

Utah State University

DigitalCommons@USU

---

All Graduate Theses and Dissertations

Graduate Studies

---

5-2016

## Development of a Three-Dimensional High-Order Strand-Grids Approach

Oisin Tong  
*Utah State University*

Follow this and additional works at: <https://digitalcommons.usu.edu/etd>



Part of the [Aerospace Engineering Commons](#), and the [Mechanical Engineering Commons](#)

---

### Recommended Citation

Tong, Oisin, "Development of a Three-Dimensional High-Order Strand-Grids Approach" (2016). *All Graduate Theses and Dissertations*. 4711.  
<https://digitalcommons.usu.edu/etd/4711>

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



DEVELOPMENT OF A THREE-DIMENSIONAL HIGH-ORDER  
STRAND-GRIDS APPROACH

by

Oisin Tong

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

of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Approved:

---

Dr. Aaron Katz  
Major Professor

---

Dr. Robert Spall  
Committee Member

---

Dr. Heng Ban  
Committee Member

---

Dr. Barton Smith  
Committee Member

---

Dr. Michael Johnson  
Committee Member

---

Dr. Mark R. McLellan  
Vice President for Research and  
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY  
Logan, Utah

2016

Copyright © Oisin Tong 2016

All Rights Reserved

## ABSTRACT

Development of a Three-Dimensional High-Order Strand-Grids Approach

by

Oisín Tong, Doctor of Philosophy

Utah State University, 2016

Major Professor: Dr. Aaron Katz  
Department: Mechanical and Aerospace Engineering

Development of a novel high-order flux correction method on strand grids is presented. The method uses a combination of flux correction in the unstructured plane and summation-by-parts operators in the strand direction to achieve high-fidelity solutions. Low-order truncation errors are cancelled with accurate flux and solution gradients in the flux correction method, thereby achieving a formal order of accuracy of 3, although higher orders are often obtained, especially for highly viscous flows.

In this work, the scheme is extended to high-Reynolds number computations in both two and three dimensions. Turbulence closure is achieved with a robust version of the Spalart-Allmaras turbulence model that accommodates negative values of the turbulence working variable, and the Menter SST turbulence model, which blends the  $k-\epsilon$  and  $k-\omega$  turbulence models for better accuracy. A major advantage of this high-order formulation is the ability to implement traditional finite volume-like limiters to cleanly capture shocked and discontinuous flows. In this work, this approach is explored via a symmetric limited positive (SLIP) limiter.

Extensive verification and validation is conducted in two and three dimensions to determine the accuracy and fidelity of the scheme for a number of different cases. Verification

studies show that the scheme achieves better than third order accuracy for low and high-Reynolds number flows. Cost studies show that in three-dimensions, the third-order flux correction scheme requires only 30% more walltime than a traditional second-order scheme on strand grids to achieve the same level of convergence.

In order to overcome meshing issues at sharp corners and other small-scale features, a unique approach to traditional geometry, coined “asymptotic geometry,” is explored. Asymptotic geometry is achieved by filtering out small-scale features in a level set domain through min/max flow. This approach is combined with a curvature based strand shortening strategy in order to qualitatively improve strand grid mesh quality.

(172 pages)

## PUBLIC ABSTRACT

Development of a Three-Dimensional High-Order Strand-Grids Approach

by

Oisín Tong, Doctor of Philosophy

Utah State University, 2016

Major Professor: Dr. Aaron Katz  
Department: Mechanical and Aerospace Engineering

The strand-Cartesian grid approach is a unique method of generating and computing fluid dynamic simulations. The strand-Cartesian approach provides highly desirable qualities of fully-automatic grid generation and high accuracy. This work focuses on development of a high-accuracy methodology (high-order scheme) on strand grids for two and three dimensions.

In this work, the high-order scheme is extended to high-Reynolds number computations in both two and three dimensions with the Spalart-Allmaras turbulence model and the Menter SST turbulence model. In addition, a simple limiter is explored to allow the high-order scheme to accurately predict discontinuous flows.

Extensive verification and validation is conducted in two and three dimensions to determine the accuracy and fidelity of the scheme for a number of different cases. Verification studies show that the scheme is indeed high-order for various flows. Cost studies show that in three-dimensions, the high-order scheme required only 30% more computational time than a traditional scheme. In order to overcome meshing issues at sharp corners and other small-scale features, a unique approach to traditional geometry, coined “asymptotic geometry,” is explored.

## ACKNOWLEDGMENTS

Thanking everyone who has contributed to my dissertation would be an impossible task, however, there are some who I must thank individually. First and foremost, I must thank my advisor, Dr. Aaron Katz, without whom, this journey would not have been possible. I am forever grateful for the unwavering belief he has shown in me, and near saint-like patience he has exhibited on the many times I came into his office asking stupid questions. His guidance and support throughout the years have been instrumental in bringing me to this point today. I am proud to be the first PhD advisee in what I believe will be a long list of students in the future in Dr. Katz's quest to revolutionize CFD.

Next, I would like to thank all my fellow colleagues in the research lab, past and present. Each of them have helped work out math with me, run cases, debug computer code, and been my friend. For all of this and more, I am grateful.

Finally, I would like to acknowledge the support of parents, family, and fiancé, who have provided me a pillar of support to lean on. Thank you all who have helped me reach this point.

Oisin Tong

## CONTENTS

	Page
ABSTRACT . . . . .	iii
PUBLIC ABSTRACT . . . . .	v
ACKNOWLEDGMENTS . . . . .	vi
LIST OF TABLES . . . . .	x
LIST OF FIGURES . . . . .	xi
1 INTRODUCTION . . . . .	1
1.1 The Strand-Cartesian Approach . . . . .	2
1.2 Flux-Correction High-Order Method . . . . .	3
1.3 Verification and Validation . . . . .	4
1.3.1 Verification . . . . .	4
1.3.2 Validation . . . . .	5
1.4 Outline of Thesis . . . . .	6
REFERENCES . . . . .	7
2 VERIFICATION AND VALIDATION OF A HIGH-ORDER STRAND GRID METHOD FOR TWO-DIMENSIONAL TURBULENT FLOWS . . . . .	11
2.1 Abstract . . . . .	11
2.2 Introduction . . . . .	11
2.3 High-Order Hybrid Discretization . . . . .	14
2.4 Semi-Implicit Multigrid Solution Method . . . . .	20
2.5 Results . . . . .	23
2.5.1 Verification Studies with the Method of Manufactured Solutions . . . . .	23
2.5.2 Zero Pressure Gradient Flat Plate . . . . .	26
2.5.3 Bump-in-Channel . . . . .	28
2.5.4 NACA 0012 Airfoil . . . . .	29
2.6 Conclusions and Future Work . . . . .	33
REFERENCES . . . . .	36
3 HIGH-ORDER METHODS FOR TURBULENT FLOWS ON THREE-DIMENSIONAL STRAND GRIDS . . . . .	41
3.1 Abstract . . . . .	41
3.2 Introduction . . . . .	41
3.3 High-Order Strand Grid Discretization . . . . .	44
3.3.1 Unstructured Flux Correction in the $r$ - $s$ Plane . . . . .	48
3.3.2 SBP Finite Differences in the $\eta$ -Direction . . . . .	50
3.3.3 Numerical Approximation of Element Mappings . . . . .	51
3.4 Results . . . . .	53



3.4.1	Verification Studies with Manufactured Solutions . . . . .	53
3.4.2	Steady Flow over a Sphere . . . . .	55
3.4.3	Unsteady Flow over a Sphere . . . . .	57
3.4.4	Hemisphere-Cylinder . . . . .	61
3.5	Conclusions . . . . .	63
3.6	Acknowledgments . . . . .	65
3.7	Compliance with Ethical Standards . . . . .	65
	REFERENCES . . . . .	66
4	HIGH-ORDER STRAND GRID METHODS FOR SHOCK TURBULENCE INTERACTION . . . . .	71
4.1	Abstract . . . . .	71
4.2	Introduction . . . . .	71
4.3	High-Order Strand Grid Discretization . . . . .	74
4.4	Results . . . . .	81
4.4.1	Bump-in-Channel . . . . .	81
4.4.2	NACA 0012 . . . . .	84
4.4.3	Transonic Hemisphere-Cylinder . . . . .	89
4.5	Conclusions and Future Work . . . . .	91
	REFERENCES . . . . .	95
5	ASSESSMENT OF A TWO-EQUATION TURBULENCE MODEL IN THE HIGH-ORDER FLUX CORRECTION SCHEME . . . . .	100
5.1	Abstract . . . . .	100
5.2	Introduction . . . . .	100
5.3	High-Order Strand Grid Discretization . . . . .	103
5.4	Turbulence Model Implementation . . . . .	109
5.4.1	Oscillation Damping Treatment . . . . .	109
5.4.2	Inviscid and Dissipation Flux Treatment . . . . .	111
5.5	Results . . . . .	114
5.5.1	Method of Manufactured Solutions . . . . .	114
5.5.2	Zero Pressure Gradient Flat Plate . . . . .	116
5.5.3	Bump in Channel . . . . .	120
5.6	Conclusions and Future Work . . . . .	122
	REFERENCES . . . . .	125
6	ASYMPTOTIC GEOMETRY REPRESENTATION FOR COMPLEX CONFIGURATIONS ON STRAND GRIDS . . . . .	131
6.1	Abstract . . . . .	131
6.2	Introduction . . . . .	131
6.3	Small-Scale Feature Smoothing Procedure . . . . .	134
6.3.1	Level Set Representation . . . . .	135
6.3.2	Min/Max Flow . . . . .	136
6.3.3	Asymptotic Geometry Representations . . . . .	137
6.4	Strand Grid Generation Procedure . . . . .	140
6.4.1	Strand Grid Shortening Procedure . . . . .	142
6.5	Strand Grid Generation on Asymptotic Geometry . . . . .	144

6.5.1 Cylinder-Block Geometry . . . . .	144
6.6 Conclusions and Future Work . . . . .	145
REFERENCES . . . . .	150
7 CONCLUSIONS AND FUTURE WORK . . . . .	153
CURRICULUM VITAE . . . . .	155

## LIST OF TABLES

Table		Page
2.1	Order of accuracy of inviscid terms using (2,2), (3,3) and (3,4) schemes (x-momentum error). . . . .	24
2.2	Order of accuracy of viscous terms using (2,2), (3,3) and (3,4) schemes (x-momentum error). . . . .	24
2.3	Order of accuracy of inviscid and viscous terms combined ( $Re = 100,000$ ) using (2,2), (3,3) and (3,4) schemes (x-momentum error). . . . .	24
2.4	Comparison of computed drag coefficients for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	28
2.5	Comparison of computed drag coefficients for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	29
2.6	Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at $M = 0.15$ , $\alpha = 15^\circ$ $Re = 6 \times 10^6$ . . . . .	33
3.1	Order of accuracy of inviscid and viscous terms combined ( $Re = 10$ ) (x-momentum). . . . .	55
3.2	Order of accuracy of inviscid, viscous and source terms combined ( $Re = 100,000$ ) (x-momentum). . . . .	57
4.1	Comparison of computed drag coefficients for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	82
4.2	Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at $M = 0.15$ , $\alpha = 0$ degrees, and $Re = 6 \times 10^6$ . . . . .	87
5.1	Error for inviscid, viscous and source terms combined ( $Re = 100,000$ ) for a cube geometry (x-momentum). . . . .	116
5.2	Comparison of computed drag coefficients for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	120
5.3	Comparison of computed drag coefficients for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	122
6.1	Asymptotic error for a cube. . . . .	140

## LIST OF FIGURES

Figure	Page
1.1 Strand-Cartesian grid system. . . . .	3
2.1 Strand grid elements and example strand/Cartesian grid system for the TRAM rotor. . . . .	12
2.2 Two-dimensional mapping of strand stack from physical space to computational space. . . . .	16
2.3 Grid and manufactured solution used for grid refinement study. . . . .	25
2.4 Error convergence at various flow regimes using the method of manufactured solutions with schemes (2,2), (3,3), and (3,4) . . . . .	26
2.5 Grid and normalized turbulent eddy-viscosity contours for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	27
2.6 Comparison of normalized stream-wise velocity and normalized turbulent eddy-viscosity profiles for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	28
2.7 Grid and turbulent viscosity contours for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	30
2.8 Comparison of stream-wise velocity and turbulent viscosity profiles for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	30
2.9 Bump-in-channel surface coefficient of pressure and friction. . . . .	31
2.10 Grid system for flow over a NACA 0012 airfoil. . . . .	31
2.11 Pressure coefficient for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at various angles of attack. . . . .	32
2.12 $C_l$ vs. $\alpha$ and $C_l$ vs. $C_d$ compared to experiment for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ . . . . .	32
3.1 Strand grid elements and example strand/Cartesian grid system for the TRAM rotor. . . . .	43
3.2 Mapping of strand stack from physical space to computational space. . . . .	46

3.3	Element mappings used for gradient reconstruction. . . . .	47
3.4	Manufactured solution used for order of accuracy verification studies. . . . .	54
3.5	Order of accuracy results for manufactured solution in a cube geometry. . . . .	54
3.6	Verification study for manufactured solution in a sphere geometry. . . . .	56
3.7	Sphere mesh configuration. . . . .	56
3.8	Comparison of computed high-order strand results with experimental data measuring various flow characteristics. . . . .	58
3.9	Field plot of pressure for steady flow over a sphere for $M = 0.2$ at various Reynolds numbers. . . . .	58
3.10	Velocity vectors for steady flow over a sphere for $M = 0.2$ at various Reynolds numbers. . . . .	59
3.11	Coefficient of force in the z-direction over time. . . . .	60
3.12	Unsteady flow visualization of a sphere at $M = 0.2$ and $Re = 600$ . . . . .	60
3.13	Hemisphere-cylinder mesh configuration. . . . .	61
3.14	Surface coefficient of pressure comparison of experimental and Strand FC for various $\phi$ . . . . .	62
3.15	Field plots of the hemisphere-cylinder at $M = 0.6$ . . . . .	63
3.16	Iterations and walltime vs. RMS density residual for the Strand FC and Strand schemes. . . . .	64
4.1	Strand grid elements and example strand/Cartesian grid system for the TRAM rotor. . . . .	73
4.2	Mapping of strand stack from physical space to computational space. . . . .	76
4.3	Element mappings used for gradient reconstruction. . . . .	77
4.4	Bump-in-channel mesh configuration. . . . .	83
4.5	Field plots of the bump-in-channel at $M = 0.2$ . . . . .	83
4.6	Comparison of stream-wise velocity and turbulent viscosity profiles for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	83
4.7	Coefficient of pressure and friction for flow over the bump at $M = 0.2$ . . . . .	84

4.8	Field plots of the bump-in-channel at $M = 0.8$ . . . . .	84
4.9	NACA 0012 mesh configuration. . . . .	85
4.10	Field plots of the NACA 0012 at $M = 0.15$ . . . . .	86
4.11	Coefficient of pressure and friction for flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at $\alpha = 0$ degrees. . . . .	86
4.12	Field plots of the hemisphere-cylinder at $M = 0.759$ . . . . .	88
4.13	Coefficient of pressure and trailing edge separation for a NACA 0012 airfoil at $M = 0.759$ and $Re = 6.3 \times 10^6$ at $\alpha = 2.05$ degrees. . . . .	88
4.14	Hemisphere-cylinder mesh configuration. . . . .	89
4.15	Field plots of the hemisphere-cylinder at $M = 0.85$ . . . . .	90
4.16	Location of velocity profiles along the hemisphere-cylinder. . . . .	90
4.17	Velocity field about hemisphere-cylinder at various $z$ locations. . . . .	92
5.1	Strand grid elements and example strand/Cartesian grid system for the TRAM rotor. . . . .	102
5.2	Mapping of strand stack from physical space to computational space. . . . .	106
5.3	Element mappings used for gradient reconstruction. . . . .	107
5.4	Manufactured solution and order of accuracy results in a cube geometry. . . . .	115
5.5	Manufactured solution and order of accuracy results in a sphere geometry. . . . .	116
5.6	Grid and normalized turbulent eddy-viscosity contours for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	117
5.7	Comparison of normalized stream-wise velocity profiles for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	118
5.8	Comparison of normalized turbulent kinetic energy and normalized specific dissipation rate profiles for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$ . . . . .	119
5.9	Comparison of normalized turbulent kinetic energy and normalized specific dissipation rate profiles with and without a SLIP limiter for medium grid refinement over a flat plate. . . . .	119
5.10	Grid and turbulent viscosity contours for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	121

