5-2017

Application of Strand-Cartesian Interfaced Solver on Flows Around Various Geometries

Yushi Yanagita

APPLICATION OF STRAND-CARTESIAN INTERFACED SOLVER ON FLOWS AROUND VARIOUS GEOMETRIES

by

Yushi Yanagita

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

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Douglas Hunsaker, Ph.D. Major Professor	Barton Smith, Ph.D. Committee Member

Nick Roberts, Ph.D.
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2017

Abstract

Application of Strand-Cartesian Interfaced Solver on Flows around Various Geometries

by

Yushi Yanagita, Master of Science

Utah State University, 2017

Major Professor: Douglas Hunsaker, Ph.D.
Department: Mechanical and Aerospace Engineering

This work examines the application of a high-order numerical method to strand-based grids to solve the Navier-Stokes equations. Coined "Flux Correction", this method eliminates error terms in the fluxes of traditional second-order finite volume Galerkin methods. Flux Correction is first examined for applications to the Reynolds-Averaged Navier-Stokes equations to compute turbulent flows on a strictly strand-based domain. Flow over three geometries are examined to demonstrate the method's capabilities: a three-dimensional bump, an infinite wing, and a hemisphere-cylinder configuration. Comparison to results obtained from established codes show that the turbulent Flux Correction scheme accurately predicts flow properties such as pressure, velocity profiles, shock location and strength. However, it can be seen that an overset Cartesian solver is necessary to more accurately capture certain flow properties in the wake region.

The Strand-Cartesian Interface Manager(SCIM) uses a combination of second-order trilinear interpolation and mixed-order Lagrange interpolation to establish domain connectivity between the overset grids. Verification of the high-order SCIM code are conducted through the method of manufactured solutions. Steady and unsteady flow around a sphere are used to validate the SCIM library.

The method is found to be have a combined order of accuracy of approximately 2.5, and has improved accuracy for steady cases. However, for unsteady cases the method fails to accurately predict the time-dependent flow field.

(58 pages)

Public Abstract

Application of Strand-Cartesian Interfaced Solver on Flows around Various Geometries

by

Yushi Yanagita, Master of Science

Utah State University, 2017

Major Professor: Douglas Hunsaker, Ph.D.
Department: Mechanical and Aerospace Engineering

This work examines the application of a high-order numerical method to strand-based grids to solve the Navier-Stokes equations, which govern fluid motion. The method is intended to provide more accurate predictions of fluid flow without significantly increasing the cost of computation. Flux Correction is first applied to turbulent flows on a strand-based domain. Flow over three geometries are examined to demonstrate the method's capabilities: a three-dimensional bump, an infinite wing, and a hemisphere-cylinder configuration. Comparison to results obtained from established codes show that the turbulent flux correction scheme accurately predicts flow properties such as pressure, velocity profiles, shock location and strength. However, it can be seen that certain regions in the flow are lacking the ability to capture certain flow properties.

The Strand-Cartesian Interface Manager resolves these issues using an overlapping rectangular grid. Data is interpolated between the overlapping grids to establish communication between the two domains. The code is checked by solving a known problem, and comparing the amount of error found with different levels of grid refinement. Time-dependent and time-independent flows around a sphere are used to confirm the accuracy of the SCIM library. The method is found to have improved accuracy for time-independent cases, but for time-dependent cases the method fails to accurately predict the unsteady flow field.

To my parents, whose never ending support has gotten me here.

Acknowledgments

I would like to thank Dr. Aaron Katz, for bringing me on board this project, and for the support and guidance rendered over the year and a half I spent under his mentorship.

A special thank you to Dr. Douglas Hunsaker, my major professor. Without him, our research would have ended along with Dr. Katz's departure from the University.

A warm thank you to each of my coworkers who helped guide me along as I came to explore the new found world of high-order numerical methods. Especially to Dalon Work, who patiently answered the hundreds, if not thousands, of questions ranging in the fields of fluids and programming all the way to simple linux operations.

Finally, I wish to thank my loving parents who have supported me in every endeavour I have put my wandering mind to. For the thirst for knowledge that they have instilled in me since my childhood; for the patience they had in raising a rowdy teenage boy, who swore he had everything figured out; for their continued support of my educational goals. Without them, I would not be where I am today, or the man I am today.

Yushi Yanagita

Contents

List of Tables

List of Figures

Acronyms

CFD	Computation Fluid Dynamics
FC	Flux Correction
FR	Flux Reconstruction
FV	Finite Volume
LES	Large Eddy Simulation
LHS	Left-Hand Side
MMS	Method of Manufactured Solutions
RHS	Right-Hand Side
RMS	Root Mean Square
SCIM	Strand-Cartesian Interface Manager

Chapter 1

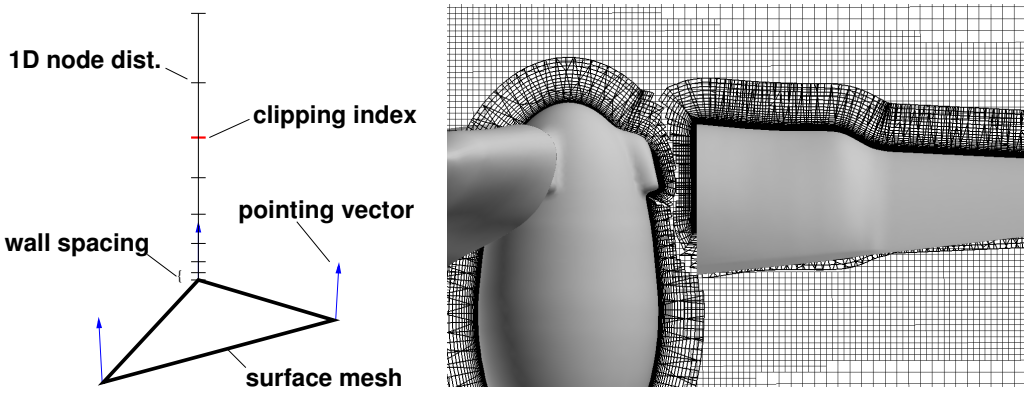
Introduction

In recent years, a number of high-order computational fluid dynamics (CFD) methods have been investigated due to their potential to generate high-fidelity solutions at a lower computational cost. However, second-order schemes are still prevalent in industry. While the use of traditional second-order schemes is adequate for many applications, there are other instances, such as vortex-dominated flows, where considerable numerical dissipation contaminates the accuracy of the solution. High-order methods, coupled with recent advances in turbulence modeling, may enable accurate results without the need for excessive grid refinement or computational cost. Additionally, automated viscous mesh generation and maintaining computational efficiency are among the current research challenges in CFD.

The strand-grid approach has potential to alleviate many of these difficulties. Strand grids allow for fully automatic volume grid generation while enhancing scalability and the possibility for high-order accuracy. The strand approach automatically creates a prismatic mesh along "strands" emanating from pointing vectors determined from a surface tessellation in order to resolve viscous boundary layers and other near-body effects, as shown in Figure 1.1(a). Away from solid bodies, adaptive Cartesian grids resolve vortex shedding and wake features with efficient high-order algorithms, shown in Figure 1.1(b).

1.1 Purpose of Report

The primary goals of this work are two-fold. The first goal is to demonstrate the capabilities of the high-order strand-based code in capturing turbulent flows over various bodies. Three test cases are employed to demonstrate the capabilities of the methodology: a three dimensional bump, an infinite wing, and a hemisphere-cylinder configuration. All configurations are taken from the NASA-Langley Turbulence Modelling Resource website [2].



(a) strand grid components (b) strand/Cartesian grid for TRAM rotor

Fig. 1.1: Strand grid elements and example strand/Cartesian grid system for the TRAM rotor

All configurations are tested in the subsonic regime. Additionally, transonic flow for the hemisphere-cylinder and infinite wing are examined to demonstrate the capability of the method in capturing turbulent shock interactions. Each case will be compared to results obtained via established methods where available.

The second goal is to show capabilities of the Strand-Cartesian Interface Manager(SCIM). Results obtained through the SCIM library for simple cases will be compared to established analytic and experimental results. Results for the method of manufactured solutions(MMS) are presented to establish the order of accuracy of the SCIM code. Steady and unsteady flows over a sphere are then computed and compared to experimental results.

1.2 Report Outline

The remainder of the report is organized as follows: Chapter 2 covers topics necessary to understand the work done in this report, including details on the computational strand domain, Flux Correction scheme, and compressible limiter employed in this work. Chapter 3 provides a discussion of the results obtained for the three turbulent cases, including subsonic flow over a bump-in-channel geometry, subsonic and transonic flow over an infinite NACA 0012 wing, and subsonic and transonic flow over a hemisphere-cylinder configuration. Chapter 4 details the domain connectivity in SCIM , focusing on the interpolation schemes

implemented. Chapter 5 provides a comparison of results obtained by the SCIM code with established analytic and experimental results, as well as a discussion of the benefits of the overset method. Chapter 6 draws conclusions on this work based on the results presented and suggests future research directions.

Chapter 2

Background

This chapter briefly explains the computational structure used in Flux Correction, as well as the Flux Correction scheme developed by Katz and Work [3], and the compressible limiter used for transonic computations.

2.1 Computational Strand Domain

The governing Navier-Stokes equations are solved on grids comprised of unstructured triangular elements extruded along strands which extend as straight lines away from solid bodies, producing "stacks" of prismatic triangular cells. Each of these stacks, existing in the physical domain, are mapped to a computational space. This mapping is shown in Fig. 2.1. For use in high-order algorithms, the triangular elements constituting each stack must be sub-divided into equally spaced triangles in the r-s plane. In this work, fourth-order elements are used, whereas only quadratic elements are shown in Fig. 2.1. In the computational domain, the unevenly spaced nodes along each strand in the physical domain become evenly spaced in the η-direction. The strand node spacing in the computational domain $\Delta\eta$ is given by

$$\Delta\eta = \frac{1}{N-1} \tag{2.1}$$

where N is the number of nodes along each strand.

2.2 Flux Correction

Flux Correction performs a high-order finite-volume flux balance in the r-s plane and uses high-order finite differences in the strand direction based on summation-by-parts (SBP) operators [4,5]. Two-dimensional median-dual control volumes, which allows the application of the Flux Correction algorithms, are built around each node that comprises the high-order

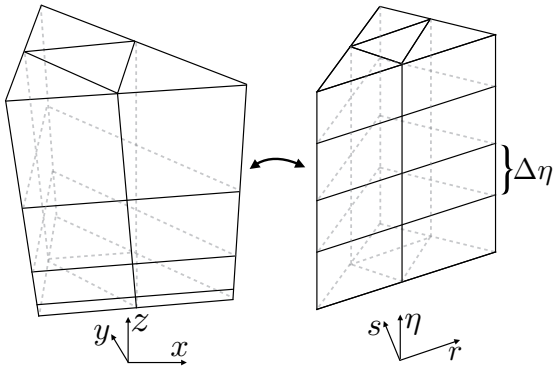
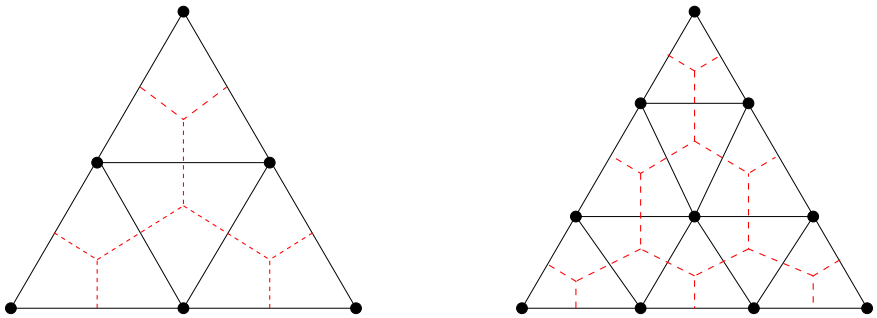


Fig. 2.1: Mapping of strand stack from physical space to computational space.



(a) Quadratic elements.

(b) Cubic elements.

Fig. 2.2: Surface element sub-division used for gradient reconstruction.

surface elements in the r-s plane. Fig. 2.2 shows median-dual control volumes created on quadratic and cubic elements in red.

