1985.
- [173] J.M. Chen and C.-C. Choa. Freestream disturbance effects on an airfoil pitching at constant rate. *Journal of aircraft*, 36(3):507–514, 1999.
- [174] T. Lee and P. Gerontakos. Investigation of flow over an oscillating airfoil. *Journal of Fluid Mechanics*, 512:313–341, 2004.
- [175] M.R. Visbal. Analysis of the onset of dynamic stall using high-fidelity large-eddy simulations. In *AIAA SciTech 52nd Aerospace Sciences Meeting National Harbor, Maryland*, number AIAA 2010-4318, 2014.

- [176] S. Wang, D.B. Ingham, L. Ma, M. Pourkashanian, and Z. Tao. Numerical investigations on dynamic stall of low Reynolds number flow around oscillating airfoils. *Computers & Fluids*, 39(9):1529–1541, 2010.
- [177] S. Wang, Z. Tao, L. Ma, D. Ingham, and M. Pourkashanian. Numerical investigations on dynamic stall associated with low Reynolds number flows over airfoils. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 5, pages V5–308. IEEE, 2010.
- [178] R.L. Bisplinghoff, H. Ashley, and R.L. Halfman. *Aeroelasticity*. Courier Corporation, 2013.
- [179] A.G. Kravchenko and P. Moin. Numerical studies of flow over a circular cylinder at $Re_D = 3900$. *Physics of Fluids*, 12(2):403–417, 2000.
- [180] P. Parnaudeau, J. Carlier, D. Heitz, and E. Lamballais. Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. *Physics of Fluids*, 20(8):085101, 2008.
- [181] X. Ma, G.-S. Karamanos, and G.E. Karniadakis. Dynamics and low-dimensionality of a turbulent near wake. *Journal of Fluid Mechanics*, 410:29–65, 2000.
- [182] L. Ong and J. Wallace. The velocity field of the turbulent very near wake of a circular cylinder. *Experiments in fluids*, 20(6):441–453, 1996.
- [183] P. Beaudan and P. Moin. Numerical experiments on the flow past a circular cylinder at sub-critical Reynolds number. Technical report, DTIC Document, 1994.
- [184] R. Mittal and P. Moin. Suitability of upwind-biased finite difference schemes for large-eddy simulation of turbulent flows. *AIAA journal*, 35(8):1415–1417, 1997.
- [185] G.S. Cardell. *Flow past a circular cylinder with a permeable wake splitter plate*. PhD thesis, California Institute of Technology, 1993.
- [186] K.K.Y. Tsang, R.M.C. So, R.C.K. Leung, and X.Q. Wang. Dynamic stall behavior from unsteady force measurements. *Journal of Fluids and Structures*, 24(1):129–150, 2008.

APPENDIX A
PRESSURE SOLUTION FOR WALSH'S EDDIES

Convecting frame of reference Eddy solutions in a convecting frame of reference can be found using the technique proposed by Walsh [137] and discussed in section 3.1. For the present test case, we find these solutions from the initial conditions given by the stream function

$$\tilde{\psi}(\tilde{x}, \tilde{y}) = -\frac{1}{5}\cos[5\tilde{x}] + \frac{1}{4}\sin[3\tilde{x}]\sin[4\tilde{y}] - \frac{1}{5}\sin[5\tilde{y}], \quad (\text{A.1})$$

leading to the following exact solution for \mathbf{u} in a convecting frame of reference:

$$\begin{aligned} \tilde{u}(\tilde{x}, \tilde{y}, t) &= e^{-25t\nu}(-\cos[5\tilde{y}] + \cos[4\tilde{y}]\sin[3\tilde{x}]), \\ \tilde{v}(\tilde{x}, \tilde{y}, t) &= e^{-25t\nu}(-\sin[5\tilde{x}] - \frac{3}{4}\cos[3\tilde{x}]\sin[4\tilde{y}]), \end{aligned} \quad (\text{A.2})$$

As in section 3.1, by tilde we denote the variables in a convecting frame of reference.

The solution for pressure is found from the relation $\nabla\tilde{p} = -\tilde{\mathbf{u}} \cdot \nabla\tilde{\mathbf{u}}$, which, in the present case, leads to two partial differential equations for pressure:

$$\frac{\partial\tilde{p}}{\partial\tilde{x}} = -\tilde{u}\frac{\partial\tilde{u}}{\partial\tilde{x}} - \tilde{v}\frac{\partial\tilde{u}}{\partial\tilde{y}} \quad (\text{A.3})$$

$$\frac{\partial\tilde{p}}{\partial\tilde{y}} = -\tilde{u}\frac{\partial\tilde{v}}{\partial\tilde{x}} - \tilde{v}\frac{\partial\tilde{v}}{\partial\tilde{y}} \quad (\text{A.4})$$

Solving this set of equations by integration gives the solution for the pressure of non-convecting eddies

$$\begin{aligned} \tilde{p}(\tilde{x}, \tilde{y}, t) = & \frac{1}{64}e^{-50t\nu}(-16 + 16\cos[6\tilde{x}] + 8\cos[8\tilde{x} - 4\tilde{y}] \\ & 32\cos[2(\tilde{x} - 2\tilde{y})] + 9\cos[8\tilde{y}] - 8\cos[4(2\tilde{x} + \tilde{y})] \\ & 32\cos[2(\tilde{x} + 2\tilde{y})] - 4\sin[3(\tilde{x} - 3\tilde{y})] + 32\sin[5(\tilde{x} - \tilde{y})] \\ & + 36\sin[3\tilde{x} - \tilde{y}] - 32\sin[5(\tilde{x} + \tilde{y})] + 36\sin[3\tilde{x} + \tilde{y}] \\ & - 4\sin[3(\tilde{x} + 3\tilde{y})]) \end{aligned} \quad (\text{A.5})$$

Stationary frame of reference To convert the above solutions to the stationary frame of reference, we need to use the coordinate transformation

$$\begin{aligned} x &= \tilde{x} + u_0t \\ y &= \tilde{y} + v_0t, \end{aligned} \quad (\text{A.6})$$

where (u_0, v_0) is the constant velocity of convection. That leads to the following solution for velocity

$$\begin{aligned} u(x, y, t) &= e^{-25t\nu}(-\cos[5(y - v_0t)] + \cos[4(y - v_0t)]\sin[3(x - u_0t)]) + u_0, \\ v(x, y, t) &= e^{-25t\nu}(-\sin[5(x - u_0t)] - \frac{3}{4}\cos[3(x - u_0t)]\sin[4(y - v_0t)]) + v_0, \end{aligned} \quad (\text{A.7})$$

while solution for pressure is obtained analogously as

$$p(x, y, t) = \tilde{p}(x - u_0t, y - v_0t, t) \quad (\text{A.8})$$

and not spelled out here due to a rather cumbersome final expression.