5.11	Comparison of normalized stream-wise velocity profiles for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	121
5.12	Comparison of normalized turbulent kinetic energy and specific dissipation rate profiles for flow through a bump-in-channel at $M = 0.2$ and $Re = 3 \times 10^6$ . . . . .	122
5.13	Bump-in-channel surface coefficient of pressure and friction. . . . .	123
6.1	Strand grid elements and example strand/Cartesian grid system for the TRAM rotor. . . . .	133
6.2	Level set representation of a cube. . . . .	135
6.3	Asymptotic geometry representation for a cube. . . . .	138
6.4	Order of accuracy results for asymptotic geometry for a cube geometry. . . . .	138
6.5	Asymptotic geometry representation for a cube with a sharp point. . . . .	139
6.6	Asymptotic geometry representation for a cylinder with an intersecting rectangular block. . . . .	141
6.7	Demonstration of strand direction vector smoothing. (a) Strands with direction vector normal to surface, no smoothing applied, (b) strands with smoothed direction vectors . . . . .	142
6.8	Volume mesh of a cube with and without strand-shortening. . . . .	145
6.9	Two-dimensional slice of a cube with and without strand-shortening. . . . .	146
6.10	Volume mesh of a duct with and without strand-shortening. . . . .	147
6.11	Two-dimensional slice of a duct with and without strand-shortening. . . . .	148
6.12	Volume mesh of cylinder-block. . . . .	149
6.13	Two-dimensional slices of cylinder-block. . . . .	149
6.14	Close-up of cylinder-block internal corner. . . . .	150

## CHAPTER 1

### INTRODUCTION

In recent years, improvements in hardware and the embracement of parallel computing has seen computational fluid dynamics (CFD) become an industry standard tool for design and an active research area. Real-time solutions for complex flows provide the ultimate goal to any CFD developer. However, in order to achieve this, there still exists challenges in obtaining high-fidelity solutions for high-Reynolds number viscous flows over complex multi-body geometry which must first be conquered. First, the mesh generation is arguably the greatest obstacle and bottleneck. Generation of meshes for complex multi-body geometry requires considerable experience, and even at that, mesh experts can still spend days or even weeks generating a satisfactory mesh. As research into massively parallel computing continues, with even the possibility of breaking the latency barrier [1], the percentage of time devoted to mesh generation relative to simulation time will only be further increased. Fully automated meshing is an obvious, yet difficult solution. Second, numerically diffuse traditional second-order schemes often display accuracy limitation on unstructured grids. Such methods are still dominant among CFD practitioners. In order for high-order methods on unstructured grids to be embraced by CFD practitioners, high-order methods need to be refined, and high-order unstructured meshes need to be accessible. Finally, complex systems require ever-increasing mesh sizes, for which scalability becomes a greater issue. Automating viscous mesh generation, preserving spatial and temporal accuracy, and maintaining computational efficiency are currently among the greatest research challenges in CFD today.

The primary goal of this work is to address these challenges and develop a three-dimensional approach capable of alleviating these issues via flux correction and strand-Cartesian approach, as shown in Figure 1.1. In this work, we focus on the near-body, (strand grids), as adaptive high-order Cartesian solution methods in the off-body are al-



ready mature. Furthermore, recent work by Wissink et. al. [2] shows excessive near-body discretization error can cause non-physical diffusion of vortical flow features, spoiling the accuracy of even the most advanced high-order off-body solution techniques. Near-body accuracy is critical to obtaining body force computations, specifically drag. While high-order wake-capturing in the off-body is critical for capturing vortex-body interactions, near-body accuracy is imperative to properly capturing the initial generation of these structures from the surface.

### 1.1 The Strand-Cartesian Approach

The strand-Cartesian approach was developed to directly tackle the challenges detailed, and has shown some potential to alleviating these issues [3–5]. Strand and Cartesian grids allow the possibility of fully automatic volume grid generation while enhancing scalability and the potential for high-order accuracy.

In the strand paradigm, a body-fitted near-body mesh is constructed by a set of straight line segments grown directly from the surface, each with the same point distribution in the normal direction, forming a thin layer of mostly prismatic elements around the body. Once outside the viscous boundary layer, strands transition to isotropic block structured Cartesian grids. Strand and Cartesian grids communicate through implicit overset interpolation [6–8], which is greatly facilitated by the fact that the entire strand-Cartesian mesh system can be stored on each processor due to its compact grid representation. The procedure is similar in concept to standard prismatic unstructured grid generation techniques, in which prismatic cells are grown at the surface in the viscous boundary layer with tetrahedra elsewhere, except in the strand approach prismatic lines are straight, and Cartesian grids are used in place of tetrahedra for the Euler solution.

In addition to streamlined and automatic meshing capability, the strand-Cartesian approach presents three other important advantages. First, both strand and Cartesian meshes may be represented with extremely low memory descriptions, enabling the entire global mesh description to fit on each processor in a parallel environment. This allows for self-satisfying domain connectivity [3] and reduces the percentage of time needed for

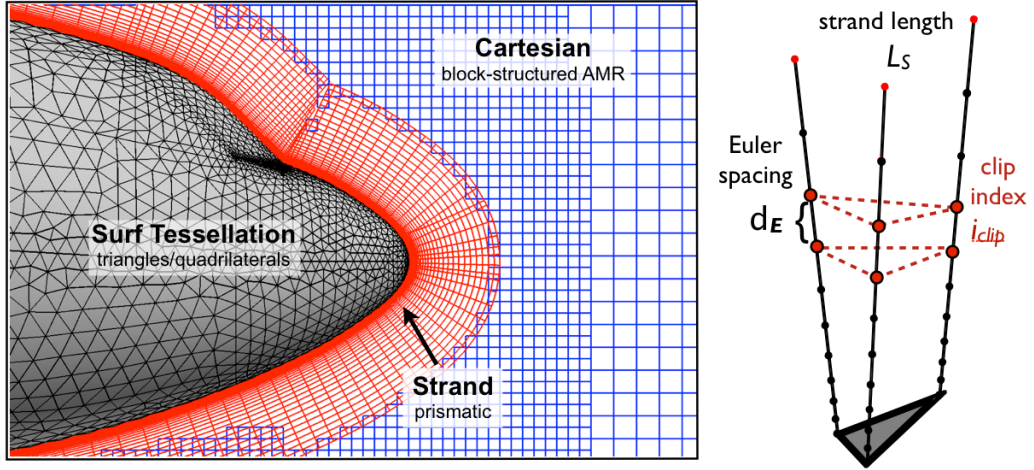


Fig. 1.1: Strand-Cartesian grid system.

inter-grid communication. [9]. The savings become even more significant in the case of moving body simulations for which domain connectivity must be re-established at each unsteady time-step. Second, both strand and Cartesian meshes possess at least some grid structure, facilitating efficient implementations of high-order accurate discretizations and solution methods. These methods include high-order finite differencing, line-implicit solvers, and directional multi-grid coarsening. Third, both the strand and Cartesian grids easily permit use of Adaptive Mesh Refinement (AMR). Because all strands use the same normal point distribution, adaptation is entirely surface-based. This avoids cell quality and edge swapping complexities that have traditionally plagued volume-based unstructured AMR. AMR on Cartesian grids has been known for years to be very effective because the logical data structure naturally facilitates a hierarchical mesh representation and Cartesian cells do not suffer cell quality issues with frequent and persistent adaptation, as can occur with tetrahedral elements.

## 1.2 Flux-Correction High-Order Method

Flux correction is a novel method of obtaining better than third-order accuracy on strand grids, proposed to directly address the issues detailed. Prior research has shown [10–12] the method shown a number of improvements over traditional second-order methods

and even some complex high-order methods. The method is unique in two ways. First, unlike most high-order methods under investigation today, the flux correction method uses a node-centered finite volume method as a starting point to which truncation error-canceling terms are added to increase accuracy. Second, the method requires no additional flux quadrature or second derivatives in the solution reconstruction like quadratic finite volume schemes [13–15]. Two-dimensional studies indicate the resulting scheme is nearly fourth-order accurate and requires minimal computational overhead beyond second-order schemes [10]. Chapters 2 and 3 give in-depth detail on the method.

### 1.3 Verification and Validation

In this work, detailed verification and validation is employed, and as such, it is necessary define them in the context of CFD.

#### 1.3.1 Verification

Verification is used in CFD to ensure computational accuracy and to test algorithm integrity. Verification in this sense may be defined as:

*The process of determining that a model’s implementation accurately represents the developer’s conceptual description of the model and the solution to the model [16].*

Verification of a solver is commonly ascertained through the method of manufactured solutions (MMS) [17–19], ensuring the solver is free from discretization errors, and supplying an order of accuracy of the solution.

Complex problems may be verified through MMS, as demonstrated by Diskin et al. [20–22], where different second-order schemes were tested extensively, as well the study on the effects of irregular grids have on accuracy. The interior solution has generally been the focus of MMS methodologies [23–28], however Folkner and Katz [29] performed a unique investigation of this methodology on boundary conditions.

Roache [17] describes a general MMS provides a general procedure for working with such analytical solutions. The procedure is very simple. A continuum solution is constructed, which in general will not satisfy the governing equations. An appropriate source term can be determined to cancel any imbalance in the PDEs caused by the choice of the continuum solution. The solution also defines the boundary conditions in all forms, be they Dirichlet, Neumann or Robin. The chosen solution need not have a physical meaning since verification (of codes or of calculations) is a purely mathematical exercise. But choosing a physically realistic manufactured problem which has a closed form solution offers a useful advantage: It exercises each term involved in the PDEs in a manner similar to that of a real problem so that similar difficulties in the solution and error estimation processes will arise.

Once the manufactured solution has been constructed and the source terms determined for the set of equations to be verified, code verification can take place on any grid in the domain covered by the MMS. By verifying the code on increasing grid resolutions, we may show that as grid resolution is increased, the solution becomes more accurate.

### 1.3.2 Validation

Validation of a solver must take place once verification has been performed. Validation of an entire CFD code for all scenarios is not possible. Specific problems where previous data exists may be tested. In validating a CFD code we aim to assess performance on specified problems. One validates a model or simulation. Applying the code to flows beyond the region of validity is therefore termed prediction. A more precise definition of validation in this scenario is given as:

*The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model [16].*

Validation examines if the conceptual models, computational models as implemented into the CFD code, and computational simulation agree with real world observations. The strategy is to identify and quantify error and uncertainty through comparison of simulation

results with experimental data. The experiment data sets themselves will contain bias errors and random errors which must be properly quantified and documented as part of the data set. The accuracy required in the validation activities is dependent on the application, and so, the validation should be flexible to allow various levels of accuracy.

#### 1.4 Outline of Thesis

This thesis is organized into a multi-paper format, where each of the following chapters corresponds to a paper that has either been published, or has been presented at a conference and been published as part of a conference proceedings. The paper references that correspond to each chapter have been provided next to the chapter number.

The thesis is outlined as follows: Chapter 2 [30] details the verification and validation for a two-dimensional flux correction solver for turbulent flows. Next, extensions to three-dimensions are outlined in Chapter 3 [31]. Presented are a number of verification and validation case studies. In Chapter 4 [32], where we investigate limiting techniques for shock-turbulence interactions with the flux correction scheme on strand grids. A number of validation cases are considered. In Chapter 5 [33], we extend the flux correction scheme to use the Menter SST RANS turbulence model for turbulence closure, presenting verification and validation of the scheme. Methods for complex geometry handling and improved strand grid automation are investigated in Chapter 6 [34]. Finally, in Chapter 7, concluding remarks are made, along with details of future directions for work.

**REFERENCES**

- [1] Alhubail, M. and Wang, Q., “The swept rule for breaking the latency barrier in time advancing PDEs,” *Journal of Computational Physics*, 2015.
- [2] Wissink, A., “Helios Solver Developments Including Strand Meshes,” Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.
- [3] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [4] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [5] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA paper* 2009-3792, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [6] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.
- [7] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [8] Lee, Y.-L. and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [9] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.

- [10] 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.
- [11] 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.
- [12] 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.
- [13] Barth, T. J. and Frederickson, P., “Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction,” *AIAA paper* 1990-0013, AIAA 28th Aerospace Sciences Meeting, Reno, NV, January 1990.
- [14] Delanaye, M. and Liu, Y., “Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids,” *AIAA paper* 1995-3259, AIAA 14th CFD Conference, Norfolk, June 1999.
- [15] Ollivier-Gooch, C., Nejat, A., and Michalak, K., “On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes,” *AIAA paper* 2007-4464, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [16] AIAA, “Guide for the Verification and Validation of Computational Fluid Dynamics Simulations,” Tech. rep., AIAA Guide, 1998.
- [17] Roache, P., “Code Verification by the Method of Manufactured Solutions,” *Transactions of the ASME*, Vol. 124, 2002, pp. 4–10.
- [18] Roy, C., Nelson, C., Smith, T., and Ober, C., “Verification of Euler/Navier-Stokes Codes Using the Method of Manufactured Solutions,” *International Journal for Numerical Methods in Fluids*, Vol. 44, 2004, pp. 599–620.

- [19] Veluri, S., Roy, C., and Luke, E., “Comprehensive Code Verification for an Unstructured Finite Volume CFD Code,” AIAA paper 2010-127, AIAA 48th Aerospace Sciences Meeting, Orlando, FL, January 2010.
- [20] Diskin, B. and Thomas, J., “Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes,” *AIAA paper* 2010-1079, AIAA 48th ASM, Orlando, January 2010.
- [21] Diskin, B., Thomas, J., Nielsen, E., Nishikawa, H., and White, J., “Comparison of node-centered and cell-centered unstructured finite-volume discretizations. Part I: viscous fluxes,” AIAA Paper 2009-0597, 2009.
- [22] Diskin, B. and Thomas, J., “Effects of Mesh Regularity on Accuracy of Finite-Volume Schemes,” AIAA paper 2012-0609, AIAA 50th Aerospace Sciences Meeting, Nashville, TN, January 2012.
- [23] Eriksson, S. and Nordström, J., “Analysis of mesh and boundary effects on the accuracy of node-centered finite volume schemes,” AIAA Paper 2009-3651, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [24] Katz, A. and Sankaran, V., “Mesh Quality Effects on the Accuracy of Euler and Navier-Stokes Solutions on Unstructured Meshes,” Tech. rep., 6th International Conference on Computational Fluid Dynamics, St. Petersburg, Russia, July 2010.
- [25] Diskin, B. and Thomas, J., “Accuracy Analysis for Mixed-Element Finite-Volume Discretization Schemes,” *NIA Report* 2007-08, National Institute of Aerospace, 2007.
- [26] Roy, C., “Review of Code and Solution Verification Procedures for Computational Simulation,” *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.
- [27] Luke, E., Hebert, S., and Thompson, D., “Theoretical and Practical Evaluation of Solver-Specific Mesh Quality,” *AIAA paper* 2008-0934, AIAA 46th ASM, Reno, January 2008.



- [28] Giles, M., “Accuracy of Node-based Solutions on Irregular Meshes,” *Lecture Notes in Physics*, Vol. 323, 1989, pp. 273–277.
- [29] Folkner, D., Katz, A., and Sankaran, V., “Design and Verification Methodology of Boundary Conditions for Finite Volume Schemes,” Tech. rep., 7th International Conference on Computational Fluid Dynamics, Big Island of Hawaii, July 2012.
- [30] Tong, O., Work, D., and Katz, A., “High-Order Methods for Turbulence Using Strand Grids,” Tech. Rep. ICCFD8-0215, 8th International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 2014.
- [31] Tong, O., Katz, A., Yanagita, Y., Casey, A., and Schaap, R., “High-Order Methods for Three-Dimensional Strand Grids,” *Journal of Scientific Computing*, Vol. 64, 2015.
- [32] 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 CFD Conference, Dallas, TX, June 2015.
- [33] Tong, O., Blakely, C., Schaap, R., and Katz, A., “Assessment of a Two-Equation Turbulence Model in the High-Order Flux Correction Scheme,” *AIAA paper 2016-1582*, AIAA 51th Aerospace Sciences Meeting, San Diego, CA, January 2016.
- [34] Tong, O., Yanagita, Y., Harris, S., Work, D., and Katz, A., “Asymptotic Geometry Representation for Complex Configurations on Strand Grids,” *AIAA paper 2016-1584*, AIAA 51th Aerospace Sciences Meeting, San Diego, CA, January 2016.