The governing equations of fluid motion derived by taking into account conservation of mass, momentum, and energy, known as the Navier-Stokes equations, can be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S, \tag{2.2}$$

where the vectors of conserved variables, Q, inviscid fluxes, $F_j = (F, G, H)$, and viscous fluxes, $F_j^v = (F^v, G^v, H^v)$, are defined as

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho\tilde{\nu} \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ \rho h u_j \\ \rho\tilde{\nu} u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ \frac{\eta}{\sigma}\frac{\partial\tilde{\nu}}{\partial x_j} \end{pmatrix}. \tag{2.3}$$

When mapped to the computational space, Eq. 2.2 becomes

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} + \frac{\partial \hat{H}}{\partial \eta} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} - \frac{\partial \hat{H}^v}{\partial \eta} = \hat{S}, \tag{2.4}$$

$$\hat{Q} \equiv JQ, \quad \hat{S} \equiv JS,$$

$$\hat{F} \equiv J\left(r_x F + r_y G + r_z H\right), \quad \hat{F}^v \equiv J\left(r_x F^v + r_y G^v + r_z H^v\right),$$

$$\hat{G} \equiv J\left(s_x F + s_y G + s_z H\right), \quad \hat{G}^v \equiv J\left(s_x F^v + s_y G^v + s_z H^v\right),$$

$$\hat{H} \equiv J\left(\eta_x F + \eta_y G + \eta_z H\right), \quad \hat{H}^v \equiv J\left(\eta_x F^v + \eta_y G^v + \eta_z H^v\right),$$

$$\begin{pmatrix} r_x & s_x & \eta_x \\ r_y & s_y & \eta_y \\ r_z & s_z & \eta_z \end{pmatrix} = \frac{1}{J}\begin{pmatrix} y_s z_\eta - z_s y_\eta & z_r y_\eta - y_r z_\eta & y_r z_s - z_r y_s \\ z_s x_\eta - x_s z_\eta & x_r z_\eta - z_r x_\eta & z_r x_s - x_r z_s \\ x_s y_\eta - y_s x_\eta & y_r x_\eta - x_r y_\eta & x_r y_s - y_r x_s \end{pmatrix},$$

$$J = x_\eta\left(y_r z_s - z_r y_s\right) + y_\eta\left(z_r x_s - x_r z_s\right) + z_\eta\left(x_r y_s - y_r x_s\right).$$

Here, J is the Jacobian of the transformation matrix; and $\hat{F}_j$ and $\hat{F}_j^v$ are the transformed inviscid and viscous fluxes respectively. All subscripts denote partial differentiation.

An important aspect of the method takes advantage of the even spacing found in the strand-wise direction in the computational domain. By treating the η-derivatives as part of the source terms found on the right-hand side of the governing equations, each layer (r-s plane) of the strand stack can be computed individually whilst preserving the accuracy of the Flux Correction procedure. By grouping the η-derivatives and the physical time derivative with the right-hand source term, the new governing equations take the form

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} = \tilde{S}, \tag{2.5}$$

$$\tilde{S} \equiv \hat{S} - \frac{\partial \hat{Q}}{\partial t} - \frac{\partial \hat{H}}{\partial \eta} + \frac{\partial \hat{H}^v}{\partial \eta}.$$

The pseudo-time derivative is added on the left-hand side to facilitate the semi-implicit time marching solution.

Though the strand stack layers are computed individually, they are coupled together through the source term $\tilde{S}$, which contains the η-derivative terms of the flux. Computation of the η-derivatives is handled by SBP operators along with penalty-based boundary conditions. As long as these terms are computed to at least second-order truncation error, the corrected flux balance in the r-s plane will retain high-order properties.

Equation 2.5 calls for discretization suitable for unstructured grids with source terms. This work uses the scheme developed by Pincock and Katz [6], which results in third-order solution error for inviscid fluxes, and fourth-order error for viscous fluxes.

Flux Correction introduces a new method for computing numerical fluxes to be used in finite volume schemes. From work done by Katz and Sankaran [7], the inviscid numerical flux between two nodes is given as

$$\hat{\mathcal{F}}_{ab,j} = \frac{1}{2}\left(\hat{\mathcal{F}}_R + \hat{\mathcal{F}}_L\right) - \frac{1}{2}\left|\hat{\mathcal{A}}(Q_R, Q_L)\right|(Q_R - Q_L) \tag{2.6}$$

where $\hat{\mathcal{F}}_L$ and $\hat{\mathcal{F}}_R$ represent reconstructed fluxes. These terms are unique to Flux Correction and are computed using high-order gradient information. The formulation for the reconstructed fluxes can be seen in equation 2.7.

$$\hat{\mathcal{F}}_L = \hat{\mathcal{F}}_{a,j} + \frac{1}{2}\Delta\mathbf{r}_{ab}^T \left(\nabla_{rs}^h \hat{\mathcal{F}}\right)_{a,j}, \quad \hat{\mathcal{F}}_R = \hat{\mathcal{F}}_{b,j} - \frac{1}{2}\Delta\mathbf{r}_{ab}^T \left(\nabla_{rs}^h \hat{\mathcal{F}}\right)_{b,j}, \tag{2.7}$$

$$\Delta\mathbf{r}_{ab}^T = (r_b - r_a, s_b - s_a)$$

In this representation, ∇_{rs}^h denotes an estimation of the gradient in the r-s plane. For Flux Correction to produce the desired high-order accuracy, these gradients must be computed to a minimum second-order accuracy. For more detail regarding Flux Correction on three-dimensional grids, see the article in the Journal of Scientific Computing by Tong and the author [1].

2.3 Compressible Limiter

Discontinuous flows continue to present challenges for high-order computational methods in the form of Gibbs oscillations. This work incorporates a traditional finite volume limiter to eliminate Gibbs oscillations by introducing first-order artificial dissipation in regions where discontinuities occur in the flow . The subtraction term in equation 2.6 uses the left and right states, Q_R and Q_L, to compute the artificial dissipation term in the fluxes. For smooth regions these states are reconstructed, similar to the left and right flux terms, using high-order gradient terms. For regions with discontinuities in the flow, a limiter is used to neglect the reconstruction of the left and right states so that they take on the values of the left and right nodes. The left and right state reconstruction equation can be written with the limiter as

$$Q_L = Q_{a,j} + \frac{\phi_{ab}}{2}\Delta\mathbf{r}_{ab}^T \nabla_{rs}^h Q_{a,j}, \quad Q_R = Q_{b,j} - \frac{\phi_{ab}}{2}\Delta\mathbf{r}_{ab}^T \nabla_{rs}^h Q_{b,j}, \tag{2.8}$$

where ϕ_{ab} is the limiter between nodes a and b. The limiter is computed in a manner similar to Jameson's symmetric limited positive (SLIP) scheme, shown in equation 2.9 [8, 9].

$$\phi_{ab} = 1 - \left| \frac{u - v}{\max(|u| + |v|, \epsilon)} \right|^3, \quad u = 2\Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{b,j} - \Delta Q_{ab}, \tag{2.9}$$

$$v = 2\Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{a,j} - \Delta Q_{ab}, \quad \Delta Q_{ab} = Q_b - Q_a,$$

The r-s gradients in the limiter are computed using the same method within the reconstruction of the left and right states, thus requiring no additional computation or MPI communication. In the limiter computation, ϵ is used to avoid any incorrect application of the limiter, such as in freestream flow where $|u| + |v| = 0$. For more detail on the compressible limiter, see the conference paper written by Tong and the author, currently pending publication in the International Journal of Computational Fluid Dynamics [10].

Chapter 3

Validation of Flux Correction

The ability of the strand code to capture turbulent flows over various geometries is investigated in this section. Three fundamental cases are inspected: flow over a 3D bump in a channel, flow over an infinite NACA 0012 wing, and flow over a hemisphere-cylinder body. The hemisphere-cylinder and infinite wing cases are examined for both subsonic and transonic flow conditions. The subsonic cases are compared extensively to results available in the NASA-Langley Turbulence Modeling Resource [2]. When available, results from two independent compressible CFD codes are used for comparison with the strand-based code, as well as experimental data whenever possible. Results are also compared to those obtained without the use of flux correction. In these cases the results obtained with the flux correction scheme are denoted "Strand FC," and the second-order results without flux correction are denoted "Strand."

3.1 Three-Dimensional Bump-in-Channel

The first case examined is subsonic flow over a bump-in-channel. A full description of the case can be found in the NASA Turbulence Modeling Resource [2], key features are described here. The lower wall consists of a curved viscous-wall bump extending from $x = 0$ to 1.5 at the two sides of the computational domain, which extends from $y = 0$ to $y = -1$. The leading and trailing edges of the viscous-wall bump follow a sinusoidal shape, as can be seen in figure 3.1. The body reference length is 1.5 units, with a maximum bump height

of $z = 0.05$. The "two-dimensional" bump geometry along the $y = 0$ plane is defined by

$$z(x) = \begin{cases} 0 & 0 < x < 0.3 \\ 0.05 \sin^4(\frac{\pi x}{0.9} - \frac{\pi}{3}) & 0.3 \leq x \leq 1.2 \\ 0 & 1.2 < x \end{cases}, \tag{3.1}$$

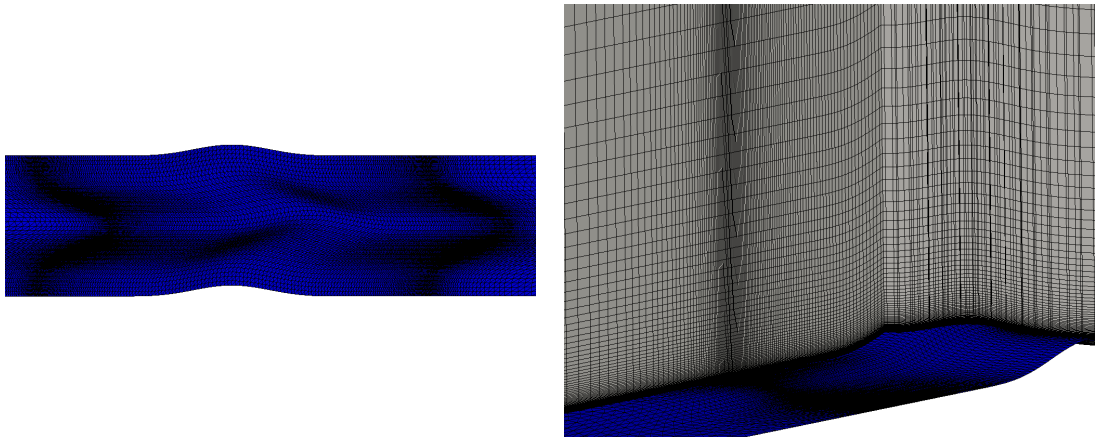
while the x-location of the bump varies in the span-wise direction between y=0 to y=-1 by the equation

$$x = x_0 + 0.3 \sin^4(\pi y) \quad -1 \leq y \leq 0, \tag{3.2}$$

where x_0 is any given x-location along the "two-dimensional" bump definition. The upstream and downstream far-field extends 25 units from the viscous-wall bump, with inviscid-wall boundary conditions imposed on the lower surface between the far-field and viscous-wall. The side walls of the channel also have inviscid wall boundary conditions imposed. The upper boundary is created by extending strands 5.0 units long from the lower surface, with far-field boundary conditions imposed on the end of the strand. For further details on the bump-in-channel geometry, see the NASA-Langley Turbulence Modeling Resource [2].

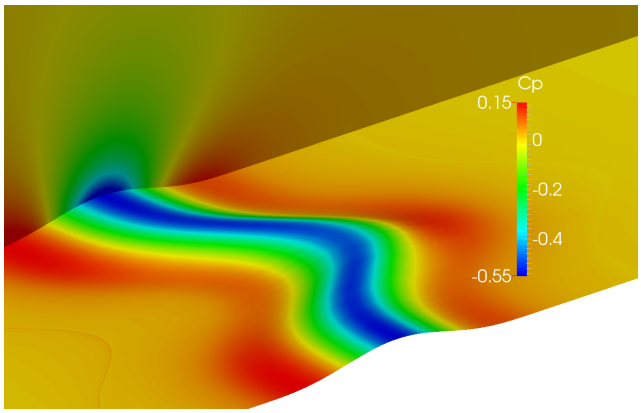
Sub-triangles are created within each parent fourth-order element, resulting in a surface grid containing $17,442$ nodes. The volume grid is created by vertically extending strands from each node, with 160 nodes along each strand, totaling approximately 2.8 million nodes. The surface mesh and volume mesh are shown in Figure 3.1.

The case is run at a Mach number of M=0.2 and a Reynolds number of $Re = 3 \times 10^6$. Coefficient of drag for the entire body, coefficient of pressure over the surface of the body, and normalized eddy viscosity are examined to quantitatively and qualitatively assess the results. Figure 3.2(a) shows the coefficient of pressure contour along the surface of the viscous-wall bump. Figures 3.2(b) and 3.2(c) show the normalized eddy viscosity contours at slices located at $x = 0.3$ and $x = 1.2$, respectively. Table 3.1 contains the computed drag coefficient over the viscous-wall as well as the drag coefficients computed by the established codes. The computed drag coefficient is in close agreement with the values computed with

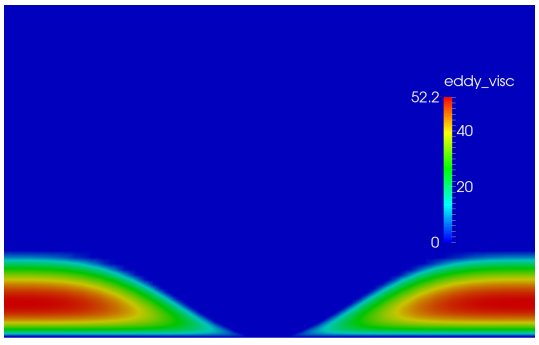


(a) Surface mesh with quartic elements (b) Strand volume mesh

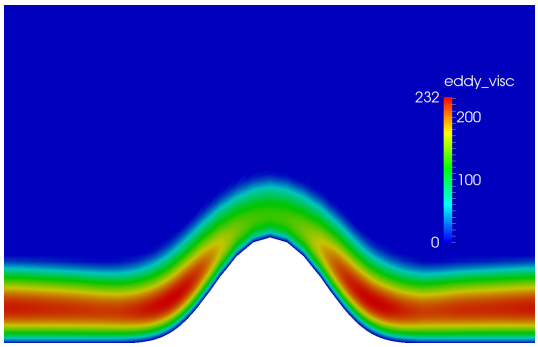
Fig. 3.1: Bump-in-channel mesh configuration



(a) C_p along surface



(b) μ_t/μ_∞ at $x = 0.3$



(c) μ_t/μ_∞ at $x = 1.2$

Fig. 3.2: Contour plots of bump-in-channel at $M = 0.2$

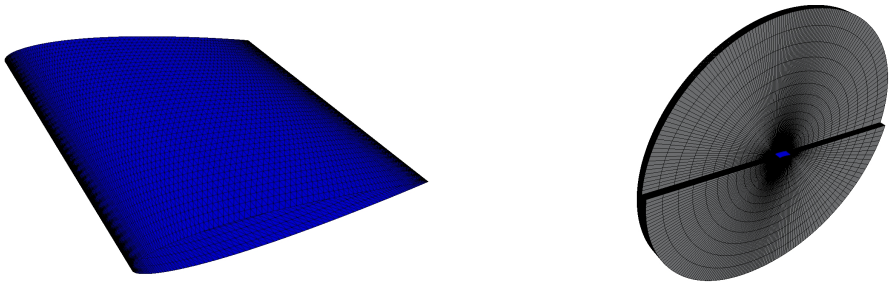
FUN3D and CFL3D.

Table 3.1: Comparison of computed drag coefficients for flow over 3D bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$

Data Source	C_d
Strand FC	3.60970E-3
FUN3D	3.57414E-3
CFL3D	3.58967E-3

3.2 NACA 0012

The next case, a NACA 0012 infinite wing, provides additional geometric complexity in the form of a sharp convex ridge at the trailing edge. To provide adequate mesh resolution around the trailing edge, the strand vectors are smoothed, introducing some degree of mesh skewing. For this case, two flow regimes are examined: subsonic and transonic turbulent flow. The surface mesh consists of 1,280 fourth-order elements and 16,064 nodes, shown in Figure 3.3(a). The volume mesh, with approximately 1 million nodes, was created using strands extending 10 chord lengths from the surface, with 64 nodes along each strand. The resulting mesh is shown in Figure 3.3(b) The wing has a chord length of $c = 1$ and a span of $b = 1$. For both flow conditions studied, far-field boundary conditions are applied on the outer boundary, viscous-wall conditions on the surface of the airfoil, and inviscid planes on the sides of the wing.



(a) Surface mesh with quartic elements

(b) Strand volume mesh

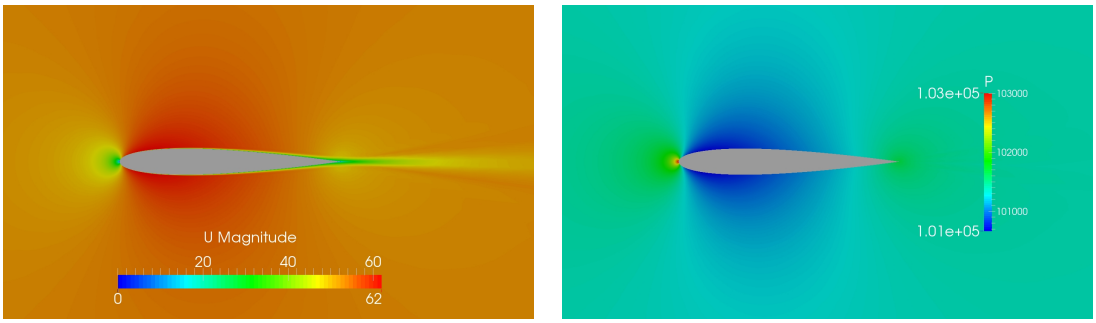
Fig. 3.3: NACA 0012 mesh configuration

3.2.1 Subsonic-Turbulent Flow

First, subsonic turbulent flow over the wing is examined, comparing the results to those outlined in the NASA-Langley Turbulence Modeling Resource [2]. The resource case consists of flow at $M = 0.15$ and $Re = 6 \times 10^6$ at various angles of attack. Corresponding experimental data from Ladson [11], and Gregory and O'Reilly [12] are used for validation. A further description and layout of the case may be found in the NASA-Langley Turbulence Modeling Resource [2].

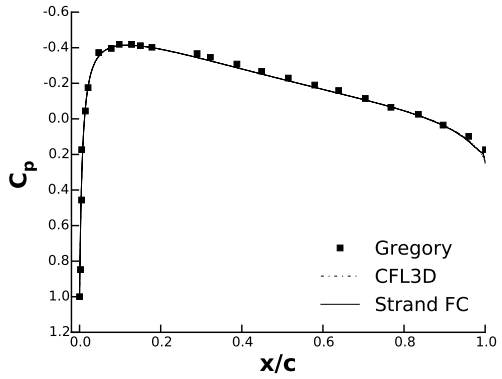
Figure 3.4 shows contours of velocity and pressure around the airfoil. The surface pressure coefficient and coefficient of friction for $\alpha = 0^\circ$ are shown in Figure 3.5, and are compared with the experimental data of Gregory and O'Reilly [12], as well as data from CFL3D [2]. The Gregory data is actually taken at $Re = 3 \times 10^6$, not $Re = 6 \times 10^6$, but little change in pressure and lift is observed between the two Reynolds numbers. Both the FUN3D and CFL3D solvers use a very fine two-dimensional C-grid, totaling 274,329 nodes. In contrast, the three-dimensional strand grid used for this problem does not make specific refinements for the wake.

In Figure 3.5(a), the coefficient of pressure (C_p) along the surface of the airfoil computed from the flux correction results is plotted alongside the results obtained by the CFL3D solver and Gregory's experimental results. Very close agreement is observed between all three results. The coefficients of drag calculated by the different methods are tabulated

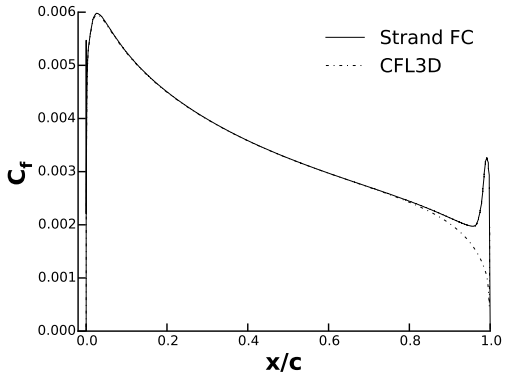


(a) Contours of velocity magnitude (b) Contours of pressure

Fig. 3.4: Field plots of the NACA 0012 at $M = 0.15$



(a) Coefficient of pressure along the surface



(b) Coefficient of friction along upper surface

Fig. 3.5: Coefficient of pressure and friction for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at $\alpha = 0^\circ$

in Table 3.2, as well as the coefficient obtained from experimental results. The calculated drag from the case falls close to the data provided from the NASA Langley turbulence resource [2], and to the experimental data of Ladson [11]. A likely reason for the minor discrepancy between the Strand FC drag and the experimental drag is the lack of an off-body wake-refining Cartesian grid. Similar inaccuracies using strand grids were also observed previously by Work et al. [13]. This is demonstrated in Figure 3.5(b), where the coefficient of friction follows the curve of the CFL3D result until the trailing edge, where the C_f exhibits a peak.

Figure 3.6 shows the effect of wake refinement on the velocity profile obtained from a second-order strand-based code [13]. Figure 3.6(c) shows the configuration of the Cartesian mesh used to obtain the refinement in the wake region. The velocity profile obtained without mesh refinement in the wake region can be seen in Figure 3.6(a). In comparison, it can be seen in Figure 3.6(b) that a refined mesh in the wake region allows the strand-based code to more accurately capture the velocity profile downstream of the airfoil.

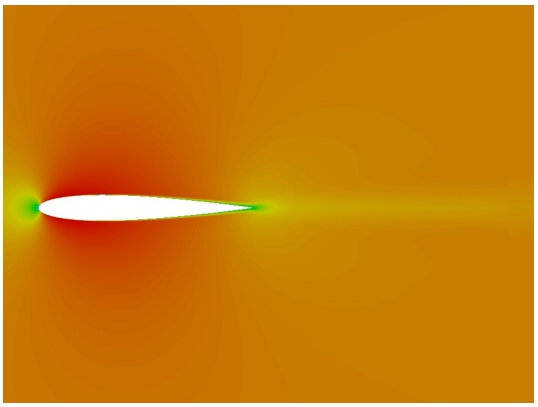
3.2.2 Transonic-Turbulent Flow

To assess the shock-capturing capabilities of the strand code, results obtained for the transonic flow case are compared with data obtained by McDevitt and Okuno [14]. The case was run at a Mach number of $M = 0.759$ and a Reynolds number $Re = 6.3 \times 10^6$ at an angle of attack of $\alpha = 2.05^o$. The results for this case are examined in more detail in the conference paper by Tong and the author [10].

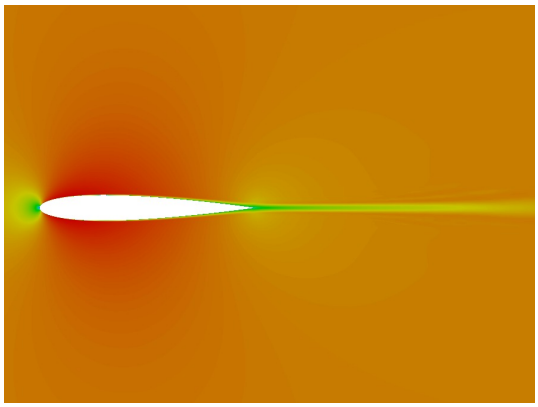
The contours of velocity and pressure around the airfoil can be seen in Figure 3.7. The contours of velocity magnitude show a shock-induced separation of the boundary layer,

Table 3.2: Comparison of computed and experimental drag coefficients for flow over a NACA 0012 airfoil at $M = 0.15$, $\alpha = 0^o$ $Re = 6 \times 10^6$.