CHAPTER 2  
VERIFICATION AND VALIDATION OF A HIGH-ORDER STRAND GRID METHOD  
FOR TWO-DIMENSIONAL TURBULENT FLOWS

### 2.1 Abstract

In this paper, we construct a novel hybrid of two one-dimensional schemes in order to leverage several advantages for solving two-dimensional turbulent flows. Building upon previous work by the authors and others, we combine one-dimensional flux correction along body surfaces along with high-order summation-by-parts finite differences normal to surfaces. A new semi-implicit multigrid solution method is presented that capitalizes on the unique directional properties of each scheme, utilizing an explicit multigrid scheme along the surface direction, and an implicit Gauss-Seidel scheme along the strand direction. Turbulence closure is achieved with a robust version of the Spalart-Allmaras turbulence model that accommodates negative values of the turbulence working variable. The hybrid scheme exhibits fourth-order convergence using the method of manufactured solutions. Fundamental validation studies of the turbulent flux correction method are conducted in two dimensions, using the NASA-Langley turbulence resource as a means for comparison. Results are presented that demonstrate improvements in accuracy with minimal computational and algorithmic overhead over traditional second-order algorithms.

### 2.2 Introduction

A present challenge in computational fluid dynamics (CFD) today is computing high-fidelity solutions for high-Reynolds number turbulent flows over complex geometry. This on-going challenge may be attributed to several sources. First, meshing tasks often require a disproportionate amount of time to configure quality viscous meshes for complex configurations compared to computational time. Complex multi-body viscous meshes, such as

those employed for rotorcraft, can require days or weeks of set up time before the computational process even begins. A need for increased mesh automation is evident, otherwise the percentage of time devoted to the meshing process will only increase relative to the total computation time. Second, many CFD practitioners rely on the use of traditional numerically diffuse second-order schemes, and thus observe limited accuracy on unstructured meshes. High-order methods show potential to resolve this issue, however high-order methods for unstructured grids are generally not at a production level. Third, poor scalability becomes an issue with ever-increasing mesh count for complex systems. Automating viscous mesh generation, preserving spatial and temporal accuracy, and maintaining computational efficiency are currently among the greatest research challenges in CFD today.

In this paper, we take the approach that a single scheme or method is unlikely to address all these challenges simultaneously. Instead, we address these diverse challenges through a carefully constructed hybrid scheme which leverages the automation of strand grids, the efficiency of a high-order finite volume flux correction scheme, and the proven stability properties of summation-by-parts methods near boundaries.

First, the strand approach has shown potential to alleviate certain meshing and scaling difficulties [1–4], allowing the possibility of fully automatic volume grid generation while providing a compact and scalable [5–8] mesh topology for self-satisfying overset domain con-

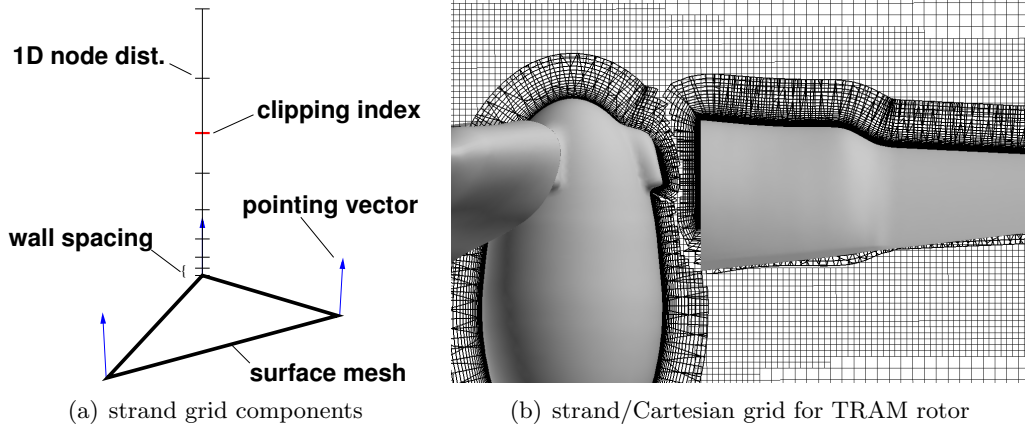


Fig. 2.1: Strand grid elements and example strand/Cartesian grid system for the TRAM rotor.

nectivity. Near solid bodies, the strand approach automatically provides 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 6.1(a). Away from solid bodies, adaptive Cartesian grids (not investigated specifically in this work) resolve vortical shedding and wake features with efficient high-order algorithms, shown in Figure 6.1(b).

Second, the flux correction (FC) finite volume methodology of Katz and Sankaran [9,10] is incorporated in order to efficiently maintain high-order accuracy in the presence of complex geometry. Previous work on turbulent flow simulations using the strand-Cartesian methodology has focused on traditional second-order finite volume methodology [11]. While this approach generally yields results that fall within the range of established codes and experiments, in this paper we investigate the potential advantages of a high-order accurate formulation for turbulent flows. The focus on high-order accuracy comes as a result of the recent findings of Wissink et al. [12] which demonstrate that even highly accurate high-order off-body solution methods can easily be spoiled by excessive near-body discretization error. As is characteristic of the FC scheme, the present method requires no additional flux quadrature or second derivatives in the solution reconstruction like quadratic finite volume schemes [13–15]. The method is also extensible to multi-dimensions, although that is not the focus of the present paper.

Third, summation-by-parts (SBP) operators, first introduced by Kreiss and Scherer [16] and further investigated by many others [17–21], provide stability and robustness near boundaries, while supporting high-order accuracy for smooth boundary layer-capturing strand distributions.

While none of these individual methods is new to this paper, the strand-FC-SBP combination represents a targeted approach that leverages the advantages of each method. In addition to the spatial discretization, we extend the hybrid nature of the methodology to the solution method, which uses a combination of explicit multigrid for FC terms and implicit Gauss-Seidel for SBP terms along strands. As such, the method only requires diagonal

block-Jacobians consisting of contributions from structured strand stencils, which are easily obtained via the inherent grid structure. The resulting scheme displays fourth-order accuracy, compares favorably to established methods for turbulent flows, and requires minimal computational overhead beyond second-order schemes.

The paper is outlined as follows: First, we briefly review the high-order strand grid discretization scheme of Katz and Work [22], focusing on extensions and modifications for turbulent flows in two dimensions. Next, we introduce a new semi-implicit multigrid procedure used to solve the discretization in the presence of high aspect ratio grids needed for turbulent flows. We then present case studies and results which highlight the advantages of the high-order method. Finally we conclude and discuss potential future work.

### 2.3 High-Order Hybrid Discretization

In this work, we solve the compressible Reynolds-Averaged Navier-Stokes (RANS) equations in two-dimensions. The “negative” Spalart-Allmaras model [23], which admits negative values of the turbulence working variable for robustness is used to achieve turbulence closure. The combined RANS-SA equations are expressed as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S, \quad (2.1)$$

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

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho \tilde{\nu} \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ \rho h u_j \\ \rho \tilde{\nu} u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ \frac{\eta}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_j} \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathcal{P} - \mathcal{D} + C_{b2} \rho \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \end{pmatrix}. \quad (2.2)$$

The turbulent source term for the SA model consists of a production term,  $\mathcal{P}$ , and a destruction term,  $\mathcal{D}$ . Here,  $\rho$  is the density,  $u_j$  is the  $j^{\text{th}}$  component of the fluid velocity,  $p$  is

the pressure,  $e$  is the total energy per unit mass,  $h \equiv e + p/\rho$  is the total enthalpy per unit mass,  $\tilde{\nu}$  is the turbulence working variable,  $\sigma_{ij}$  is the deviatoric stress tensor,  $q_j$  is the  $j^{\text{th}}$  component of the heat flux vector, and  $\frac{\eta}{\sigma}$  is the turbulent diffusion coefficient. The stress tensor is defined as

$$\sigma_{ij} = 2(\mu + \mu_T) \left( S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (2.3)$$

where  $\mu$  is the dynamic viscosity,  $\mu_T$  is the turbulent viscosity, and  $S_{ij}$  is the rate of strain tensor, defined as

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.4)$$

The heat flux vector is obtained with Fourier's Law, and defined as

$$q_j = -C_p \left( \frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \frac{\partial T}{\partial x_j}, \quad (2.5)$$

where  $T$  is temperature,  $C_p$  is the specific heat,  $Pr$  is the Prandtl number, and  $Pr_T$  is the turbulent Prandtl number. In addition, Sutherland's Law is utilized to relate viscosity and temperature, and the ideal gas equation of state is used.

The “negative” SA model is designed to use the standard SA model when the turbulent working variable,  $\tilde{\nu}$  is positive, and add modifications to the standard model when the turbulent working variable is negative. The details of the standard SA model, including the well-known definitions of the production and destruction terms, may be found in the original work by Spalart and Allmaras [24]. Modifications to the model to accommodate negative values of the turbulence working variable have been recently suggested by Allmaras [23], and are employed in this work. Negative values of  $\tilde{\nu}$  are potentially encountered on under-resolved grids, and at the edge of boundary layers and wakes.

We now explain the grid topology constructed to solve the RANS equations. Strand grids consisting of an unstructured surface tessellation are extruded along straight lines (strands) away from solid bodies in the physical domain and are locally mapped to a uniform computational domain, as shown in Figure 5.2. The surface itself, lying in the  $s$ -direction, is described by high-order one-dimensional “surface-elements,” which are shown as the bold

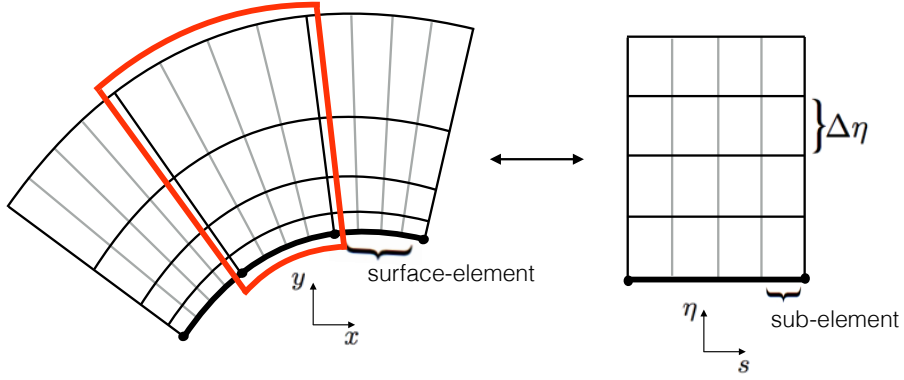


Fig. 2.2: Two-dimensional mapping of strand stack from physical space to computational space.

line in the figure. Quartic surface-elements are used for all applications in this work, and are curved to smoothly capture geometry [25, 26]. Each high-order surface-element is subdivided equally into a number of “sub-elements,” which for purposes of a finite volume flux balance, may be considered as linear line segments. To demonstrate this, a singular surface-element has been highlighted with a red box in Figure 5.2, and its mapping to computational space shown. A smoothly stretched distribution of nodes along each strand in the physical space is mapped to an equally spaced distribution in the  $\eta$ -direction in the computational space, where  $\eta \in (0, 1)$ . In the computational space, the strand spacing is  $\Delta\eta = 1/(N - 1)$ , where  $j = 1, \dots, N$  is the strand node numbering beginning with the node on the surface. The combination of smoothly stretched strand distributions, high-order surface-elements, and linear sub-elements, enable a novel discretization strategy, discussed below.

Upon transformation to the computational space, Equation 5.1 in two dimensions becomes

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial s} + \frac{\partial \hat{G}}{\partial \eta} - \frac{\partial \hat{F}^v}{\partial s} - \frac{\partial \hat{G}^v}{\partial \eta} = \hat{S}, \quad (2.6)$$

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

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

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

$$\begin{pmatrix} s_x & \eta_x \\ s_y & \eta_y \end{pmatrix} = \frac{1}{J} \begin{pmatrix} y_\eta & -y_s \\ -x_\eta & x_s \end{pmatrix},$$

$$J = x_s y_\eta - y_s x_\eta.$$

Here,  $J$  is the Jacobian of the transformation, and  $(\hat{F}, \hat{G})$  and  $(\hat{F}^v, \hat{G}^v)$  are the transformed inviscid and viscous fluxes, where partial differentiation is denoted with a subscript (e.g.  $\partial x / \partial s = x_s$ ).

The unique aspect of the present scheme is the novel manner in which the discretization is carried out and solved in the  $s$ - and  $\eta$ -directions independently to obtain high-order accuracy. The general strategy is to perform a high-order finite-volume flux balance in the  $s$ -direction, known as “flux correction,” and to use high-order finite-differences and penalties based on SBP operators in the  $\eta$ -direction. In previous work [22] it was shown that treating the  $\eta$ -derivatives in Equation 5.8 with a particular source term discretization preserves the accuracy of the flux correction procedure in the  $s$ -direction. Applying this idea in the present context, the  $\eta$ -derivatives and physical time derivative are moved to the right-hand side and treated as source terms:

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial s} - \frac{\partial \hat{F}^v}{\partial s} = \tilde{S}, \quad \tilde{S} \equiv \hat{S} - \frac{\partial \hat{Q}}{\partial t} - \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{G}^v}{\partial \eta}. \quad (2.7)$$

Note that a pseudo-time derivative is added on the left-hand side of Equation 5.9 to facilitate the time-marching solution procedure to be described later.

Examining Equation 5.9, we now must solve a series of one-dimensional conservation laws in the  $s$ -direction with a source term. In order to accomplish this, we borrow directly from the FC methodology of Katz and Sankaran [9], which constructs numerical fluxes at node  $i$  of a given surface-element as

$$\frac{\partial \hat{F}}{\partial s} \approx \frac{1}{\Delta s} \left( \hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}} \right), \quad \hat{F}_{i+\frac{1}{2}} = \frac{1}{2} \left( \hat{F}_L + \hat{F}_R \right) - \frac{1}{2} |\hat{A}| (Q_R - Q_L), \quad (2.8)$$



where left and right fluxes and states are defined as

$$\begin{aligned}\hat{F}_L &= \hat{F}_i + \frac{1}{2}\Delta s \hat{F}_{s,i}^h, & \hat{F}_R &= \hat{F}_{i+1} - \frac{1}{2}\Delta s \hat{F}_{s,i+1}^h, \\ Q_L &= Q_i + \frac{1}{2}\Delta s Q_{s,i}^h, & Q_R &= Q_{i+1} - \frac{1}{2}\Delta s Q_{s,i+1}^h.\end{aligned}\tag{2.9}$$

Here,  $\hat{F}_s^h$  and  $Q_s^h$  are numerical approximations to the flux and solution derivatives in the  $s$ -direction, respectively,  $\Delta s$  is the (uniform) spacing of  $s$  of sub-elements in the computational domain, and  $|\hat{A}|$  is the flux Jacobian computed via the method of Roe [27]. Note that the FC methodology requires direct reconstruction of the non-linear flux,  $\hat{F}$ .

The above numerical fluxes operate on linear sub-elements to form a finite-volume flux balance at each node. The high-order surface-elements enter into the formulation in the construction of the derivative approximations for  $\hat{F}_s^h$  and  $Q_s^h$  which, according to Katz and Sankaran [9], must be computed to second-order or better. These approximations are formed by considering a finite element-like interpolation of  $\hat{F}$  and  $Q$  in each element, which for  $Q$  is

$$Q^h(s) = \sum_m l_m(s) Q_m, \tag{2.10}$$

where  $l_m(s)$  is the Lagrange polynomial at the  $m^{\text{th}}$  node in a given high-order surface-element. The approximate gradient,  $Q_s^h$ , is then

$$Q_s^h = \sum_m \frac{\partial l_m(s)}{\partial s} Q_m.$$

At nodes shared by adjacent surface-elements, the derivative approximations are averaged to achieve improved stencil centering. The computation of the flux derivative,  $\hat{F}_s^h$ , proceeds in the same way.