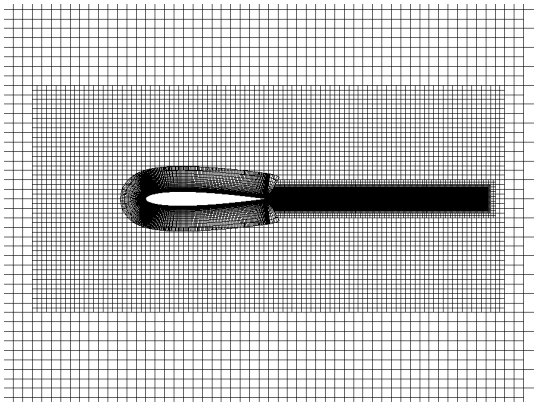
Data Source	C_d
Strand FC	8.44E-3
FUN3D	8.12E-3
CFL3D	8.19E-3
Ladson	8.04E-3



(a) Flow over the airfoil without mesh refinement in the wake region

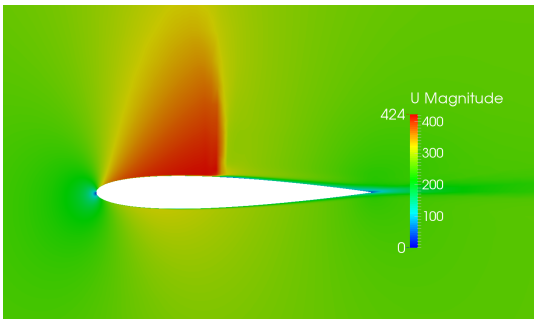


(b) Flow over the airfoil with mesh refinement in the wake region

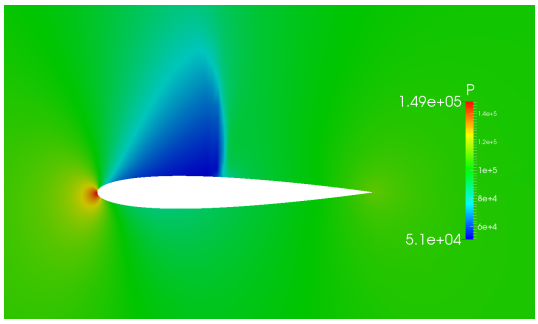


(c) Cartesian mesh grid refinement in the airfoil wake region

Fig. 3.6: Effects of mesh refinement in the NACA 0012 wake region



(a) Velocity magnitude

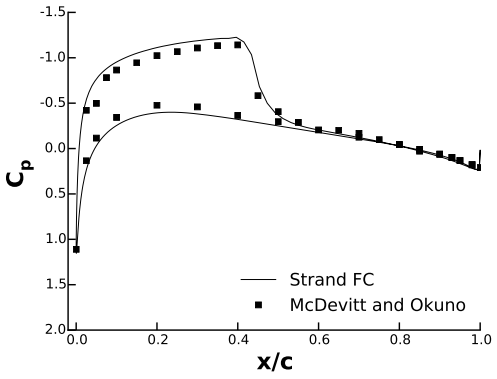


(b) Pressure

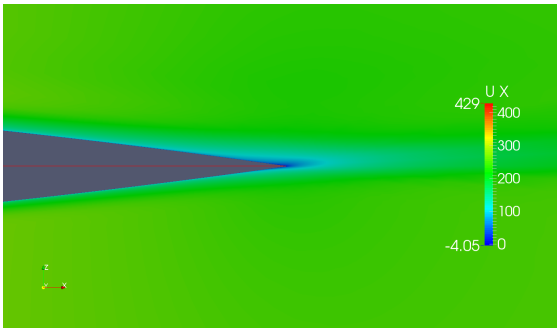
Fig. 3.7: Flow over a NACA 0012 infinite wing, $M = 0.759$, $Re = 6.3 \times 10^6$, and $\alpha = 2.05^o$.

effectively altering the shape of the airfoil, similar to the upward deflection of a trailing edge flap, as shown in Figure 3.8(b). This behavior concurs with that observed by McDevitt and Okuno [14]. The calculated surface pressure coefficient is plotted along with the experimental results from McDevitt and Okuno [14] in Figure 3.8(a).

Looking at Figure 3.8(a), where the pressure coefficient results along the airfoil surface are plotted along with the experimental results from McDevitt and Okuno [14], it can be seen that with the limiter, the shock location is accurately captured on the upper surface. Moreover, observing the strong agreement with experimental data both upstream and downstream of the turbulent shock, it can be concluded that the shock strength is accurately captured by the limiter as well.



(a) Coefficient of pressure along the surface



(b) Contour of velocity showing trailing edge separation

Fig. 3.8: Coefficient of pressure and trailing edge separation for a NACA 0012 airfoil at $M = 0.759$ and $Re = 6.3 \times 10^6$ at $\alpha = 2.05^o$.

3.3 Hemisphere-Cylinder

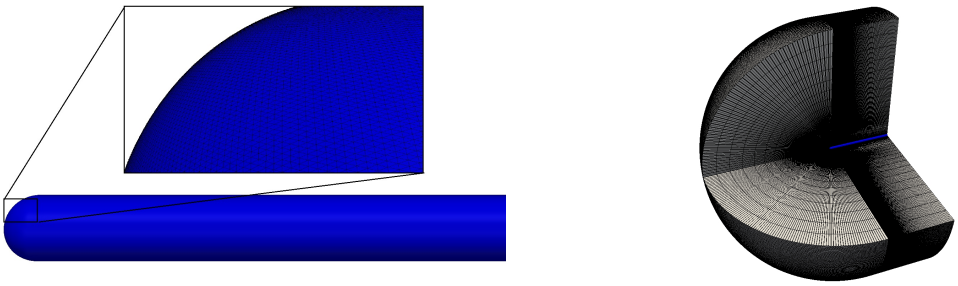
The final validation case examined for flux correction applied to the Spallart-Allmaras turbulence model is flow over a hemisphere-cylinder configuration. Geometry is taken from work by Hsieh [15], where the cylinder has a radius of 0.5 and a length of 10. Further details and grids may be found on the NASA-Langley turbulence resource website [2].

Figure 3.9 shows the hemisphere-cylinder surface and volume mesh configuration. The surface mesh, shown in Figure 3.9(a), is tessellated with 19,050 fourth-order surface elements, totalling 152,701 surface nodes. The strand grid volume mesh, shown in Figure 3.9, consists of 64 nodes along the strands extending for a distance of 40 diameters, resulting in a total mesh size of 9.5 million nodes.

3.3.1 Subsonic-Turbulent Flow

Flow conditions for the subsonic case consist of a Mach number $M = 0.6$ and unit length Reynolds number of $Re = 3.5 \times 10^5$ at an angle of attack of 5°. Details for this flow case and the results obtained can be found in more detail in the article published in the Journal of Scientific Computing by the author and Tong [1].

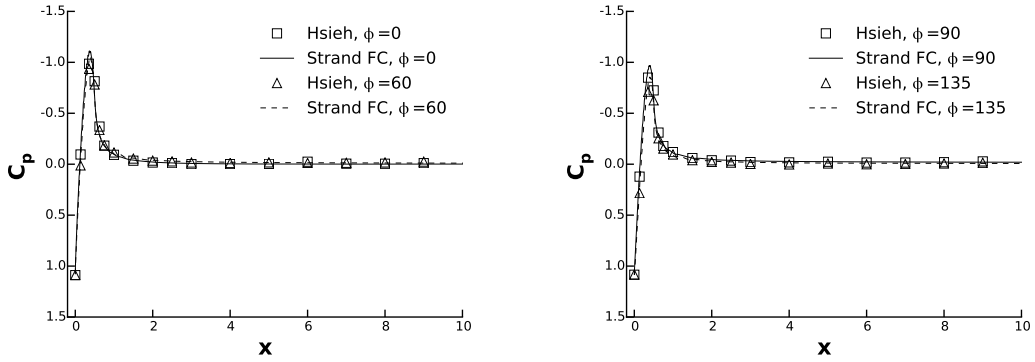
The coefficient of pressure calculated on the surface along the length of the body is compared with the experimental results found by Hsieh [15]. Various angles around the azimuth of the body are considered, namely the $\phi = 0°$, 60°, 90°, 135° positions, with 0° corresponding to the leeward side of the body, pointing away from the incoming wind. Plots



(a) Surface mesh with quartic elements

(b) Strand volume mesh

Fig. 3.9: Hemisphere-cylinder mesh configuration

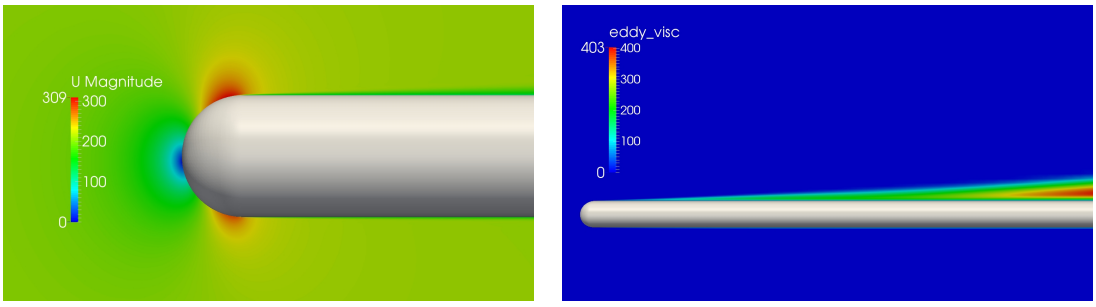


(a) Coefficient of pressure for $\phi = 0°$ and $\phi = 60°$ (b) Coefficient of pressure for $\phi = 90°$ and $\phi = 135°$

Fig. 3.10: Surface coefficient of pressure comparison of experimental and Strand FC for various ϕ

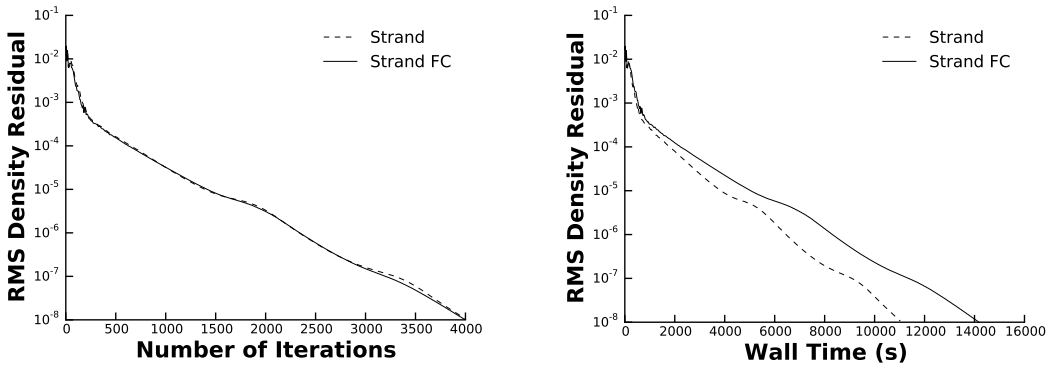
containing the data computed by the strand algorithm in comparison with experimental results are shown in Figure 3.10. It is observed that the pressure coefficient calculated by the flux correction algorithm is in excellent agreement with the pressure data collected by Hsieh.

Field plots of velocity magnitude and normalized eddy-viscosity may be found in Figure 3.11. The velocity and normalized eddy-viscosity profiles shown correspond to the $\phi = 0°$ plane. Figure 3.12 shows the the RMS density residual against the number of iterations and walltime for the second-order scheme ("Strand") and the high-order flux correction scheme ("Strand FC"). The number of iterations required for flux correction strand



(a) Contours of velocity magnitude (b) Contours of normalized eddy-viscosity

Fig. 3.11: Field plots of the hemisphere-cylinder at $M = 0.6$



(a) Iterations vs. RMS density residual

(b) Walltime vs. RMS density residual

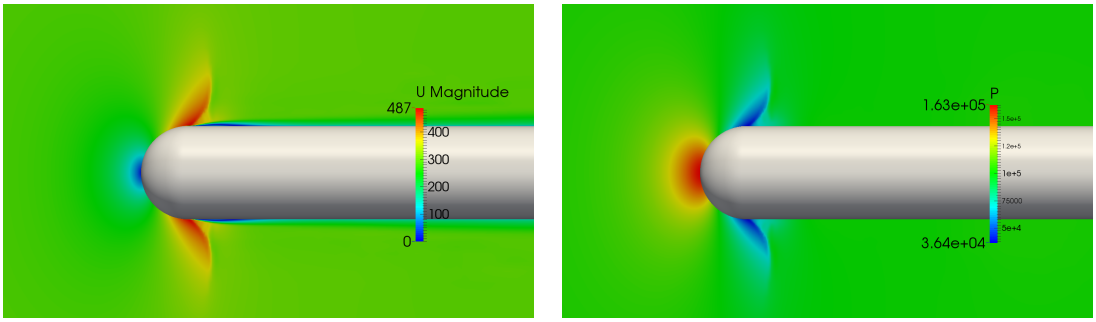
Fig. 3.12: Iterations and walltime vs. RMS density residual for the Strand FC and Strand schemes

scheme to reach the same level of convergence as the second-order strand scheme is nearly identical, as shown in Figure 3.12(a). It is noteworthy that in terms of walltime, the flux correction strand scheme requires less than a 30% increase in computational time from the second-order strand scheme to reach an identical level of convergence. This is observed in Figure 3.12(b). This is a small price to pay for the increased accuracy observed by adding the flux correction terms.

3.3.2 Transonic-Turbulent Flow

For the transonic flow over the hemisphere-cylinder configuration, flow conditions consist of a Mach number $M = 0.85$ and a unit length Reynolds number of $Re = 5 \times 10^6$ at a 0° angle of incidence. The results for this case are examined in more detail in the conference paper by Tong and the author [10].