Recently, Pincock and Katz [10] extended the original FC scheme to include the viscous terms in the Navier-Stokes equations by simply *not averaging* gradients that appear in the viscous fluxes. Additionally, no artificial dissipation term is added to the numerical viscous

flux. In this manner, the viscous flux derivative is approximated with

$$\frac{\partial \hat{F}^v}{\partial s} \approx \frac{1}{\Delta s} \left( \hat{F}_{i+\frac{1}{2}}^v - \hat{F}_{i-\frac{1}{2}}^v \right), \quad \hat{F}_{i+\frac{1}{2}}^v = \frac{1}{2} \left( \hat{F}_L^v + \hat{F}_R^v \right)$$

where left and right viscous fluxes are defined as

$$\hat{F}_L^v = \hat{F}_i^v + \frac{1}{2} \Delta s \hat{F}_{s,i}^{vh}, \quad \hat{F}_R^v = \hat{F}_{i+1}^v - \frac{1}{2} \Delta s \hat{F}_{s,i+1}^{vh}.$$

Again, no averaging between adjacent elements is required for the viscous flux derivatives, nor where gradients of  $Q$  are needed to form the viscous fluxes themselves.

By treating the flux derivatives in the  $s$ -direction in the above manner, we expect to obtain at least third-order discretization accuracy. However, this will only be true if each term in  $\tilde{S}$  in Equation 5.9 is likewise computed to at least second-order accuracy in terms of the truncation error. Examining equation 5.9,  $\tilde{S}$  contains  $\eta$ -derivatives of the flux, which can be computed to high-order accuracy via the now standard SBP/SAT-treatment [16–19, 28–30]. In this work, we utilize specific operators for the first- and second-derivatives from Fernandez and Zingg [21] and Mattsson [20]. The  $\eta$ -derivatives of the flux in  $\tilde{S}$  are expressed as

$$\frac{\partial \hat{G}}{\partial \eta} \approx D_\eta \hat{G}, \quad \frac{\partial \hat{G}^v}{\partial \eta} \approx D_\eta (B^s Q_s^v) + D_{2\eta} (B^\eta) Q^v.$$

Here,  $D_\eta$  and  $D_{2\eta}$  are discrete first- and second-derivative operators designed for stability in the energy norm, and are coupled with consistent penalty boundary conditions at the endpoints of the strands. Note that the viscous flux is first decomposed as  $\hat{G}^v = B^s Q_s^v + B^\eta Q_\eta^v$ , where the  $B$  matrices contain non-constant coefficients (eg. viscosity), and  $Q^v = (u, v, T)$  consists of quantities appearing as derivatives in the viscous flux. The mixed-derivative is treated with the conventional  $D_\eta$  operator, while the pure  $\eta$ -derivative is treated with the special  $D_{2\eta}$  operator for the second-derivative with variable coefficients. Further details of  $D_{2\eta}$  may be found in the work by Mattsson [20].

At this point, we wish to highlight the eclectic nature of the spatial discretization scheme just described. We borrow pieces from finite volume (FC), finite element (gradient

approximations), and finite difference (SBP) methodology to create an approach that hopefully contains the advantages of each, including complex geometry handling, high-order accuracy, stability, and simplicity of implementation. We implement all these methods within a strand grid approach to take advantage of automatic mesh generation and scalable infrastructures.

#### 2.4 Semi-Implicit Multigrid Solution Method

In this section, we extend our eclectic approach to the solution method of Equation 5.9. One advantage of the flux correction scheme is that it is based upon finite volume methodology, for which numerous mature solution techniques already exist. While the related method of Katz and Work [22] was limited to explicit solution techniques appropriate for inviscid and laminar flows, it is unlikely that such techniques are optimal in the present context for turbulent flows which require very high aspect ratio cells. In this work we seek to take advantage of the best qualities of several solution techniques to achieve optimal efficiency in terms of memory and computation time for such flows. Specifically, we combine a non-linear LUSGS [31–33] implicit scheme in the strand direction with an explicit Runge-Kutta method [34] in each unstructured layer of the strand grid. The entire procedure is wrapped in an FAS multigrid algorithm [35]. In this way, stiffness due to high aspect ratio cells needed for turbulent boundary layers is relieved via the implicit scheme, while maintaining simplicity, robustness, and scalability in the more isotropic unstructured layers of the grid via the Runge-Kutta algorithm and multigrid.

The solution algorithm proceeds by considering the contributions to the residual at surface index  $i$  and strand index  $j$ , expressed as,

$$R_{i,j} = R_{i,j}(Q_{\in i}, Q_{\in j}), \quad (2.11)$$

where  $Q_{\in i}$  represents contributions from nodes in the  $s$ -direction at level  $j$ , and  $Q_{\in j}$  represents contributions from nodes along the strand passing through node  $i$ . Following a non-linear LUSGS procedure along strands, contributions from layers above and below layer  $j$

are treated implicitly in a Gauss-Seidel procedure. Sweeps are performed by advancing each layer in the strand grid up and down, using the latest available data on the right hand side, and maintaining left-hand side contributions from the currently layer only. The result is a block diagonal equation, each line of which reads,

$$D_{i,j}(Q^{k+1} - Q^k) + R_{i,j}(Q_{\in i}^k, Q_{\in j}^*) = 0. \quad (2.12)$$

Note that the nodes in the current layer are treated explicitly at the current pseudo-time station,  $k$ , while the layers above and below the current layer are treated at the latest available station,  $*$ . The block diagonal consists of Jacobian terms of a lower order discretization taken with respect to the current layer nodes only, while contributions from nodes in adjacent layers are promoted to high-order and moved to the right-hand side in the form of residual evaluations. The resulting block diagonal at each node is,

$$D_{i,j} = \frac{1}{2} (|B_{i,j-1/2}| + |B_{i,j+1/2}|) + \frac{\partial}{\partial Q_{i,j}} (D_{2\eta} (B^\eta) Q^p), \quad (2.13)$$

where  $B = \partial \hat{G} / \partial Q$  is the Jacobian of the inviscid flux in the strand direction, and  $D_{2\eta}$  is the SBP operator for the second derivative of the viscous variables,  $Q^v = (u, v, T)$ , with variable coefficient matrix,  $B^\eta$ , representing the discretization of the viscous fluxes. Extensive numerical experiments have shown that it is sufficient to treat the inviscid fluxes in the strand direction with first-order accuracy for obtaining the contributions to the block diagonal, while the viscous contributions are maintained high-order. Note that mixed-derivative viscous terms are not included on the left-hand side.

To avoid the need for complex and expensive linearization in each unstructured  $s$ -line, an explicit Runge-Kutta algorithm is used. While the pseudo-time step size is limited by the use of the explicit scheme, the limitation is based solely upon grid spacing in each unstructured line. Because the grid spacing in each unstructured line is relatively isotropic compared with the highly stretched spacing along strands, the procedure remains quite efficient for cases tested so far. Following a standard  $n_s$ -stage Runge-Kutta scheme of

Jameson, updates in each layer of the strand grid may be computed with,

$$V_{i,j} \frac{Q^{k+1,m} - Q^k}{\alpha_m \Delta \tau} + D_{i,j}(Q^{k+1,m} - Q^k) + R_{i,j}(Q_{\in i}^{k+1,m-1}, Q_{\in j}^*) = 0, \quad (2.14)$$

leading to a block diagonal equation at each node of the form,

$$\left[ \frac{V_{i,j}}{\alpha_m \Delta \tau} I + D_{i,j} \right] (Q^{k+1,m} - Q^k) = -R_{i,j}(Q_{\in i}^{k+1,m-1}, Q_{\in j}^*), \quad (2.15)$$

Here,  $k$  is the pseudo-time counter,  $m$  is the stage counter,  $\alpha_m$  is the Runge-Kutta coefficient for stage  $m$ , and  $V_{i,j} = \Delta s J_{i,j}$  is the “volume” associated with node  $i, j$ . Before the updates are applied, they are smoothed with an implicit residual smoothing operation [36]. The residuals are smoothed in the  $s$ -direction with approximately two Jacobi iterations. Once the  $k + 1$  station has been computed in layer  $j$  using a Runge-Kutta step, the  $j + 1$  layer is updated, followed by  $j + 2$ , all the way up the strand grid, and back down again. The residual in each layer is computed using the latest available data from layers above and below.

In this work, we use the surface-element data structure to form coarse levels for multi-grid. This alleviates the need for agglomeration procedures and results in coarse surface-elements that are perfectly nested. This simplifies the code and allows for optimal reuse of subroutines for coarse and fine levels. Additionally, it facilitates discretization of the viscous terms on coarse levels. Using fourth order elements, we typically form two coarse levels - the first with quadratic elements, and the second with linear elements. This results in ideal coarsening on each level. The multigrid algorithm provides good convergence acceleration for the cases tested in this work.

Once coarse levels are obtained, restriction and prolongation operations are performed by interpolating solutions, residuals, and corrections using Lagrange basis polynomials over each element. Such an element representation is already available to us because we use a Lagrange basis to compute flux and solution gradients, as described in Equation 2.10. This allows for more accurate transfer operations than conventional agglomeration procedures,

which often rely on low-order averaging or injection [37]. Multigrid forcing terms, formed from the difference of the coarse level residual and agglomerated fine level residuals are added on coarse levels in the standard fashion.

## 2.5 Results

The ability of the strand-FC-SBP method to accurately compute two-dimensional high Reynolds number turbulent flows with strand grids is investigated in this section. Here, we examine fundamental cases for verification and validation purposes, with extensive use of the NASA-Langley turbulence modeling resource [38]. All cases examined use strand grids only. Cases requiring an off-body overset Cartesian mesh are left as future work. Two independent compressible CFD codes are used to aid validation: CFL3D and FUN3D. When available, experimental data is also used for comparison.

### 2.5.1 Verification Studies with the Method of Manufactured Solutions

First, to ensure that algorithmic and/or coding errors in the flux correction solver are minimized, fundamental grid refinement studies are employed. Verification is performed using the method of manufactured solutions [39, 40]. Five meshes of increasing refinement are constructed for the grid refinement procedure. The meshes contain 2,080 nodes, 8,256 nodes, 32,896 nodes, 131,328 nodes and 524,800 nodes on a square shape, placed irregularly along the bottom of the square, as shown in Figure 2.3(a). Nodes along strands are distributed with a hyperbolic tangent stretching function to simulate a boundary layer mesh. The manufactured solution is chosen based on smooth trigonometric functions similar to previous work [41], which varies by approximately 10% in amplitude over the domain with respect to reference values. Reference values are chosen to correspond roughly to physically meaningful values of density, pressure, velocity, and  $\tilde{\nu}$ . The manufactured solution for  $\tilde{\nu}$  is shown in Figure 2.3(b).

Three flow cases are examined: inviscid dominated, viscous dominated, and mixed inviscid-viscous (including turbulent source terms) at  $Re = 100,000$ . These flow conditions in turn are examined at three different solution orders: (2,2), (3,3) and (3,4). Here,

Table 2.1: Order of accuracy of inviscid terms using (2,2), (3,3) and (3,4) schemes (x-momentum error).

Mesh Nodes	Scheme (2,2)		Scheme (3,3)		Scheme (3,4)	
	Error	Order	Error	Order	Error	Order
2080	1.41e-6	-	1.33e-7	-	3.46e-8	-
8256	3.12e-7	2.135	1.59e-8	3.175	3.27e-9	3.354
32896	6.70e-8	2.827	1.85e-9	3.547	2.51e-10	4.157
131328	1.59e-8	1.872	2.35e-10	3.184	1.94e-11	4.124
524800	3.25e-9	2.095	2.42e-11	3.616	1.08e-12	4.253

Table 2.2: Order of accuracy of viscous terms using (2,2), (3,3) and (3,4) schemes (x-momentum error).

Mesh Nodes	Scheme (2,2)		Scheme (3,3)		Scheme (3,4)	
	Error	Order	Error	Order	Error	Order
2080	8.79e-7	-	2.00e-8	-	5.65e-9	-
8256	1.19e-7	2.933	1.38e-9	3.925	2.31e-10	4.688
32896	1.60e-8	2.914	6.85e-11	4.371	7.95e-12	4.903
131328	2.01e-9	3.010	1.61e-12	5.432	1.32e-13	5.935
524800	2.65e-10	2.926	6.83e-14	4.571	1.14e-14	6.884

Table 2.3: Order of accuracy of inviscid and viscous terms combined ( $Re = 100,000$ ) using (2,2), (3,3) and (3,4) schemes (x-momentum error).

Mesh Nodes	Scheme (2,2)		Scheme (3,3)		Scheme (3,4)	
	Error	Order	Error	Order	Error	Order
2080	3.05e-6	-	6.15e-7	-	1.69e-7	-
8256	6.84e-7	2.195	7.05e-8	3.176	1.72e-8	3.354
32896	1.65e-7	2.068	6.15e-9	3.548	9.89e-10	4.158
131328	4.11e-8	2.013	6.83e-10	3.185	5.74e-11	4.125
524800	1.06e-8	1.961	5.57e-11	3.623	2.94e-12	4.292

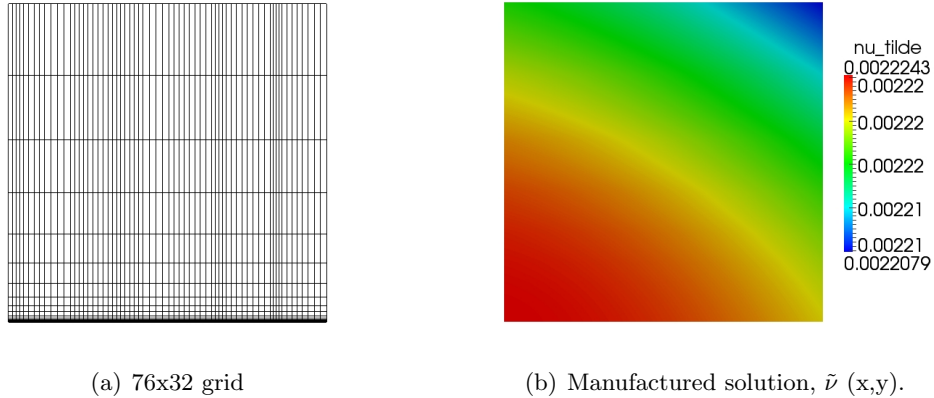


Fig. 2.3: Grid and manufactured solution used for grid refinement study.

order  $(p,q)$  refers to a  $p^{th}$ -order unstructured discretization ( $s$ -direction) and a  $q^{th}$ -order strand discretization ( $\eta$ -direction). For  $p = 3$ , the full flux correction algorithm is used, while for  $p = 2$ , the flux gradient correction terms are omitted, reverting to a conventional finite volume scheme. Additionally, we test schemes with  $q = 2, 3, 4$ . The results of the grid refinement studies for the inviscid dominated, viscous dominated, and combined  $Re = 100,000$  cases are shown in Figure 5.4 and reported in Tables 2.1-2.3. In the figures, the characteristic cell size,  $h$ , is defined as the inverse of number of cells per strand. All solutions are converged to machine zero. Shown clearly in Figure 5.4, the schemes (3,3) and (3,4) deliver between third- and fourth-order accuracy. The scheme (2,2) generally delivers second-order accuracy. While the flux correction scheme is formally third-order, it should be noted that most of the time the scheme (3,4) yields near fourth-order accuracy. This can be attributed to the high-order  $\eta$ -derivatives employed in the interior. Consistent with results from previous work [10, 22], scheme (3,4) produces a lower error than scheme (3,3) due to the higher order derivative approximations in the  $\eta$ -direction. It is interesting to note the improved accuracy using scheme (3,4) for the pure viscous case. Also of interest is the ability of the algorithm to achieve near fourth-order accuracy with the SA turbulence model. This is a consequence of the fully-coupled treatment of the SA model as opposed to the common segregated approach. Additionally, the negative-SA modifications allow us to run the model fully high-order instead of the common approach of reducing the model to



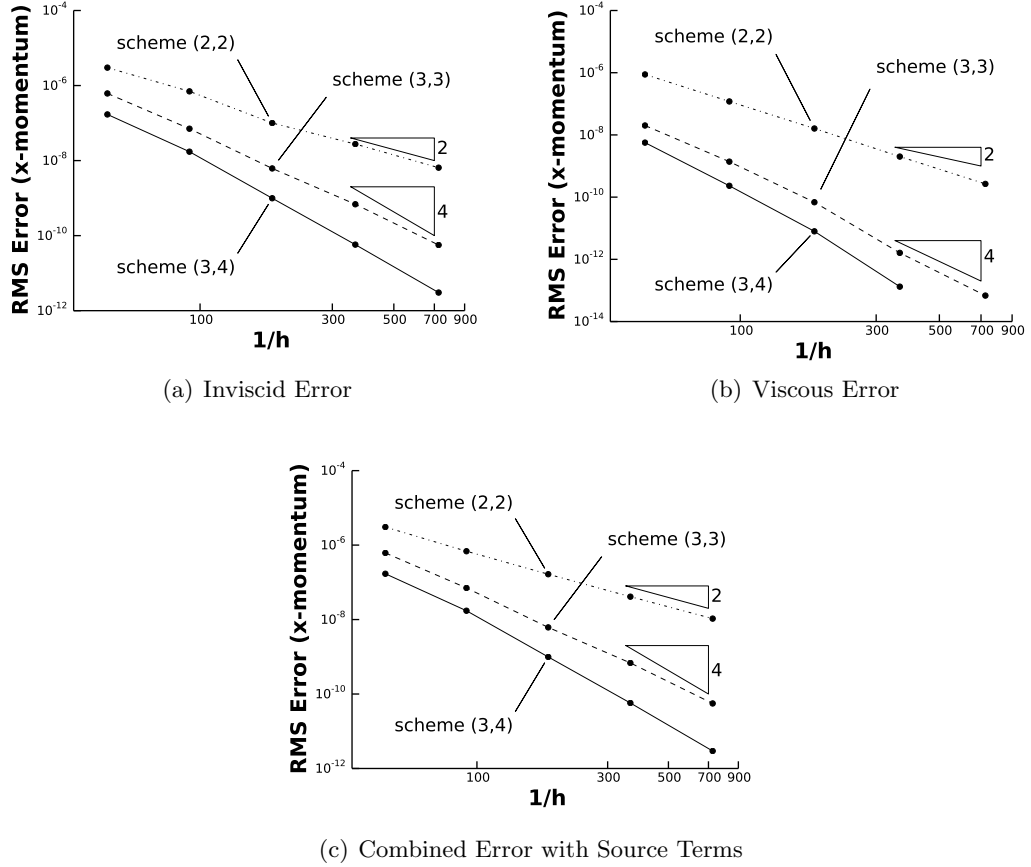


Fig. 2.4: Error convergence at various flow regimes using the method of manufactured solutions with schemes (2,2), (3,3), and (3,4)

first-order.

### 2.5.2 Zero Pressure Gradient Flat Plate

With the implementation of the SA turbulence model verified through MMS, validation of the solver is now performed. The first case we examine is a zero pressure gradient flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ , based on a plate of length unity. Grids from the NASA-Langley turbulence modeling resource are used. However, sub-elements are generated within each parent fourth-order element, thus giving more nodes along the surface than the original grids provide. A grid size of  $341 \times 49$  is used (originally  $69 \times 49$  grid), shown in Figure 5.6(a). The plate leading edge begins at  $x = 0$  and extends for a length of 2. A short inviscid

wall entry way beginning at  $x = -0.33$  is provided to allow for proper inflow conditions. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow. Details of the exact boundary conditions and case set-up may be found on NASA-Langley turbulence resource website [38].

The turbulent viscosity field for this case is shown in Figure 2.6(b), which has been scaled by a factor of 40 vertically to facilitate visualization. Stream-wise velocity and turbulent viscosity profiles are shown for two locations downstream on the plate, and are over-plotted with FUN3D and CFL3D results in Figure 5.8. Good agreement is obtained, even for the  $341 \times 49$  grid, approximately 13 times coarser than the grid ( $545 \times 385$ ) used in the FUN3D and CFL3D results. It should be noted that even the coarsest grid available with quartic elements computed matching normalized velocity and normalized turbulent eddy-viscosity profiles.

It is evident from the figure that the strand FC-SBP solver requires significantly fewer cells to produce a  $C_d$  value that both FUN3D and CFL3D achieve with considerably finer meshes. The computed drag coefficient, which is entirely due to skin friction for this case, is shown in Table 5.2 for the  $341 \times 49$  grid, along with FUN3D and CFL3D results for the same grid. The drag coefficient falls within the range predicted by the established codes.

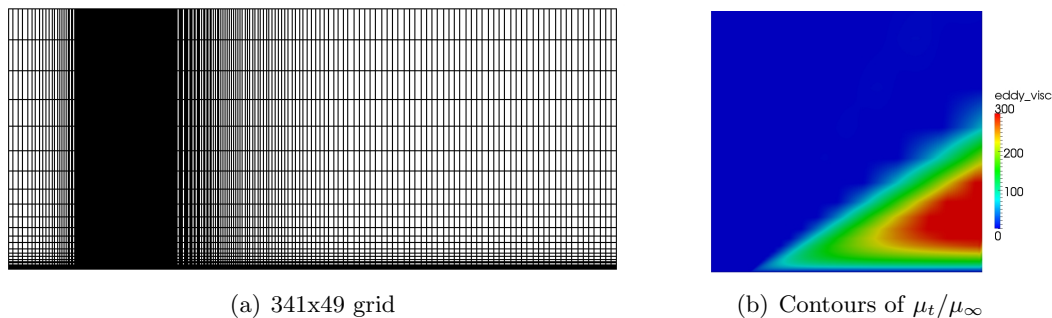


Fig. 2.5: Grid and normalized turbulent eddy-viscosity contours for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .

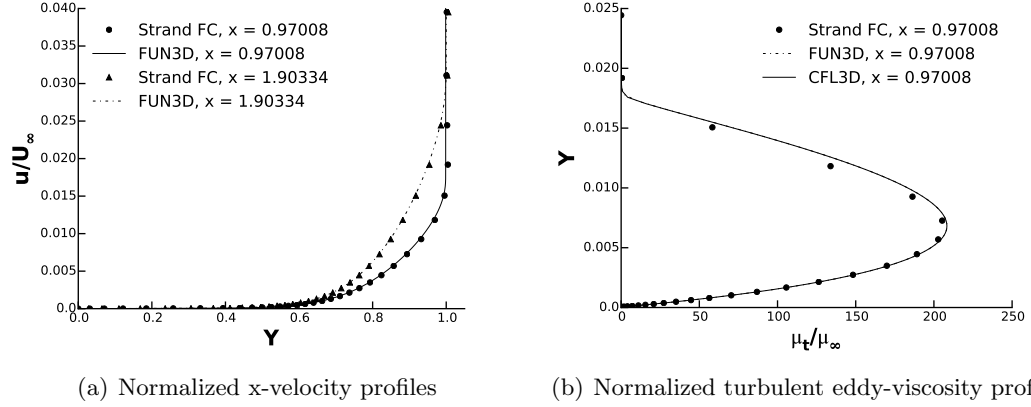


Fig. 2.6: Comparison of normalized stream-wise velocity and normalized turbulent eddy-viscosity profiles for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .

### 2.5.3 Bump-in-Channel

Further validation is performed for turbulent flux correction by way of a bump-in-channel case. This case is conducted at a Mach number of  $M = 0.2$ , at a Reynolds number of  $Re = 3 \times 10^6$  based on a grid length of unity. The body reference length is 1.5 units, where the lower wall is a viscous-wall bump extending from  $x = 0$  to 1.5. The maximum bump height is  $y = 0.05$ . The upstream and downstream farfield extends 25 units from the viscous-wall, with symmetry boundary conditions imposed on the lower wall between the farfield and the solid wall. The upper boundary at  $y = 5.0$  is set to an inviscid plane. A further description and layout of the case may be found on the NASA-Langley turbulence website [38].

Fourth-order elements are used to create a grid with  $705 \times 96$  nodes, shown in Figure 5.10(a), and close up in Figure 5.10(b). Figures 5.10(b) and 5.10(c) are scaled by a factor

Table 2.4: Comparison of computed drag coefficients for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .

Solver	$C_d$
Strand FC	2.85836E-3
FUN3D (quads)	2.85246E-3
FUN3D (triangles)	2.84067E-3
CFL3D	2.85986E-3

of 15 vertically to facilitate visualization. To avoid internal corner issues over the bump, strands are fixed vertically to avoid crossings. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow. The turbulent viscosity field for this case is shown in Figure 5.10(c).

Stream-wise velocity and turbulent viscosity profiles are shown for two locations downstream on the bump, and are over-plotted with FUN3D and CFL3D results in Figure 5.11. Good agreement is obtained for all profiles, even for the  $705 \times 96$  grid, which is approximately 13.5 times more coarse than the FUN3D and CFL3D results shown in the figure. In Figure 5.13, a plot of the surface coefficient of pressure and friction along the bump is shown, and is over plotted with CFL3D and FUN3D results. The computed drag coefficient is shown in Table 5.3. The drag coefficient falls within the range predicted by the established codes. Drag prediction shows good agreement, despite containing skewed high-aspect ratio cells over the critical areas of the bump.

#### 2.5.4 NACA 0012 Airfoil

While the previous cases provide good initial test beds for smooth geometry, ultimately, strand grids must be able to compute high Reynolds number turbulent flows over geometry containing sharp corners. The final case presented provides the challenge of accurately computing flow in the presence of a sharp corner at the trailing edge of a NACA 0012 airfoil. The case consists of flow at  $M = 0.15$  and  $Re = 6 \times 10^6$  at various angles of attack. The grid used is shown in Figure 4.9, with a close up in Figure 2.10(b). The surface mesh consists of 1024 nodes around the airfoil, and 256 nodes along each strand, which extend for 10 chords, resulting in a volume mesh with a total of 262,144 nodes.

Table 2.5: Comparison of computed drag coefficients for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

Solver	$C_d$
Strand FC	3.58300E-3
FUN3D	3.56106E-3
CFL3D	3.57238E-3

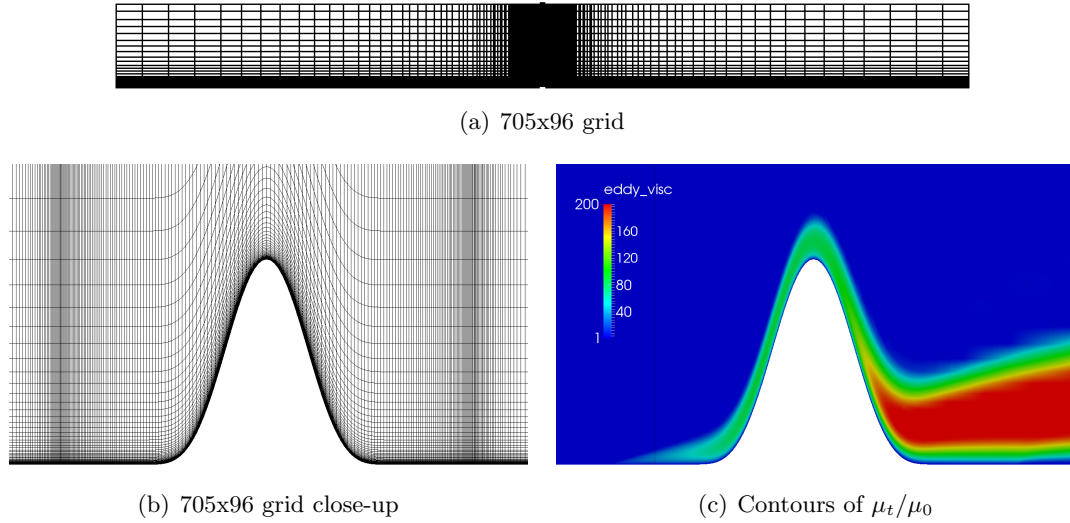


Fig. 2.7: Grid and turbulent viscosity contours for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

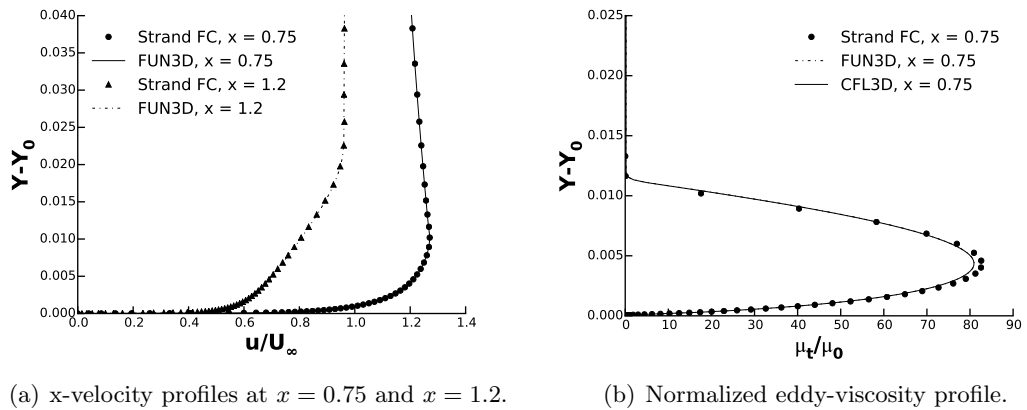


Fig. 2.8: Comparison of stream-wise velocity and turbulent viscosity profiles for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

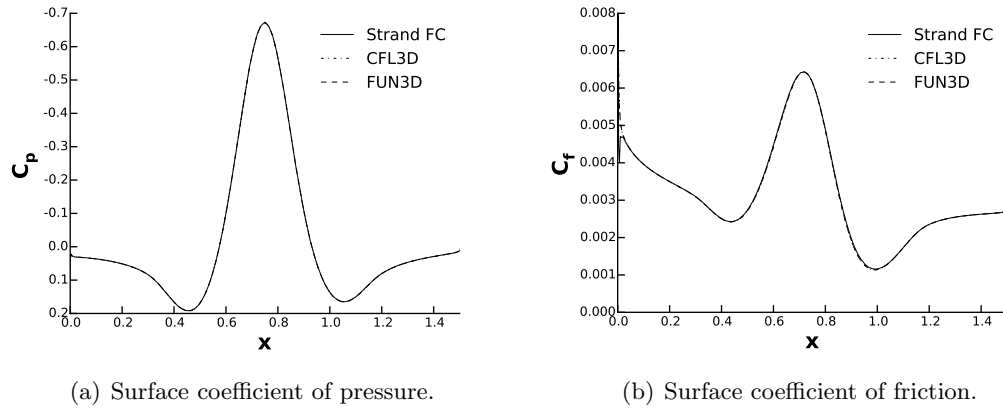
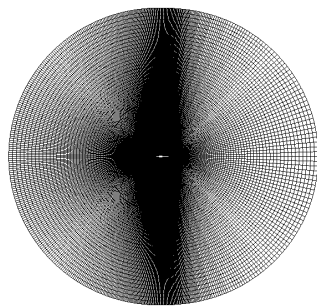
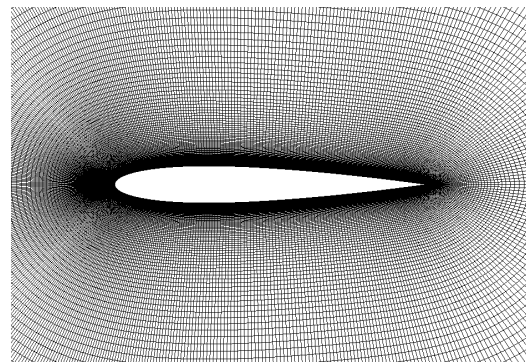


Fig. 2.9: Bump-in-channel surface coefficient of pressure and friction.



(a) NACA 0012 strand grid.



(b) NACA 0012 strand grid close-up.

Fig. 2.10: Grid system for flow over a NACA 0012 airfoil.