Field plots of velocity magnitude and pressure are found in Figure 3.13. An oblique shock along with a separation bubble can be seen in Figure 3.13(a) which shows velocity magnitudes surrounding the hemisphere-cylinder body. The formation of the shock and separation bubble are consistent with the experimental results of Hsieh [15]. The coefficient of pressure over the surface of the hemisphere-cylinder is shown in Figure 3.14 alongside Hsieh's experimental results, and theoretical results obtained from the relaxation solution of



(a) Contours of velocity magnitude (b) Contours of pressure

Fig. 3.13: Field plots of the hemisphere-cylinder at $M = 0.85$

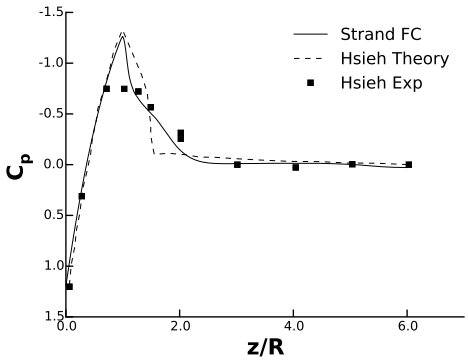


Fig. 3.14: Coefficient of Pressure for transonic flow over a hemisphere-cylinder body

the steady, full potential equation [16]. Good agreement is observed between the results and Hsieh's inviscid theory predictions upstream of the shock. Hsieh attributes the discrepancy of his theoretical and experimental results to boundary layer separation, hence explaining the inviscid theory's failure to accurately predict the flow field. It is unclear exactly why the strand code fails to match the experimental of Hsieh [16], but it is speculated that it is due to the decreased solution order in the shock region caused by the limiter.

Each subfigure in Figure 3.16 corresponds to the normalized velocity found at each point, shown by letter location, in Figure 3.15. The flux correction scheme shows excellent agreement with the theoretical potential flow results presented by Hsieh, particularly in the regions where $Z/R \leq 0.4$, where viscous effects are considered to be of little impact. There are some discrepancies between the flux correction and theory after the shock. However, these may be likely attributed to the separation bubble that forms after the shock, which potential flow theory does not account for. In the regions where $Z/R \geq 0.8$, the flux correction scheme and theory show some deviation from the experimental results. Hsieh [15] claims that this deviation is due to particle lag, which cannot be predicted by the code or the theory, from the laser Doppler velocimetry method used to obtain the results. Nonetheless, the flux correction results generally agree well with the theory here.

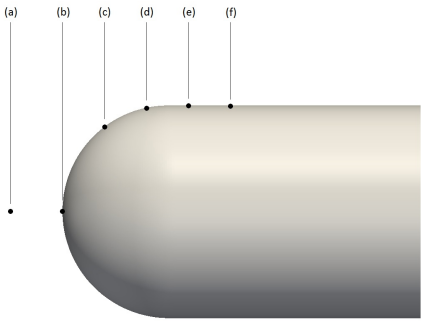
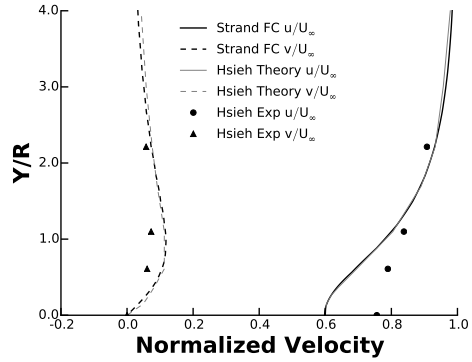
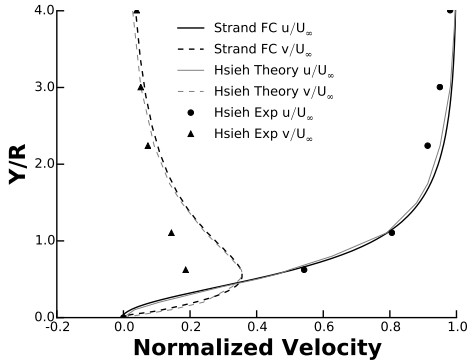


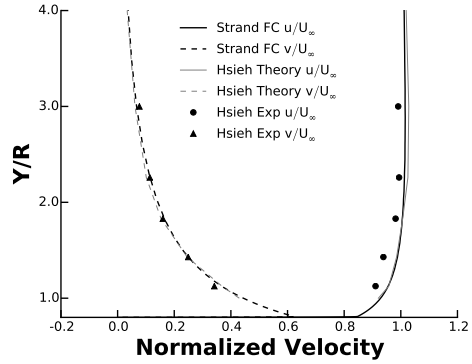
Fig. 3.15: Location of velocity profiles along the hemisphere-cylinder



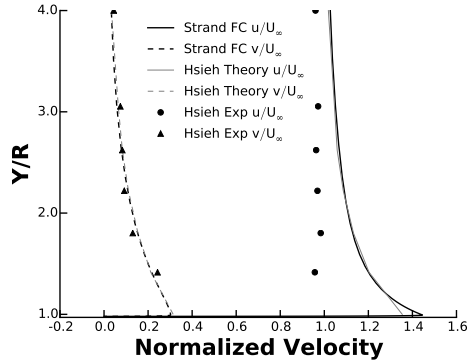
(a) Velocity profile in y-direction at $z/R = -0.5$



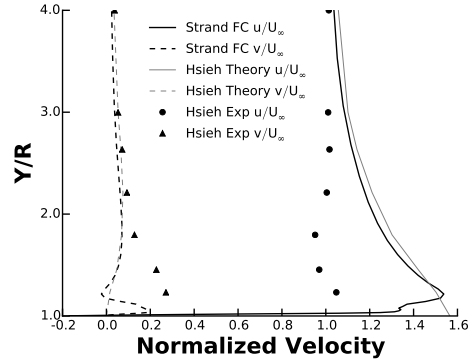
(b) Velocity profile in y-direction at $z/R = 0.0$



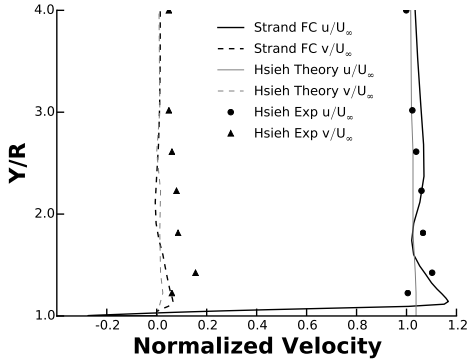
(c) Velocity profile in y-direction at $z/R = 0.4$



(d) Velocity profile in y-direction at $z/R = 0.8$



(e) Velocity profile in y-direction at $z/R = 1.2$



(f) Velocity profile in y-direction at $z/R = 1.6$

Fig. 3.16: Velocity field about hemisphere-cylinder at various z/R locations

Chapter 4

Domain Connectivity

The domain connectivity between the overset grids takes advantage of the inherent structures found in the stand and Cartesian grids to provide quick and scalable connectivity. Donor-recipient relations are established to maintain the transfer of information between the grids.

4.1 Cartesian Donor Cells for Strand Recipients

The strand solver determines which nodes receive information from the Cartesian solver via the local clipping index. For this work, the clipping index has been set such that the last two strand nodes on each strand receives information from the Cartesian solver.

The index of the Cartesian cell that each recipient strand node resides in is determined by equation 4.1, where $\mathbf{x}_s$ is the position vector for the strand node, $\mathbf{x}_0$ is the coordinate where the Cartesian domain begins, Δx_l is the Cartesian spacing used on a particular refinement level, and $\mathbf{I}_s$ is the global Cartesian index vector that the strand node resides in for any particular level of refinement.

$$\mathbf{I}_s = floor\Big(\frac{\mathbf{x}_s - \mathbf{x}_0}{\Delta x_l}\Big) \tag{4.1}$$

Since each level of refinement does not span the entirety of the Cartesian domain, a search is conducted to determine if a Cartesian cell associated with the computed index exists. If the cell does not exist, a lower refinement level is used to recompute the Cartesian index. If the cell is found, the donor-recipient relation for that strand node is established, and each node composing that cell is marked as a donor node. The solution variables are then interpolated onto each recipient strand node using a trilinear interpolation from the 8 donor nodes comprising the Cartesian cell in which the strand node resides.

4.2 Strand Donor cells for Cartesian Recipients

Establishing correct donor-recipient relations for the Cartesian solver to receive information from the strand solver is a much more difficult task. An implicit hole-cutting scheme is implemented to determine which Cartesian nodes require donor cells. First, a bounding box is formed around the global strand mesh, eliminating the need to search through any Cartesian nodes that are not within close proximity of the strand mesh. An octree is then formed, containing the spatial location, surface index j, and strand index n for each strand node. A search algorithm introduced by Hjaltason [17] can then be used to create a list of the closest strand nodes to each Cartesian node within the bounding box. If the strand index n for the closest strand node is larger than the clipping index c_j for the strand on which it lies, the Cartesian node is close enough to the outer boundary of the strand mesh that it can be treated as a regularly computed node in the Cartesian domain. Similarly, if the Cartesian node is connected to a surface node ($n = 0$) the Cartesian node is close enough to the surface mesh that it can be considered to be inside the body, and therefore cut out as a "hole" from the Cartesian domain. In practice, it has been found helpful to change the hole cutting condition from $n = 0$ to $n < k$, where k denotes some number of nodes up the strand still close enough to the surface that any Cartesian node connected to those nodes are also considered to be close enough to the surface. This measure is employed to prevent any convergence issues that may arise in this region due to the high aspect ratios found in the cell geometries.

Once all unnecessary nodes have been removed by the implicit hole-cutting method, the Cartesian node is ready to be connected to its strand donor cell. Due to the unstructured nature of the surface and non-orthogonality of the strands to the surface, it is difficult to determine if a Cartesian node is contained within a particular strand stack. A combination of Lagrange polynomials and a three-dimensional Newton's method can be used to determine the (r, s, η) coordinates of a particular Cartesian node in the computational domain of any given strand stack. For each strand node (j, n), a list of all strand cells containing that node is compiled, and Newton's method cycles through each individual strand cell in

this list. The coordinates found by Newton's method are compared to established limits for the reference cell. If Newton's method fails to converge to an acceptable value in all cells surrounding the strand node, the search moves onto the next closest strand node found by the octree, and repeats the search.

Once an acceptable set of coordinates is found, the donor-recipient relation for that Cartesian node is established and the surface element index and strand index are recorded. With (r, s, η) known, Lagrange polynomials can be used to interpolate values onto each receiver Cartesian node. This method accounts for the high-order surface curvature and assures at least second-order spatial accuracy.

4.2.1 Lagrange Polynomials

A continuous mapping between the spatial domain and computational domain can be created from a finite number of points using Lagrange polynomials. If the values of a variable ϕ are known on the nodes of a strand cell, they can be interpolated to any point within that strand cell via

$$\phi(r, s, \eta) = \sum_{i=0}^{q \cdot N - 1} \phi_i L_i(r, s, \eta) \tag{4.2}$$

where N is the number of nodes in a surface element and the Lagrange polynomials, L_i, are defined as

$$L_i(r, s, \eta) = \sum_{l=0}^{q-1} \sum_{j=0}^{p} \sum_{k=0}^{p-j} \alpha_{ijkl} r^j s^k \eta^l. \tag{4.3}$$

Here, q is the stencil size in the strand direction, p is the surface mesh order, and (r, s, η) represent the coordinates of interpolation within the computational domain. In this work, a second-order strand stencil ($q = 2$) and fourth-order surface order ($p = 4$) are used. Since the computational coordinates of the strand nodes are known, the linear system shown in Eq. 4.4 can be solved for the coefficients α_{ijkl}. The form for the f matrix shown in Eq. 4.4

was chosen to simplify the organization and computation of the matrices.

$$\begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{q*N-1} \end{pmatrix} = \Big[f(r,s,\eta) \Big] \Big[\alpha_{ijkl} \Big] \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{q*N-1} \end{pmatrix} \tag{4.4}$$

$$\Big[f(r,s,\eta) \Big] = \begin{bmatrix} [A] & [A]\eta_0 & [A]\eta_0^2 \\ [A] & [A]\eta_1 & [A]\eta_1^2 \\ [A] & [A]\eta_2 & [A]\eta_2^2 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & \dots & r_0^j s_0^k \\ \vdots & \ddots & \vdots \\ 1 & \dots & r_{N-1}^j s_{N-1}^k \end{bmatrix}$$

4.2.2 Newton's Method

To numerically compute the computational coordinates (r, s, η) of any given Cartesian node, Newton's method is used to minimize the square of the distance function

$$d^2 = \mathbf{d} \cdot \mathbf{d}. \tag{4.5}$$

The squared distance function was chosen as an alternative to the plain distance function as it is smooth in space and shares the same minimum as the distance function. The distance vector $\mathbf{d}$ is defined as the difference between the Cartesian node coordinates and the spatial coordinates of a given point (r, s, η) in the computational domain, as shown in Eq. 4.6.

$$\mathbf{d} = \mathbf{x}_{cart} - \mathbf{x}(r,s,\eta). \tag{4.6}$$

Here, $\mathbf{x}(r, s, \eta)$ is computed using the Lagrange interpolant

$$\mathbf{x}(r, s, \eta) = \sum_{i=0}^{q*N-1} \mathbf{x}_i L_i(r, s, \eta), \tag{4.7}$$

where $\mathbf{x}_i$ are the spatial coordinates of each strand node included in the interpolation stencil.

$$\mathbf{r}^{\tau+1} = \mathbf{r}^{\tau} - [H]^{-1} \nabla d^2 \tag{4.8}$$

Newton's method must be applied to a function's derivative in order to find its minimum, as shown in Eq. 4.8. In the equation, $[H]^{-1}$ represents the inverse of the Hessian matrix and $\mathbf{r}$ represents the computational coordinates at each iteration of the Newton's method. The gradient of d^2 is defined in Eq. 4.9a, and the components of the Hessian matrix are defined,in Einstein notation, in Eq. 4.9b.

$$\frac{\partial(d^2)}{\partial r_j} = 2 d_i \frac{\partial d_i}{\partial r_j}, \tag{4.9a}$$

$$H_{jk} = \frac{\partial}{\partial r_k}\Big(\frac{\partial(d^2)}{\partial r_j}\Big) = 2\Big(d_i \frac{\partial d_i}{\partial r_k \partial r_j} + \frac{\partial d_i}{\partial r_j}\frac{\partial d_i}{\partial r_k}\Big) \tag{4.9b}$$

Chapter 5

Numerical Results

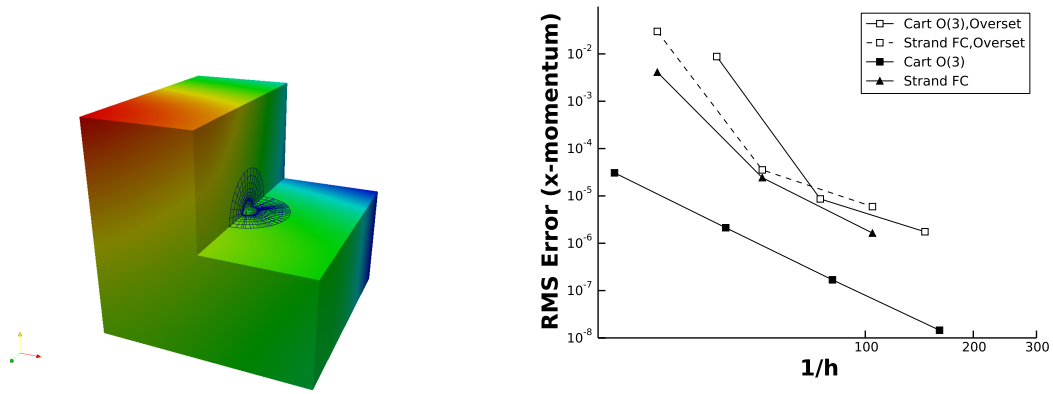
Verification and validation of the SCIM library are examined through a variety of methods. Verification is examined through the method of manufactured solutions and the inviscid flow over a sphere case. Validation of the SCIM library is conducted with steady and unsteady computations of flow over a sphere at various low Reynolds numbers.

5.1 Method of Manufactured Solutions

The method of manufactured solutions, first introduced by Roache [18], provides a means for verifying the order of accuracy of a numerical method implementation by creating and solving for an artificial solution. An analytic solution is first chosen, and the numerical method is forced to converge on that solution through the use of source terms that allow the solution to satisfy the governing equations. A grid refinement study is then conducted to determine the order of accuracy of the method. The manufactured density solution contour used in this work can be seen in figure 5.1(a).

Figure 5.1(b) shows the grid refinement results. The SCIM code shows a high sensitivity to grid refinement ($O(h^{10})$) when coarser grids are being used, but becomes less sensitive ($O(h^{2.5})$) as the refinement is increased. From this trend, the conclusion can be drawn that the overset method performs poorly when the mesh used in its computations is too coarse.

As the Cartesian solver independently demonstrates second and third-order accuracy and the strand solver demonstrates third and fourth-order accuracy, it is apparent that the order of accuracy is being limited by the domain connectivity. This is most likely due to the interpolation methods being used in the SCIM library. As the refinement is increased, the number of interpolated values also increases, increasing the amount of error introduced by the interpolation schemes, and therefore negating many of the benefits obtained by the



(a) Manufactured density solution

(b) SCIM grid refinement study

Fig. 5.1: Verification of SCIM using MMS

high-order solution methods on each domain. The interpolation errors reduce the order of accuracy on the strands to $O(h^{2.525})$ and the Cartesian order to $O(h^{2.377})$. Further research into higher-order interpolation schemes could help to remedy this decrease in order of accuracy.

5.2 Steady Sphere

Steady laminar flow over a sphere is examined for validation purposes. The surface mesh used in these computations was comprised of 1,048 fourth-order triangular surface elements, with strands being extended two diameters from the surface containing 32 nodes along each strand, creating 503,040 cells. The Cartesian grid, extending from 8 diameters upstream to 14 diameters downstream and 8 diameters on all sides of the sphere, provided the majority of the spacial domain. The Cartesian grid was initially refined around the strand mesh such that the size of the Cartesian cells were similar to the size of the strand cells near the ends of the strands. An adaptive mesh refinement algorithm was then implemented to obtain the necessary refinement in the wake region. The final Cartesian mesh contained 461,237 cells.

The flow was computed for low Reynolds numbers at Re=40, 80, 120, 160, and 200 at a Mach number $M = 0.2$. The separation angle, vortex recirculation length, and vortex center

coordinates for the standing ring vortex were recorded for each Reynolds number. It can be observed in figure 5.2 that the SCIM library accurately predicts the increasing recirculation length as the Reynolds number is increased. This is shown in more detail in figure 5.3(b). Though the third-order SCIM library predicts the recirculation length to be a little high, the SCIM library comes within 5% of results found by Tomboulides, et al. [19] and only 15% of results found by Magnaudet et al [20]. The SCIM library correctly exhibits the linear relationship of the four parameters shown in figure 5.3 with the logarithm of the Reynolds number, as described by Taneda [21] in his experimental work. In both figures 5.3(c) and 5.3(d), the second-order overset results deviate from this linear trend at $Re = 200$. The flux corrected overset results successfully follow the expected linear trend. Overall, the flux corrected overset method produced results that more closely fit the experimental results of Taneda [21], and the computational results by Tomboulides [19] and Magnaudet [20].

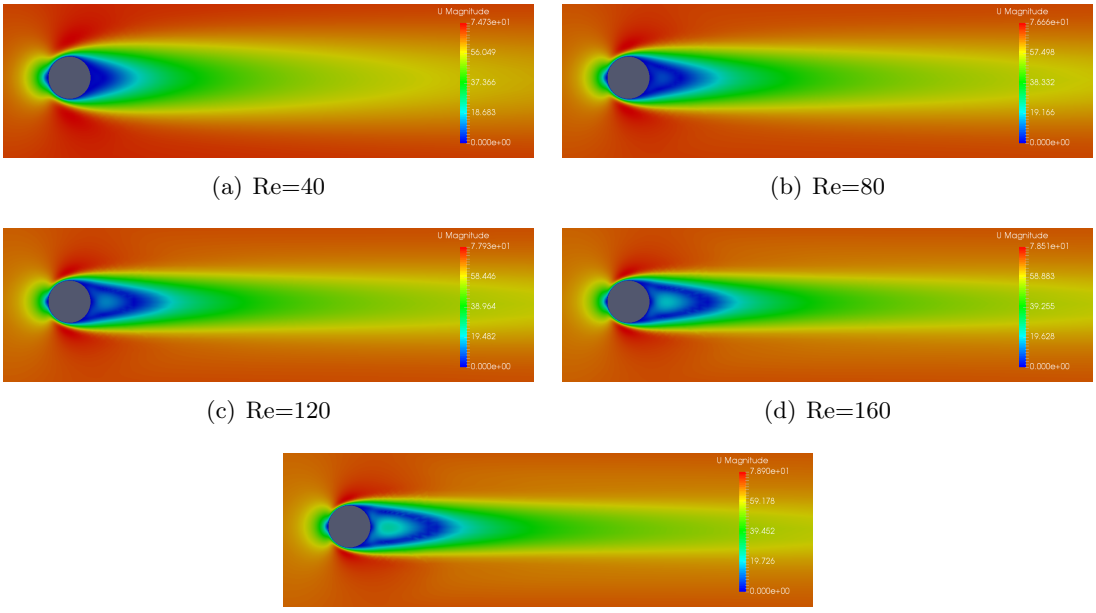
Tong's computations were made using only strand grids which extended to 20 diameters with 128 nodes along each strand, creating a volume mesh of over 2.1 million cells [1]. It is important to note that the SCIM library produced superior results with less than half as many computational cells as the strand grid alone.

5.3 Unsteady Sphere

For the final SCIM validation case, unsteady flow around a sphere is examined. At Reynolds numbers greater than 300, the flow around a sphere becomes unsteady with trailing vortices being shed behind the sphere. Results obtained from SCIM are compared to experimental results presented by Sakamoto and Haniu [22].

To quantitatively compare the computational and experimental results, the Strouhal number of the flow is computed using the frequency of fluctuation in the flow velocity at 3.5 diameters downstream of the sphere's rear surface. A spectrum analysis conducted via Fourier transformation gives the dominant frequency to be used in the computation of the Strouhal number given in equation 5.1.

$$St = \frac{fD}{U_o} \tag{5.1}$$



(a) Re=40 (b) Re=80

(c) Re=120 (d) Re=160

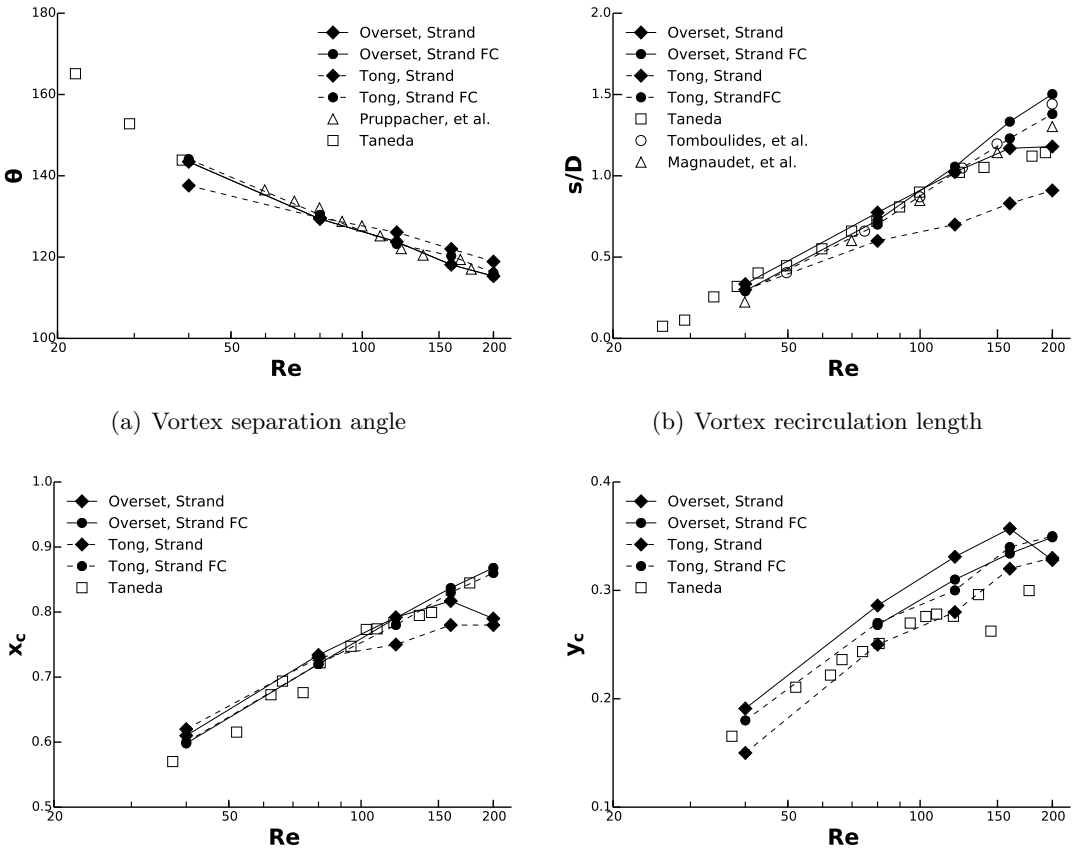
(e) Re=200

Fig. 5.2: Velocity profiles for various Reynolds numbers.