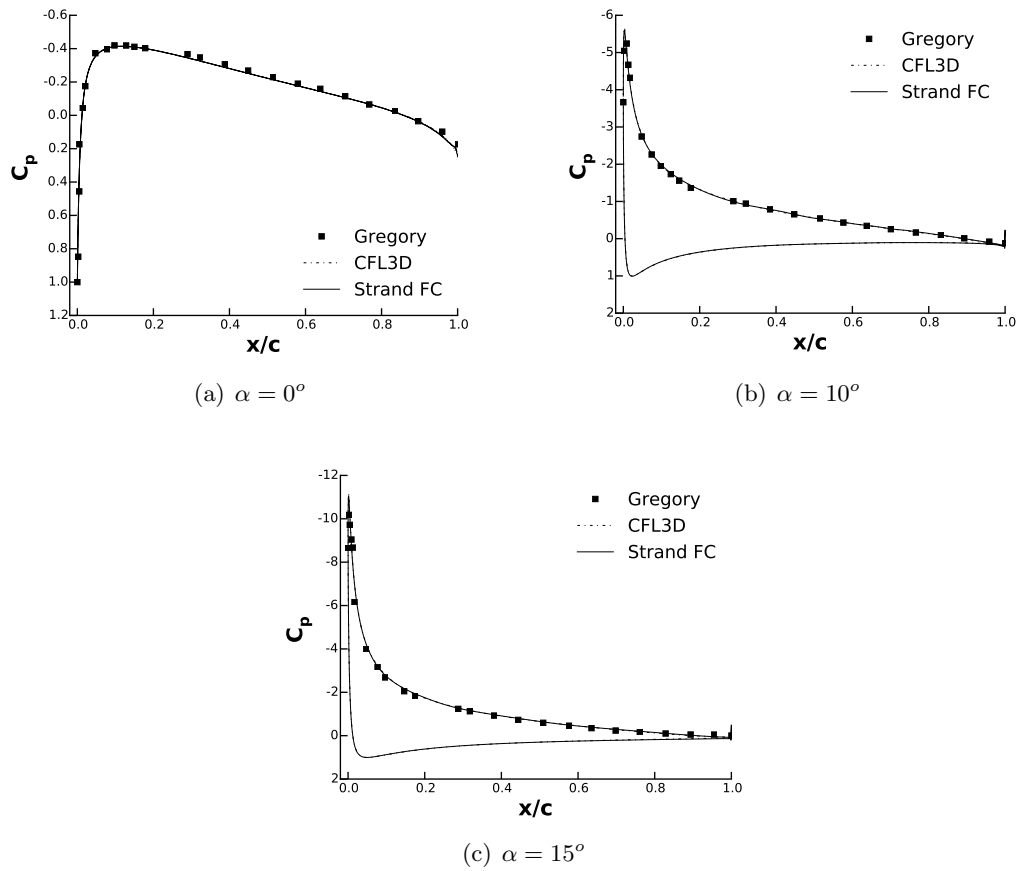


Fig. 2.11: Pressure coefficient for flow over a NACA 0012 airfoil at  $M = 0.15$  and  $Re = 6 \times 10^6$  at various angles of attack.

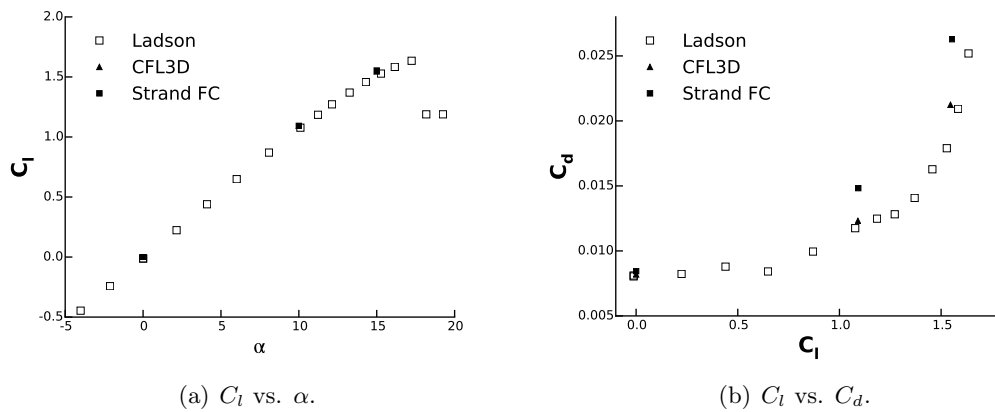


Fig. 2.12:  $C_l$  vs.  $\alpha$  and  $C_l$  vs.  $C_d$  compared to experiment for flow over a NACA 0012 airfoil at  $M = 0.15$  and  $Re = 6 \times 10^6$ .

The surface pressure coefficients for  $\alpha = 0^\circ, 10^\circ, 15^\circ$  are shown in Figure 4.11(a) and compared with the experimental data of Gregory and O’Reilly [42]. 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. Excellent agreement is shown with the flux correction method for all instances observed. Both the FUN3D and CFL3D solver use a very fine two-dimensional C-grid, totaling 274,329 nodes. In contrast, the two-dimensional grid used for this problem does not make specific refinements for the wake.

The calculated lift and drag from the case generally falls within range of the data provided from the NASA Langley turbulence resource [38], and to the experimental data of Ladson [43], as shown in Figure 2.12 and Table 4.2. However, larger discrepancies in the drag than in the lift are observed. A likely reason for the discrepancy is the lack of an off-body wake-refining Cartesian grid, which has been shown to provide enhanced accuracy at sharp corners [44]. Future work will focus on finding optimal methods of coupling strand and Cartesian grids for these types of geometries and flow features.

## 2.6 Conclusions and Future Work

A novel high-order algorithm for strand grids has been presented for two-dimensional turbulent high-Reynolds number flows. The method utilizes a combination of flux correction in the unstructured direction and summation-by-parts finite differences in the strand direction to achieve high-order accuracy. The flux correction method works by canceling low-order truncation errors with accurate flux and solution gradients. By treating the high-order flux derivatives in the strand direction as a source term, it is possible to retain the error cancellation of the flux correction method. The scheme does not require

Table 2.6: Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at  $M = 0.15$ ,  $\alpha = 15^\circ$   $Re = 6 \times 10^6$ .

Solver	$C_d$	$C_l$
Strand FC	3.57122E-3	1.6127
FUN3D	3.56106E-3	1.5547
CFL3D	3.57238E-3	1.5461



high-order quadrature or second-derivative information, except in the case of source terms, which are computed locally within surface-elements. The Reynolds-Averaged Navier-Stokes equations with the Spalart-Allmaras turbulence model were employed within the flux correction methodology as a fully coupled system of equations. Near fourth-order accuracy was observed for all solution variables, including the turbulent working variable.

When applied to two-dimensional high-Reynolds number turbulent flow over a flat plate with zero pressure gradient, the flux correction method accurately predicted drag, even on coarse meshes with significant stretching. Profiles of velocity and normalized eddy viscosity were also accurately predicted. A simple grid refinement study comparing drag against the number of cells used shows the flux correction solver requires significantly fewer cells to produce a coefficient of drag within the range of the FUN3D and CFL3D codes. When applied to high Reynolds number turbulent flow over a bump-in-channel, the high-order strand scheme produces accurate velocity and eddy viscosity profiles, even on a mesh with significant stretching and a high level of grid-skewness around the critical area of the bump. When applied to high Reynolds number turbulent flow over a NACA 0012 airfoil at various angles of attack, the flux correction method showed accurate coefficient of pressure prediction. Small discrepancies were observed in the coefficient of drag. This would likely be resolved by the use of a high-order adaptive off-body Cartesian mesh, as shown previously by Work et al. [44].

The performance of strand-based schemes in the presence of non-smooth surface mesh features, such as corners and ridges, is an on-going research issue requiring further investigation. Previous studies using second-order schemes have indicated sensitivity of strand grid solutions to these features [44]. These studies need to be repeated with the present high-order scheme to investigate any similar sensitivity. Moreover, as the ultimate goal of the strand grid method is to enable fully automatic viscous quality mesh generation for arbitrary geometry, these studies need to be carried out in three dimensions, for which the challenges become more acute. Nonetheless, the present study represents an important and encouraging step in this direction. Furthermore, optimal methods of coupling to off-body

Cartesian grids need to be devised to handle these features. Other future efforts will focus on extending the flux correction method to other turbulence models such as the Menter SST  $k-\omega$  model.

### **Acknowledgments**

Development was performed with the support of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office, by the Army Research Office Fluid Dynamics Program directed by Dr. Frederick Ferguson, and by the Office of Naval Research Sea-Based Aviation program directed by Dr. Judah Milgram and Mr. John Kinzer.

**REFERENCES**

- [1] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [2] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [3] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA paper* 2009-3792, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [4] Katz, A., Wissink, A., Sankaran, V., Meakin, R., and Sitaraman, J., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 6512–6530.
- [5] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.
- [6] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [7] Lee, Y.-L. and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [8] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.

- [9] 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.
- [10] 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.
- [11] Tong, O., Katz, A., and Wissink, A., “Turbulent Flow Validation in the Helios Strand Solver,” *AIAA paper*, AIAA 52nd Aerospace Sciences Meeting, National Harbor, MD, January 2014.
- [12] Wissink, A., “Helios Solver Developments Including Strand Meshes,” Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.
- [13] Barth, T. J. and Frederickson, P., “Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction,” *AIAA paper* 1990-0013, AIAA 28th Aerospace Sciences Meeting, Reno, NV, January 1990.
- [14] Delanaye, M. and Liu, Y., “Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids,” *AIAA paper* 1995-3259, AIAA 14th CFD Conference, Norfolk, June 1999.
- [15] Ollivier-Gooch, C., Nejat, A., and Michalak, K., “On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes,” *AIAA paper* 2007-4464, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [16] Kreiss, H. and Scherer, G., “Finite element and finite difference methods for hyperbolic partial differential equations,” *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. D. Boor, Academic Press, 1974.
- [17] Strand, B., “Summation by parts for finite difference approximation for  $d/dx$ ,” *Journal of Computational Physics*, Vol. 110, 1994, pp. 47–67.

- [18] Carpenter, M. H., Gottlieb, D., and Abarbanel, S., “The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes,” *Journal of Computational Physics*, Vol. 108, No. 2, 1993, pp. 272 – 295.
- [19] Carpenter, M., Gottlieb, D., and Abarbanel, S., “Time-Stable Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High-Order Compact Schemes,” *NASA Contract Report 191436*, March 1993.
- [20] 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.
- [21] Fernandez, D. C. D. R. and Zingg, D., “High-Order Compact-Stencil Summation-By-Parts Operators for the Second Derivative with Variable Coefficients,” Tech. Rep. ICCFD7-2803, 7th International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, HI, July 2012.
- [22] 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.
- [23] Allmaras, S., Johnson, F., and Spalart, P., “Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model,” Tech. Rep. ICCFD7-1902, 7th International Conference on Computational Fluid Dynamics, July 2012.
- [24] Spalart, P. and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *Recherche Aerospaciale*, Vol. 1, 1994, pp. 5–21.
- [25] Solin, P., Segeth, K., and Dolezel, I., *Higher-Order Finite Element Methods*, 2003.
- [26] Bassi, F. and Rebay, S., “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics*, Vol. 138, December 1997, pp. 251–285.

- [27] Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.
- [28] Svård, M., Carpenter, M., and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, far-field boundary conditions,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1020–1038.
- [29] Svård, M. and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, No-slip wall boundary conditions,” *Journal of Computational Physics*, Vol. 227, 2008, pp. 4805–4824.
- [30] Nordström, J., Forsberg, K., Adamsson, C., and Eliasson, P., “Finite Volume Methods, Unstructured Meshes and Strict Stability for Hyperbolic Problems,” *Applied Numerical Mathematics*, Vol. 45, 2003, pp. 453–473.
- [31] Yoon, S. and Jameson, A., “Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations,” *AIAA Journal*, Vol. 26, 1988, pp. 1025–1026.
- [32] Jameson, A. and Caughey, D., “How Many Steps are Required to Solve the Euler Equations of Steady, Compressible Flow: In Search of a Fast Solution Algorithm,” *AIAA paper 2001-2673*, AIAA 15th Computational Fluid Dynamics Conference, Anaheim, CA, June 2001.
- [33] Caughey, D. and Jameson, A., “Fast Preconditioned Multigrid Solution of the Euler and Navier-Stokes Equations for Steady, Compressible Flows,” *International Journal for Numerical Methods in Fluids*, Vol. 43, 2003, pp. 537–553.
- [34] Mavriplis, D. and Jameson, A., “Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes,” *AIAA paper 1987-0353*, AIAA 25th Aerospace Sciences Meeting, Reno, NV, January 1987.
- [35] Brandt, A., “Multi-Level Adaptive Solutions to Boundary-Value Problems,” *Mathematics of Computation*, Vol. 31, 1977, pp. 333–390.

- [36] Jameson, A. and Mavriplis, D., “Finite volume Solution of the Two-dimensional Euler Equations on a Regular Triangular Mesh,” *AIAA Journal*, Vol. 24, 1986, pp. 611–618.
- [37] Venkatakrisnan, V. and Mavriplis, D. J., “Agglomeration Multigrid for the Three Dimensional Euler Equations,” *AIAA paper* 1994-0069, AIAA 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1994.
- [38] Rumsey, C., “NASA Langley Turbulence Modeling Resource (<http://turbmodels.larc.nasa.gov>),” 2012.
- [39] Roache, P., “Code Verification by the Method of Manufactured Solutions,” *Transactions of the ASME*, Vol. 124, 2002, pp. 4–10.
- [40] Roy, C., “Review of Code and Solution Verification Procedures for Computational Simulation,” *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.
- [41] Katz, A. and Sankaran, V., “Mesh Quality Effects on the Accuracy of Euler and Navier-Stokes Solutions on Unstructured Meshes,” *Journal of Computational Physics*, Vol. 230, No. 20, 2011, pp. 7670–7686.
- [42] 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.
- [43] 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.
- [44] 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.

## CHAPTER 3

### HIGH-ORDER METHODS FOR TURBULENT FLOWS ON THREE-DIMENSIONAL STRAND GRIDS

#### 3.1 Abstract

In this paper, we formulate a high-order flux correction method for three-dimensional laminar and turbulent flows on strand grids. Building on previous work, we treat flux derivatives along strands with high-order summation-by-parts operators and penalty-based boundary conditions. Where turbulence modeling is required, a robust version of the Spalart-Allmaras model is employed that accommodates negative values of the turbulence working variable. Fundamental verification and validation studies are considered, which demonstrate the flux correction method achieves high-order accuracy for both laminar and turbulent flows. The high-order flux correction requires only 30% more walltime to converge when compared to a second-order scheme.

#### 3.2 Introduction

Computing high-fidelity solutions for unsteady high-Reynolds number viscous flows over complex geometry presents one of the greatest challenges for computational fluid dynamics (CFD) today. First, geometry handling and meshing can require even experts to spend days or weeks before the computation even begins. Without increasing mesh automation, the percentage of time devoted to mesh generation using current practices will continue to increase relative to total simulation time. Second, the use of traditional second-order schemes on unstructured grid configurations often results in accuracy limitations evident once the computation is complete. Though high-order methods have seen an increasing level of research in recent years, numerically diffuse second-order methods are still in use by the vast majority of CFD practitioners. By in large, high-order accurate methods for



unstructured grids are not yet at production level. Finally, poor scalability becomes a serious issue with ever-increasing mesh sizes required by complex systems. Automating viscous mesh generation, preserving spatial and temporal accuracy, and maintaining computational efficiency via scalability are currently among the greatest research challenges in CFD today.

The strand-Cartesian approach has shown great potential to alleviate many of these difficulties [1–4]. Strand and Cartesian grids allow the possibility of fully automatic volume grid generation while enhancing scalability and the potential for high-order accuracy. Near solid bodies, 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 6.1(a). Away from solid bodies, adaptive Cartesian grids resolve vortical shedding and wake features with efficient high-order algorithms, shown in Figure 6.1(b). Due to the robust and automatic nature of the strand-Cartesian grid generation process, the technique is easily extensible to moving-body problems for which the grid can readily be regenerated at each time step. Strand and Cartesian grids communicate through implicit overset interpolation [5–7], which is facilitated by the compact grid representation of the strand-Cartesian mesh system. A typical three-dimensional strand-Cartesian grid system may be stored on each processor in a parallel computation, allowing for self-satisfying domain connectivity [1] and reducing the percentage of time needed for intergrid communication [8].

The primary goal of this work is to demonstrate improved near-body accuracy and efficiency for turbulent flows through high-order flux correction methods in three dimensions. We focus on the near-body strand region because even advanced off-body high-order solution techniques may become ineffective if excessive near-body discretization error causes non-physical diffusion of vortical flow features, as shown in recent work by Wissink et al. [9]. While high-order wake-capturing in the off-body is critical for capturing vortex-body interactions, near-body accuracy is critical to properly capturing the initial generation of these structures from the surface and to obtaining meaningful body force computations, especially drag. Thus, in this work, we focus only on the flux correction method for strand

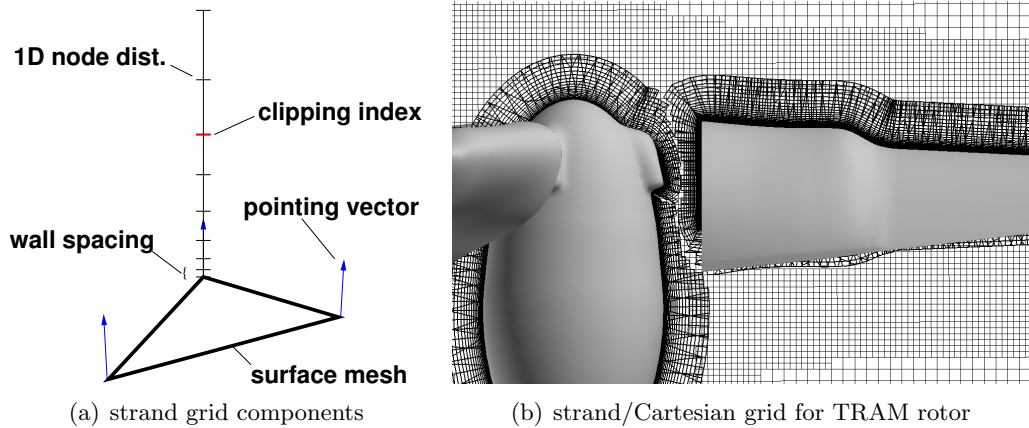


Fig. 3.1: Strand grid elements and example strand/Cartesian grid system for the TRAM rotor.

grids. Coupling with an off-set Cartesian grid is an eventual goal and is not addressed in this work. Figure 6.1(b) shows an example of the goal we are currently working towards.

The high-order flux correction method, a novel method of obtaining third- and fourth-order accuracy on strand grids, was recently investigated by Work and Katz [10] and Tong et al. [11], building upon previous encouraging results [12, 13] designed to address these issues. The high-order strand method involves correction of the flux in the unstructured plane, combined with stable summation-by-parts (SBP) operators [14–18] implemented as source terms to approximate flux derivatives along strands. We impose boundary conditions weakly through simultaneous approximation terms (SAT) [19] added as penalties at boundaries, both as part of the SBP operator along strands [20, 21] as well as the flux correction operator in the unstructured plane, consistent with a stable finite volume scheme [22]. Unlike most high-order methods under investigation today, the flux correction method uses a node-centered finite volume method as a starting point to which truncation error-canceling terms are added to increase accuracy. The method requires no additional flux quadrature or second derivatives in the solution reconstruction like quadratic finite volume schemes [23–25]. Recent studies indicate the resulting scheme requires minimal computational overhead beyond second-order schemes [10]. This paper aims to extend our previous work to three-dimensional turbulent flows in order to take an important step toward practical application.

The paper is outlined as follows: First, we provide details of the high-order strand grid discretization scheme, including turbulent flux correction and high-order summation-by-parts operators for first derivatives and second derivatives with variable coefficients. Next, we present results for three-dimensional cases involving the method of manufactured solutions, flow over a sphere at various Reynolds numbers, and flow over a hemisphere-cylinder. Finally, we conclude and discuss potential future work.

### 3.3 High-Order Strand Grid Discretization

In this work, we solve the compressible Reynolds-Averaged Navier-Stokes (RANS) equations in three-dimensions. The “negative” Spalart-Allmaras model [26], which admits negative values of the turbulence working variable, is used to achieve turbulence closure. Negative values of  $\tilde{\nu}$  are often admitted on under-resolved grids, and at the edge of boundary layers and wakes. The combined RANS-SA equations may be expressed as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S, \quad (3.1)$$

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

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho \tilde{\nu} \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ \rho h u_j \\ \rho \tilde{\nu} u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ \frac{\eta}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_j} \end{pmatrix}, \quad (3.2)$$

and the vector of source terms,  $S$ , is defined as

$$S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathcal{P} - \mathcal{D} + c_{b2} \rho \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \end{pmatrix}. \quad (3.3)$$

The turbulent source term for the SA model consists of a production term,  $\mathcal{P}$ , and a destruction term,  $\mathcal{D}$ . Here,  $\rho$  is the density,  $u_j$  is the  $j^{\text{th}}$  component of the fluid velocity,  $p$  is the pressure,  $e$  is the total energy per unit mass,  $h \equiv e + p/\rho$  is the total enthalpy per unit mass,  $\tilde{\nu}$  is the turbulence working variable,  $\sigma_{ij}$  is the deviatoric stress tensor,  $q_j$  is the  $j^{\text{th}}$  component of the heat flux vector, and  $\frac{\eta}{\sigma}$  is the turbulent diffusion coefficient. The stress tensor is defined as

$$\sigma_{ij} = 2(\mu + \mu_T) \left( S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (3.4)$$

where  $\mu$  is the dynamic viscosity,  $\mu_T$  is the turbulent viscosity, and  $S_{ij}$  is the rate of strain tensor, defined as

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (3.5)$$

The heat flux vector is obtained with Fourier's Law, and defined as

$$q_j = -C_p \left( \frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \frac{\partial T}{\partial x_j}, \quad (3.6)$$

where  $T$  is temperature,  $C_p$  is the specific heat,  $Pr$  is the Prandtl number,  $Pr_T$  is the turbulent Prandtl number. In addition, Sutherland's Law is utilized to relate dynamic viscosity and temperature, and the ideal gas equation of state is used.

The “negative” SA model is designed to use the standard SA model when the turbulent working variable,  $\tilde{\nu}$ , is positive, and adds modifications to the standard model when the turbulent working variable is negative. The details of the standard SA model, including the well-known definitions of the production and destruction terms, may be found in the original work by Spalart and Allmaras [27].

In this work, Equation 5.1 is solved on strand grids consisting of an unstructured triangular surface tessellation extruded along straight lines (strands) away from solid bodies. Each stack of prismatic cells emanating from the surface in the physical space may be mapped to a standard computational space as shown in Figure 5.2. To facilitate high-order algorithms, the triangular base of each prismatic element may be divided into equally spaced sub-triangles in the  $r$ - $s$  plane. In this work, we investigate up to fourth-order surface

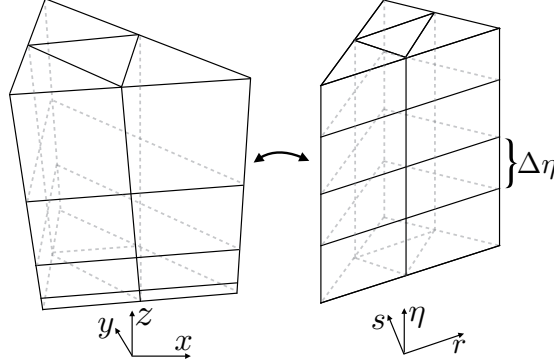


Fig. 3.2: Mapping of strand stack from physical space to computational space.

elements. Quadratic surface elements are shown in Figure 5.2. The stretched distribution of nodes along each strand in the physical space is mapped to an equally spaced distribution in the  $\eta$ -direction in the computational space, where  $\eta \in (0, 1)$ . In the computational space, the strand spacing is  $\Delta\eta = 1/(N - 1)$ , where  $j = 1, \dots, N$  is the strand node numbering beginning with the node on the surface. To avoid confusion, the triangular-shaped elements forming the various levels in the prism stack will be referred to as “surface elements,” while the three-dimensional elements formed by an equally-spaced stacking the surface elements in the  $\eta$ -direction will be referred to as “volume elements.” Additionally, the triangles formed from sub-dividing each surface element will be referred to as “sub-triangles.”

Upon transformation to the computational space, Equation 5.1 becomes

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} + \frac{\partial \hat{H}}{\partial \eta} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} - \frac{\partial \hat{H}^v}{\partial \eta} = \hat{S}, \quad (3.7)$$

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

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

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

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

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

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

Here,  $J$  is the Jacobian of the transformation,  $\hat{F}_j$  and  $\hat{F}_j^v$  are the transformed inviscid and viscous fluxes, and partial differentiation is denoted with a subscript (e.g.  $\partial x / \partial s = x_s$ ).

In previous work, we explain in detail the novel manner in which a high-order discretization is carried out on strand grids [10]. Here, we focus on significant improvements and refinements to the original method. The general strategy is to perform a high-order finite-volume flux balance in the  $r$ - $s$  plane, known as “flux correction,” and to use high-order finite-differences based on SBP operators in the  $\eta$ -direction. To facilitate the flux correction algorithm, two-dimensional median-dual control volumes are constructed around each node in each high-order surface element in the  $r$ - $s$  plane. The sub-triangles and median-dual control volumes in a single surface element are shown as the black solid lines and red dashed lines in Figure 5.3 for quadratic and cubic surface elements, respectively. In this work, quartic elements are used, but are not shown in Figure 5.3. It is known that in order for high-order schemes to deliver theoretical orders of accuracy, high-order curved boundary elements are essential [28, 29], which we employ in this work.

A critical aspect of the method is to treat the  $\eta$ -derivatives in Equation 5.8 with a

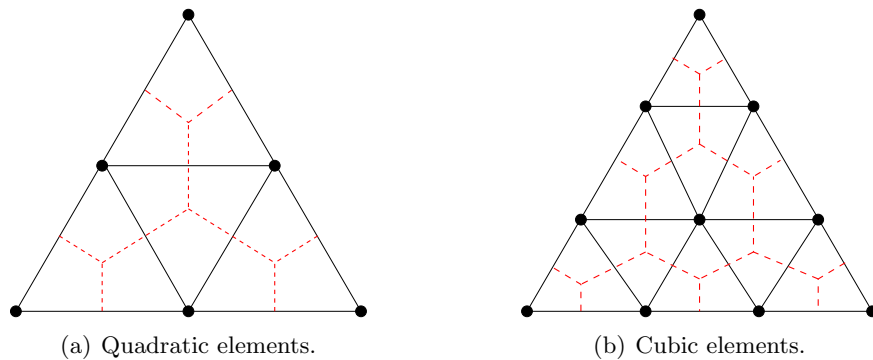


Fig. 3.3: Element mappings used for gradient reconstruction.

particular source term discretization which preserves the accuracy of the flux correction procedure in the  $r$ - $s$  plane. Therefore, the  $\eta$ -derivatives and physical time derivative are moved to the right-hand side and treated as source terms:

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} = \tilde{S}, \quad (3.8)$$

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

A pseudo-time derivative is added on the left-hand side of Equation 5.9 to facilitate a semi-implicit time-marching solution [11]. As long as each term in  $\tilde{S}$  is computed to at least second-order accuracy in terms of the truncation error, the corrected flux balance in the  $r$ - $s$  plane will retain desirable truncation error properties, resulting in a high-order accurate scheme. Examining equation 5.9, the three-dimensional equations essentially reduce to a two-dimensional problem in the  $r$ - $s$  plane at each layer in the strand grid. The layers are coupled together through the new source term,  $\tilde{S}$ , which contains  $\eta$ -derivatives of the flux.

In the following subsections new aspects of the flux correction procedure in the  $r$ - $s$  plane are discussed, followed by a discussion of the SBP operators used to approximate the  $\eta$ -derivatives contributing to the modified source term,  $\tilde{S}$ . Additionally, we discuss issues related to computing element mappings, as well as parallel communication strategies.

### 3.3.1 Unstructured Flux Correction in the $r$ - $s$ Plane

Equation 5.9 requires a two-dimensional discretization in the  $r$ - $s$  plane suitable for unstructured grids with a source term. The discretization used for this purpose here is the flux correction scheme of Katz and Sankaran [12]. Recently, Pincock and Katz [13] extended the original scheme to include the viscous terms in the Navier-Stokes equations. A defining feature of the flux correction scheme is its ability to retain high-order accurate truncation error on general simplex grids, which results in third-order discretization error for inviscid fluxes, and fourth-order discretization error for viscous fluxes.

The flux correction procedure differs from a conventional finite volume scheme in the

definition of the numerical fluxes. Following previous work on flux correction [12], the inviscid numerical fluxes computed between nodes  $a$  and  $b$  are computed as

$$\hat{\mathcal{F}}_{ab,j} = \frac{1}{2} \left( \hat{\mathcal{F}}_R + \hat{\mathcal{F}}_L \right) - \frac{1}{2} \left| \hat{\mathcal{A}}(Q_R, Q_L) \right| (Q_R - Q_L). \quad (3.9)$$

The unique aspect of the flux correction scheme is the use of reconstructed fluxes, along with high-order gradient information. The left and right reconstructed fluxes are computed as

$$\hat{\mathcal{F}}_L = \hat{\mathcal{F}}_{a,j} + \frac{1}{2} \Delta \mathbf{r}_{ab}^T \left( \nabla_{rs}^h \hat{\mathcal{F}} \right)_{a,j}, \quad \hat{\mathcal{F}}_R = \hat{\mathcal{F}}_{b,j} - \frac{1}{2} \Delta \mathbf{r}_{ab}^T \left( \nabla_{rs}^h \hat{\mathcal{F}} \right)_{b,j}, \quad (3.10)$$

where  $\Delta \mathbf{r}_{ab}^T = (r_b - r_a, s_b - s_a)$ , and  $\nabla_{rs}^h$  is an estimate of the gradient in the  $r$ - $s$  plane computed to at least second-order accuracy, such that

$$\nabla_{rs}^h = \nabla_{rs} + O(h^q), \quad q \geq 2.$$

The strategy to compute such gradients is to compute the derivatives of a local polynomial representation within each surface element, and then volume-average among the shared representations in neighboring elements. The averaging is critical to centering the gradients and to the overall stability of the scheme. The requirement that  $q \geq 2$  may be satisfied by using at least quadratic surface elements shown in Figure 5.3.

The ability to compute accurate gradients, therefore, is critical to the overall accuracy of the FC scheme. We may express the gradients in the  $r$ - $s$  plane as

$$\begin{pmatrix} \partial_r \\ \partial_s \end{pmatrix} \Big|_{0,j}^h = \begin{pmatrix} x_r & y_r & z_r \\ x_x & y_s & z_s \end{pmatrix} \Big|_{0,j}^h \begin{pmatrix} \overline{\partial_x} \\ \overline{\partial_y} \\ \overline{\partial_z} \end{pmatrix}_{0,j} = \begin{pmatrix} x_r & y_r & z_r \\ x_x & y_s & z_s \end{pmatrix} \Big|_{0,j}^h \begin{pmatrix} \overline{r_x \partial_r + s_x \partial_s + \eta_x \partial_\eta} \\ \overline{r_y \partial_r + s_y \partial_s + \eta_y \partial_\eta} \\ \overline{r_z \partial_r + s_z \partial_s + \eta_z \partial_\eta} \end{pmatrix}_{0,j},$$

where the overbar denotes a volume average at nodes shared by more than one element. Because  $\eta$ -contributions possess a unique value, even at shared nodes, then  $(\overline{\eta_x \partial_\eta})_{0,j} =$



$(\eta_x \partial_\eta^h)_{0,j}$ . The same is true for the  $\eta_y$  and  $\eta_z$  terms. Additionally, the quantities

$$(x_r \eta_x + y_r \eta_y + z_r \eta_z)_{0,j}^h, \quad (x_s \eta_x + y_s \eta_y + z_s \eta_z)_{0,j}^h$$

are identically zero from the metric relations, allowing us to compute the gradients in the  $r$ - $s$  plane as

$$\begin{pmatrix} \partial_r \\ \partial_s \end{pmatrix} \Big|_{0,j}^h = \begin{pmatrix} x_r & y_r & z_r \\ x_s & y_s & z_s \end{pmatrix} \Big|_{0,j}^h \begin{pmatrix} \overline{r_x \partial_r + s_x \partial_s} \\ \overline{r_y \partial_r + s_y \partial_s} \\ \overline{r_z \partial_r + s_z \partial_s} \end{pmatrix}_{0,j} + O(h^q). \quad (3.11)$$

In this manner, gradient reconstructions of the flux and solution in Equation 5.11 may be performed entirely within each  $r$ - $s$  plane of the strand grid.