An example of the spectrum analysis can be seen in figure 5.4(a). The results obtained for various Reynolds numbers are shown in figure 5.4(b).

As can be seen by the discrepancy in Strouhal numbers, the SCIM library fails to accurately compute the vortex shedding behind the sphere. In fact, the SCIM library's accuracy decreased when Flux Correction was applied to the computations. The reason for this is still unclear as the application of Flux Correction has in every other case shown significant benefits to the accuracy of the computed results. It is speculated that the lower-order interpolation methods used in the domain connectivity play a role in introducing error into the system, as the stand-alone strand solver with Flux Correction predicts the Strouhal number more accurately than the second-order schemes.

One improvement that comes from the use of Flux Correction, however, lies in the nature of the force fluctuations measured. Sakamoto and Haniu show that the fluctuation waveform should become irregular at $Re > 480$. As seen in figure 5.3,which shows the force oscillations in the $Re = 600$ case, without Flux Correction the fluctuation in the force computed remains periodic; but with the use of Flux Correction the flow is correctly



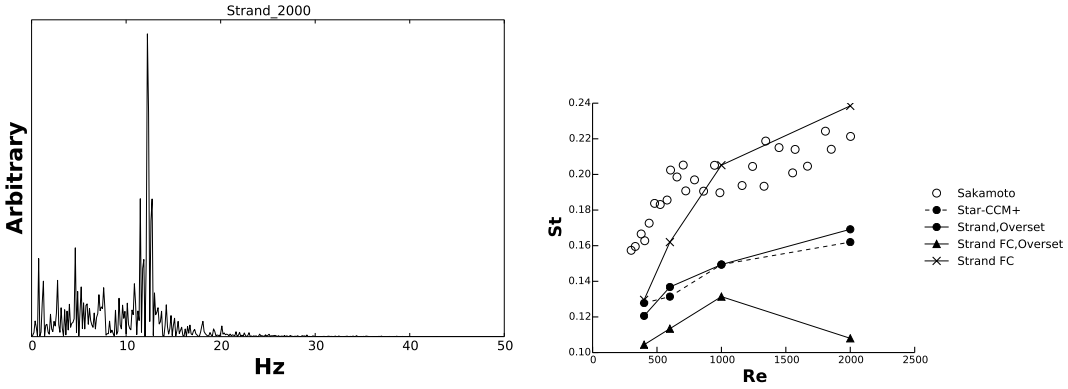
(a) Vortex separation angle

(b) Vortex recirculation length

(c) x_c vortex center

(d) y_c vortex center

Fig. 5.3: Low Reynolds number laminar sphere results from overset SCIM compared against results obtained by Tong [1] on the strand solver and experimental results. Both second-order FV methods(Strand) and high-order Flux Correction (Strand FC) are shown.

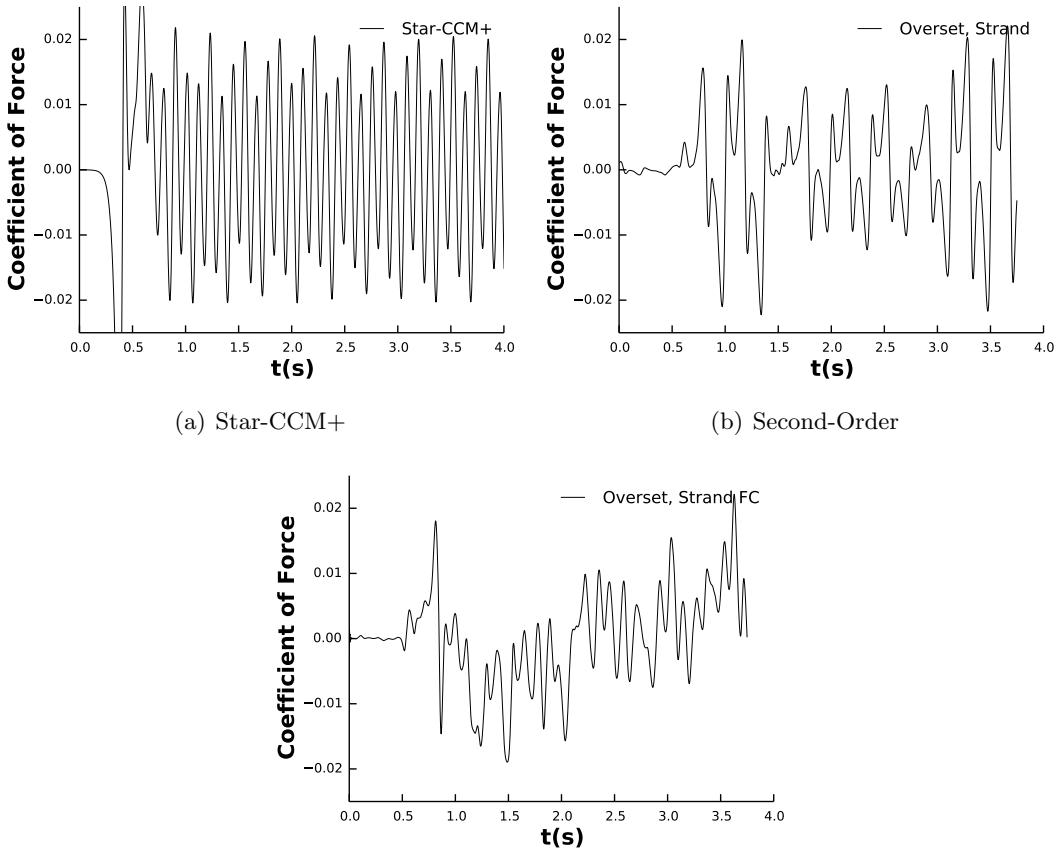


(a) Spectrum analysis

(b) Strouhal numbers plotted against Reynolds number

Fig. 5.4: Spectrum analysis and Strouhal numbers. (a) shows only an example of the spectrum analysis results. See Appendix for results from various Reynolds numbers.

predicted to be irregular. This incorrect periodic fluctuation is also seen in results obtained through the second-order commercial code Star-CCM+.



(a) Star-CCM+

(b) Second-Order

(c) Third-Order

Fig. 5.5: Flux Correction correctly computed irregular force fluctuations, whereas second-order methods failed to capture this behavior.

Chapter 6

Conclusions and Future Work

The primary goal of strand grids is to provide a framework to efficiently generate near-body volume meshes on which flow solutions can be accurately computed in conjunction with off-body solvers. Second-order schemes have been used in the past on strand grids, and have been successful in revealing the potential they hold for fast and accurate flow computations. However, second-order schemes have also proven inadequate when coupled with high-order off-body solvers.

Flux correction has been developed as a means of increasing the order of accuracy within the near-body strand solver. Flux correction reduces truncation error within the solution by using high-order gradients to compute reconstructed fluxes. These reconstructed fluxes are used within a modified finite volume scheme on the strand grids, resulting in near fourth-order accuracy within the strand solution.

Validation of flux correction has been conducted on various geometries and multiple flow regimes, and has proven flux correction to be an effective method for increasing solution accuracy. However, due to the nature of strand grids, mesh resolution decreases as the strands extend further from the surface. This loss in resolution decreases the reliability of the solution obtained in these regions, demonstrating the need for the strand solver to be coupled with an off-body solver.

The strand-Cartesian interface manager incorporates second-order trilinear interpolation and mixed order Lagrange interpolation to successfully compute flow solutions. The SCIM library was verified using the method of manufactured solutions. The solution scheme as a whole was found to be near third-order accurate, though its constituents are third and near fourth-order accurate. This decrease in accuracy is attributed to the lower order-of-accuracy interpolation methods used within the scheme. Nonetheless, the strand-Cartesian

overset method successfully provides near and far-body computational frameworks that are fast and scalable.

With this work, the domain connectivity framework has been laid for the SCIM library to take on more complex problems. The incorporation of Reynolds-Averaged Navier-Stokes turbulence models allows the SCIM library to handle a wider range of flow regimes. With further research into conservative high-order interpolation methods, the order of accuracy of the combined method can be improved. More scalable domain connectivity methods would help to extend the method to more complicated geometries. Implementation of moving grid methods into the current flux correction scheme would also help to broaden SCIM's applications.

References

[1] Tong, O., Katz, A., Yanagita, Y., Casey, A., and Schaap, R., "High-Order Methods for Turbulent Flows on Three-Dimensional Strand Grids," *Journal of Scientific Computing*, 2015.

[2] Rumsey, C., "NASA Langley Turbulence Modeling Resource (http://turbmodels.larc.nasa.gov)," 2015.

[3] Katz, A. and Work, D., "High-Order Flux Correction/Finite Difference Schemes for Strand Grids," *Journal of Computational Physics*, Vol. 282, February 2015, pp. 360–380.

[4] Strand, B., "Summation by parts for finite difference approximation for d/dx," *Journal of Computational Physics*, Vol. 110, 1994, pp. 47–67.

[5] Mattsson, K., "Summation by Parts Operators for Finite Difference Approximations of Second-Derivatives with Variable Coefficients," *Journal of Scientific Computing*, Vol. 51, 2012, pp. 650–682.

[6] Pincock, B. and Katz, A., "High-Order Flux Correction for Viscous Flows on Arbitrary Unstructured Grids," *AIAA paper*, AIAA 21st Computational Fluid Dynamics Conference, San Diego, CA, June 2013.

[7] Katz, A. and Sankaran, V., "An Efficient Correction Method to Obtain a Formally Third-Order Accurate Flow Solver for Node-Centered Unstructured Grids," *Journal of Scientific Computing*, Vol. 51, 2012, pp. 375–393.

[8] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 1 Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence," *International Journal of Computational Fluid Dynamics*, Vol. 4, 1995, pp. 171–218.

[9] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 2 Artificial Diffusion and Discrete Shock Structure," *International Journal of Computational Fluid Dynamics*, Vol. 5, 1995, pp. 1–38.

[10] Tong, O., Yanagita, Y., Schaap, R., Harris, S., and Katz, A., "High-Order Strand Grid Methods for Shock Turbulence Interaction," *AIAA paper* 2015-2283, AIAA 22nd Computatonal Fluid Dynamics Conference, Dallas, TX, 2015.

[11] Ladson, C., "Effects of Independent Variation of Mach and Reynolds Numbers on the Low-Speed Aerodynamic Characteristics of the NACA 0012 Airfoil Section," *NASA TM* 4074, October 1988.

[12] Gregory, N. and O'Reilly, C., "Low-Speed Aerodynamic Characteristics of NACA 0012 Aerofoil Sections, including the Effects of Upper-Surface Roughness Simulation Hoar Frost," *NASA RM* 3726, January 1970.

[13] Work, D., Tong, O., Workman, R., Katz, A., and Wissink, A., "Strand-Grid-Solution Procedures for Sharp Corners," *AIAA Journal*, Vol. 52, 2014, pp. 1528–1541.

[14] McDevitt, J. and Okuno, A., "Static and Dynamic Pressure Measurements on a NACA 0012 Airfoil in the Ames High Reynolds Number Facility," *NASA TM* 2485, June 1985.

[15] Hsieh, T., "An Investigation of Separated Flow about a Hemisphere-Cylinder at 0- to 19- Deg Incidence in the Mach Number Range from 0.6 to 1.5," *AEDC-TR* 76-112, Arnold Engineering Development Center, 1976.

[16] Hsieh, T., "Hemisphere-cylinder in transonic flow, free-stream Mach number from 0.7 to 1.0," *AIAA Journal*, Vol. 13, 1975, pp. 1411–1413.

[17] Hjaltason, G. R. and Samet, H., "Ranking in Spatial Databases," Tech. rep., Proceedings of the 4th Symposium on Spatial Databases, 1995.

[18] Roache, P. J., "Code Verification by the Method of Manufactured Solutions," *Journal of Fluids Engineering*, Vol. 124, March 2002, pp. 4–10.

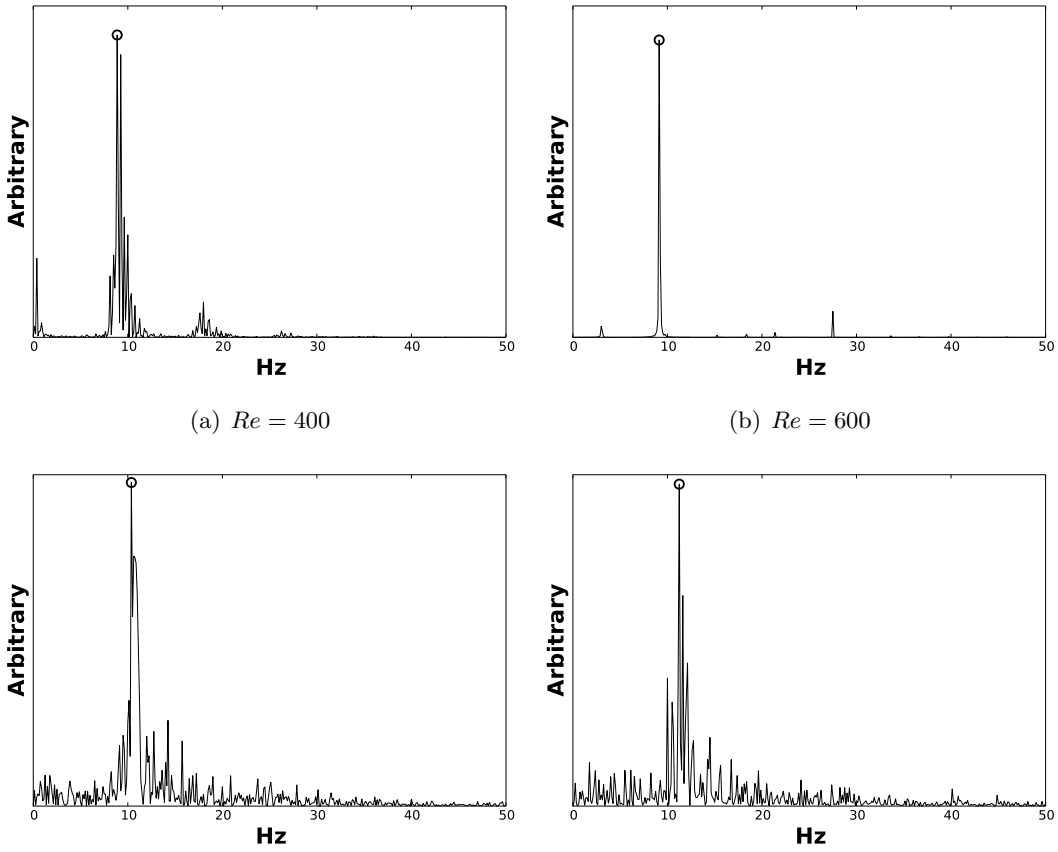
[19] Tomboulides, A., Orszag, S., and Karniadakis, G., "Direct and Large Eddy Simulations of Axisymmetric Wakes," *AIAA paper* 93-0546, 1993.

[20] Magnaudet, J., Riviero, M., and Fabre, J., "Accelerated Flows Past a Sphere or Spherical bubble. Part 1: Steady Streaming Flow," *Journal of Fluid Mechanics*, Vol. 284, 1995, pp. 97–135.

[21] Taneda, S., "Experimental Investigation of Wake Behind a Sphere at Low Reynolds Numbers," *J. Physical Society of Japan*, Vol. 11, 1956, pp. 1104–1108.

[22] Sakamoto, H. and Haniu, H., "A Study on Vortex Shedding From Spheres in a Uniform Flow," *Journal of Fluids Engineering*, Vol. 112, December 1990, pp. 386–392.

Appendix



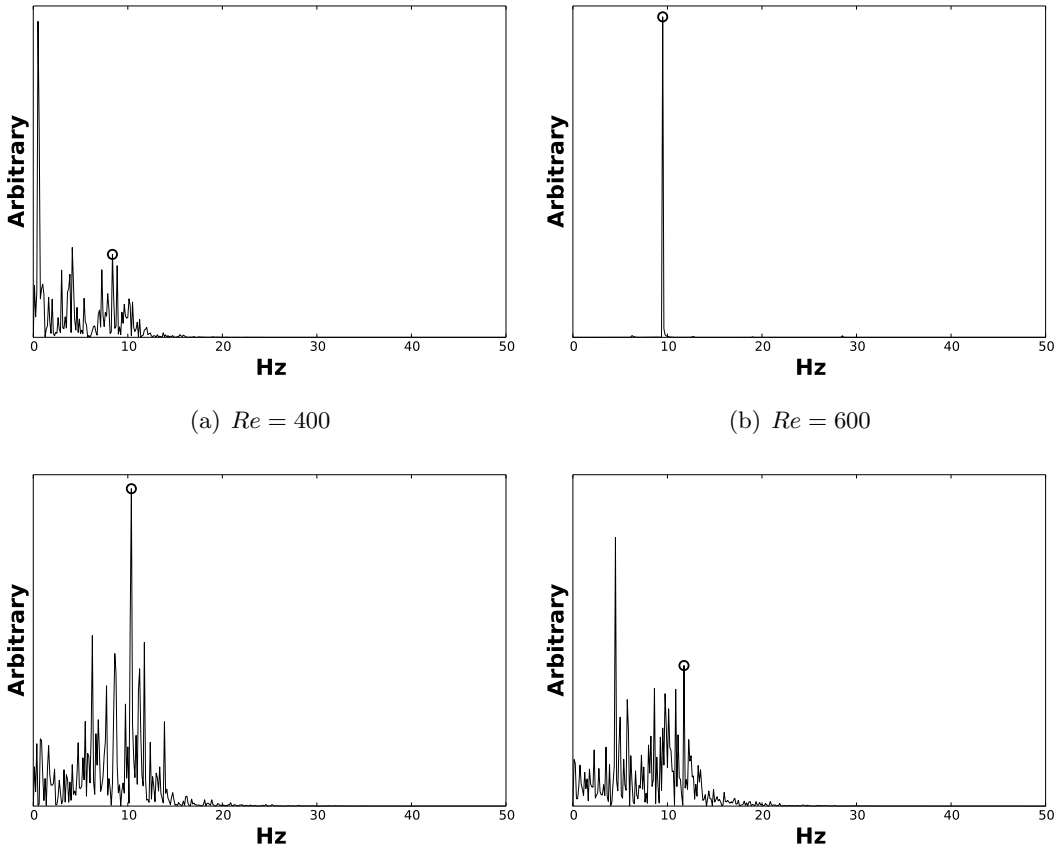
(a) $Re = 400$

(b) $Re = 600$

(c) $Re = 1000$

(d) $Re = 2000$

Fig. 1: Spectral analysis plots for Star-CCM+ cases



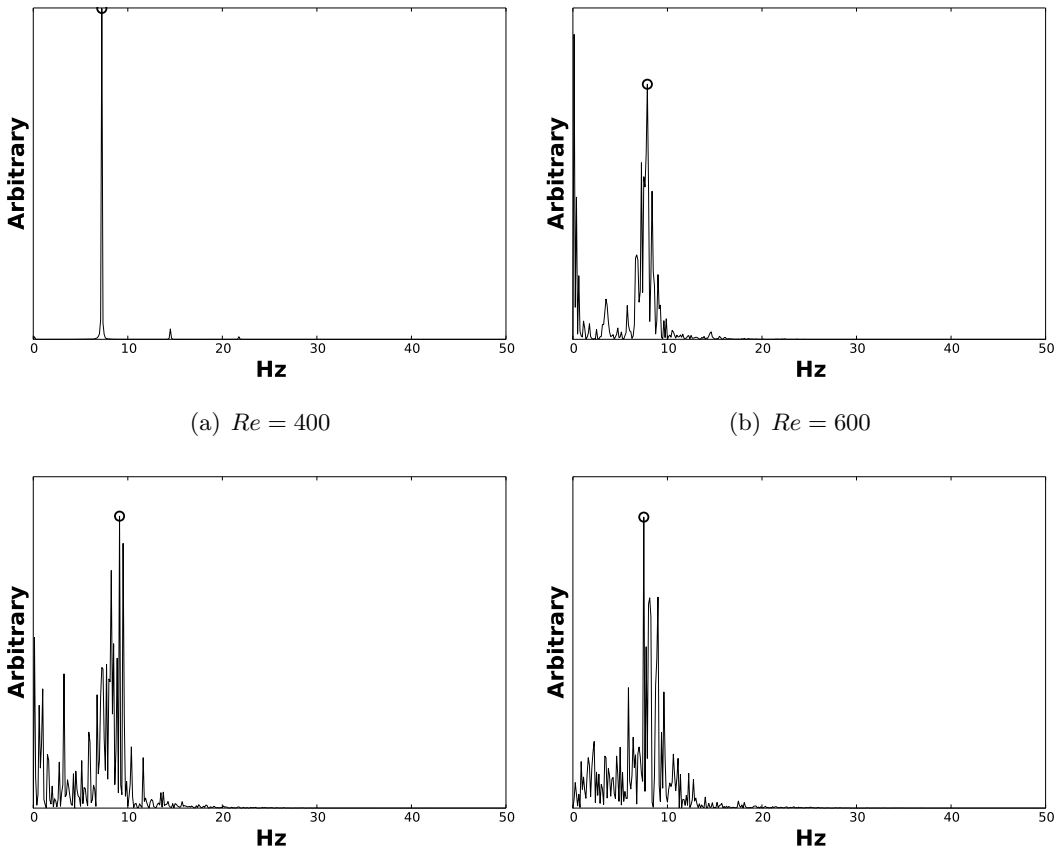
(a) $Re = 400$

(b) $Re = 600$

(c) $Re = 1000$

(d) $Re = 2000$

Fig. 2: Spectral analysis plots for Second-Order SCIM cases



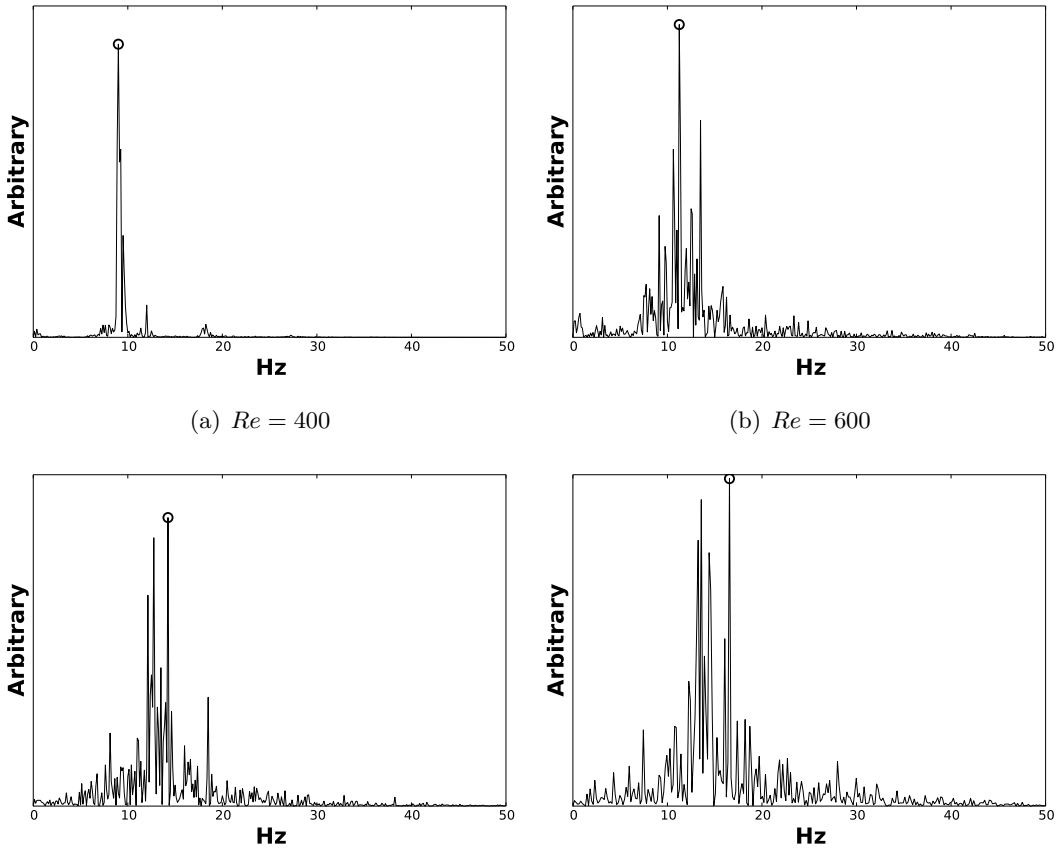
(a) $Re = 400$

(b) $Re = 600$

(c) $Re = 1000$

(d) $Re = 2000$

Fig. 3: Spectral analysis plots for Third-Order SCIM cases



(a) $Re = 400$

(b) $Re = 600$

(c) $Re = 1000$

(d) $Re = 2000$

Fig. 4: Spectral analysis plots for Third-Order Strand cases