Viscous fluxes are computed in a manner similar to inviscid fluxes, but without averaging gradients in order to maintain a compact viscous stencil [13]. In addition to the inviscid and viscous flux treatment, the flux correction method requires special treatment involving the gradient and Hessian of the source term,  $\tilde{S}_{0,j}^h$ . For the gradient and Hessian terms, local computations with no element volume averaging of gradients is sufficient, as was done for the viscous fluxes. Details of the source treatment may be found in previous work [10].

### 3.3.2 SBP Finite Differences in the $\eta$ -Direction

In Equation 5.9, the  $\eta$ -flux derivatives are grouped within a new source term such that the order of accuracy of the truncation error of the flux correction scheme may be maintained. This strategy proves successful as long as the terms in  $\tilde{S}$  are computed to second-order accuracy or better in terms of truncation error. Consequently, the terms  $\partial \hat{H} / \partial \eta$  and  $\partial \hat{H}^v / \partial \eta$  require additional care to ensure high-order accuracy, stability, and discrete conservation. Summation by parts (SBP) finite difference operators [14–18] provide a framework for ensuring these properties are satisfied. As in our previous work [10], we investigate inviscid and viscous SBP operators that are second, fourth, or sixth-order accurate in the interior based on the work of Mattsson [17] and Fernandez and Zingg [18].

We also add artificial dissipation based on the operators of Diener et al. [30]. The dissipation operators possess order of accuracy greater than or equal to the derivative operators and avoid spurious oscillations arising from non-linearities in the governing equations. In the results section, we refer to our high-order scheme as ‘‘Strand FC.’’ Unless otherwise stated, the Strand FC scheme uses SBP operators with fourth-order truncation error. We impose boundary conditions weakly through simultaneous approximation terms (SAT) [19] added as penalties at boundaries, both as part of the SBP operator along strands [20, 21] as well as the flux correction operator in the unstructured plane, consistent with a stable finite volume scheme [22].

Following Mattsson [17], a combination of SBP operators for the second derivative with variable coefficients along with compatible first derivative operators may be used to discretize,  $\partial \hat{H}^v / \partial \eta$ . In this approach, a second derivative operator is employed for the pure  $\eta$ -derivative, while a compatible first derivative operator is used for the mixed derivative terms:

$$\frac{\partial \hat{H}^v}{\partial \eta} \approx D_\eta \left[ B^r (Q_r^p)^h + B^s (Q_s^p)^h \right] + D_{2\eta} (B^\eta) Q^p. \quad (3.12)$$

Here, the partial derivative terms,  $(Q_r^p)^h$  and  $(Q_s^p)^h$ , are computed locally within surface elements with no averaging at nodes shared among adjacent elements. In Equation 3.12,  $B^r$ ,  $B^s$ , and  $B^\eta$  are variable coefficient matrices containing viscosity and heat flux coefficients. In this manner the viscous stencil remains compact. Further details of the source treatment in the  $\eta$ -direction, including SBP operators, consistent source term discretization, and boundary condition enforcement via penalty terms, may be found in previous work [10].

### 3.3.3 Numerical Approximation of Element Mappings

Despite the requirements of using high-order gradients and SBP operators in the FC scheme, if consistent approximation of the mapping terms in Equation 5.8 is not followed, high-order accuracy will be lost. In fact, the simplest flow–freestream preservation–will be unattainable. Computation of mapping terms that admit freestream solutions have been investigated most often in the context of high-order finite-difference schemes on curvilinear

meshes [31]. Here, we formulate a novel method of computing element mappings for the unique case of high-order strand grids.

In arbitrary freestream flow, Equation 5.8 leads to three constraints that must be satisfied discretely:

$$(\hat{r}_x)_r + (\hat{s}_x)_s + (J\eta_x)_\eta = 0, \quad (3.13)$$

$$(\hat{r}_y)_r + (\hat{s}_y)_s + (J\eta_y)_\eta = 0,$$

$$(\hat{r}_z)_r + (\hat{s}_z)_s + (J\eta_z)_\eta = 0,$$

Only the first of these is discussed in detail as the other two follow a similar method.

In order to preserve freestream flow, we discretize Equation 3.13 in a manner completely consistent with the flux formulation itself, complete with artificial dissipation and penalty terms at boundaries. We first compute the mapping terms,  $\hat{r}_x$  and  $\hat{s}_x$ , locally within each element according to the definition in Equation 5.8, except in “conservative” form,

$$\hat{r}_x \equiv Jr_x = (y_s z)_\eta - (y_\eta z)_s, \quad \hat{s}_x \equiv Js_x = (y_\eta z)_r - (y_r z)_\eta.$$

We then discretize Equation 3.13 to find  $\eta_x$  at each node, which may be expressed as

$$\sum_{i \in 0} \left[ \frac{1}{2} (\hat{r}_{x,L} + \hat{r}_{x,R}) \hat{A}_r + \frac{1}{2} (\hat{s}_{x,L} + \hat{s}_{x,R}) \hat{A}_s \right] \\ + D_\eta (MJ\eta_x) - \frac{1}{2} d_{j+\frac{1}{2}} (MJ\eta_x) + \frac{1}{2} d_{j-\frac{1}{2}} (MJ\eta_x) - \text{penalty} = 0.$$

Here,  $d_{j+\frac{1}{2}}$  represents artificial dissipation acting on the  $MJ\eta_x$  term,  $M$  represents the source discretization operator, and “penalty” represents a boundary penalty term needed to solve the mapping equation in a well-posed manner. Note the similarity in form of the discretization to find  $\eta_x$  to the discretization of Equation 5.8 to solve for  $Q$  itself. This is intentional because it ensures discrete satisfaction of the identity in Equation 3.13.

While the solution of Equation 3.13 requires little computational effort, it does require inversion of the source operator,  $M$ , similar to a mass matrix inversion. In practice a few

Jacobi iterations to invert the source operator is sufficient and is easily parallelizable.

### 3.4 Results

The ability of the flux correction method to accurately compute three-dimensional laminar and high Reynolds number turbulent flow within the strand grid paradigm is investigated. First, we conduct an accuracy test through the method of manufactured solutions. Next we perform a study of three-dimensional laminar flow over a sphere for a range of Reynolds numbers. Finally we compute turbulent flow over a hemisphere-cylinder configuration.

#### 3.4.1 Verification Studies with Manufactured Solutions

First, we rigorously test the accuracy of the high-order strand algorithm by performing grid refinement studies using manufactured solutions (MMS) [32, 33]. The use of MMS to assess accuracy of discretizations for compressible viscous flows is important due to the lack of exact solutions. Smooth trigonometric functions for velocity, pressure and temperature are shown in Figure 5.4(a) for a cube geometry. When assessing high-Reynolds number flows, the working variable of the Spalart-Allmaras model  $\tilde{\nu}$  is treated in a similar manner. Four levels of grid refinement are used, and the error between the discrete and exact manufactured solution is measured on each mesh after converging to machine precision.

Results of this study are shown in Figure 5.4. Two Reynolds number cases are shown;  $Re = 10$  and  $Re = 100,000$ . Each show the inviscid and viscous terms in isolation as well as combined. Where the RANS-SA equations are solved, the SA closure source terms are turned off when examining inviscid and viscous terms in isolation, and turned on when all the terms are combined. One advantage of verification via manufactured solutions is the ability to isolate various terms (e.g. inviscid or viscous), to assess the accuracy of each. As the figures show, the inviscid terms asymptote sharply to third-order accuracy. In the case of the  $Re = 100,000$ , the inviscid curve and combined curve fall directly on top of one another. The viscous terms are significantly more accurate, generating accuracy better than fourth-order. In fact, the viscous discretization is so accurate for both cases that error

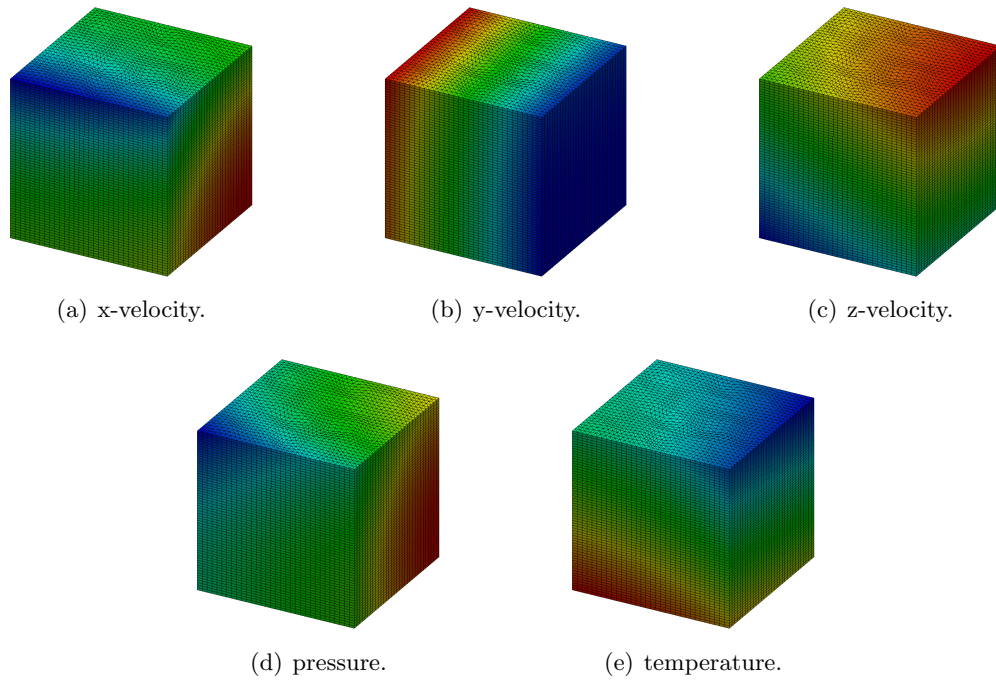


Fig. 3.4: Manufactured solution used for order of accuracy verification studies.

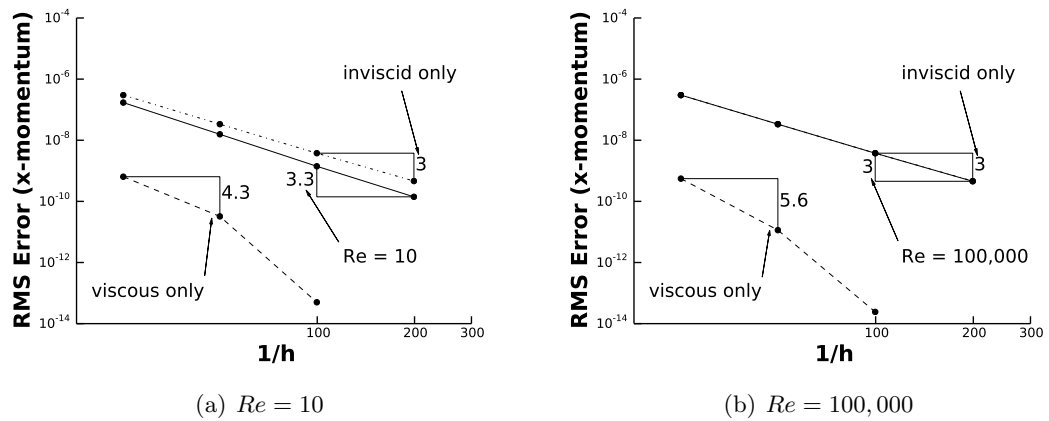


Fig. 3.5: Order of accuracy results for manufactured solution in a cube geometry.

levels fall below machine precision after the second refinement level. Most importantly, the combined inviscid and viscous discretizations are compatible, yielding an order of accuracy of 3.3 and 3 for  $Re = 10$  and  $Re = 100,000$ , respectively. These results are valid for irregular surface meshes and smooth strand distributions with potentially very high aspect ratios. We demonstrate this with an additional MMS verification study over a sphere, shown in Figure 5.5, where we observe the flux correction method maintains its high-order of accuracy on an irregular sphere grid. It should be noted that conventional second order schemes are easily recovered by turning off the flux correction terms and lowering the difference order along strands, which is easily accomplished with an input flag.

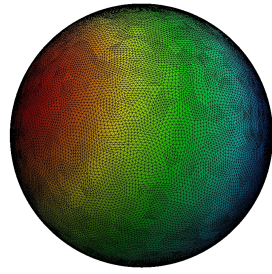
We wish to emphasize the fact that the flux correction solver is able to maintain high-order of accuracy with the Spalart-Allmaras turbulence model (e.g.  $\tilde{\nu}$  RMS error follows a trend similar the to x-momentum error). The SA turbulence model within the flux correction solver is fully coupled with the RANS equations, and thus exhibits greater accuracy many traditional RANS solvers with segregated turbulent models, or models that reduce the turbulence equation to first-order.

### 3.4.2 Steady Flow over a Sphere

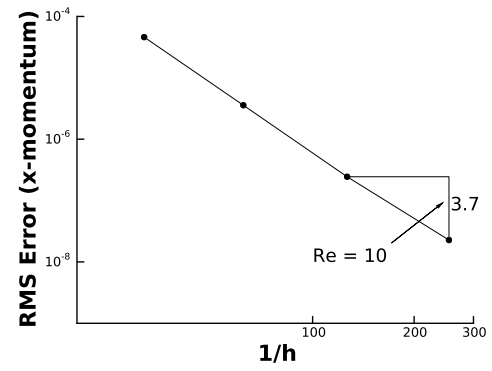
We next examine the accuracy of the high-order strand flux correction method for steady flow over a sphere at  $M = 0.2$  and low Reynolds number. For this study, we use a grid containing 1024 fourth-order surface elements, resulting in 9236 nodes on the surface. We extend the strand length 20 diameters to the far field and use 128 nodes along each strand. The resulting volume grid contains roughly 1 million nodes total. Figure 3.7(a)

Table 3.1: Order of accuracy of inviscid and viscous terms combined ( $Re = 10$ ) (x-momentum).

Mesh Nodes	Error	Order
2,048	4.943232823e-08	-
16,384	4.991030317e-09	3.473
131,072	5.296494751e-10	3.489
1,048,576	8.361858580e-11	3.326

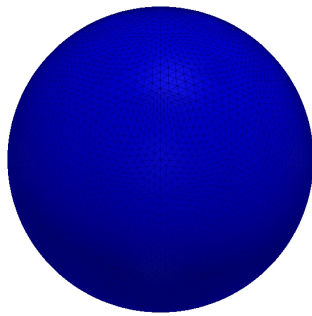


(a) Manufactured solution for pressure.

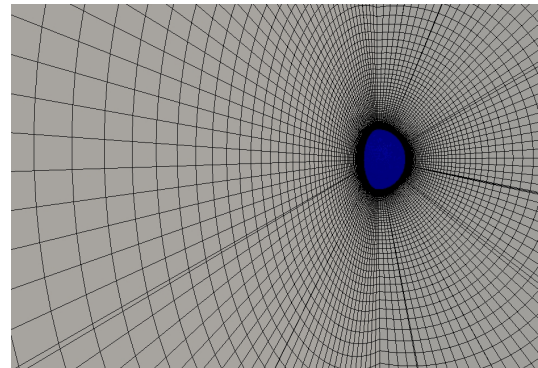


(b) Order of accuracy results.

Fig. 3.6: Verification study for manufactured solution in a sphere geometry.



(a) Sphere surface mesh with quartic elements.



(b) Sphere volume mesh.

Fig. 3.7: Sphere mesh configuration.

shows the surface grid, and the volume strand grid mesh with a section cut out is shown in Figure 3.7(b). At run time, we partition the grid using 24 cores. We also compare the high-order results to a second-order strand method which uses the same mesh configuration, denoted as “Strand” in the corresponding figures.

Because steady flow over a sphere has been studied extensively in the literature, we have a large amount experimental and computational data available for validation. In this study we examine three specific flow features: separation angle, length of the recirculation bubble, and the location of the standing ring vortex center located just downstream of the sphere. We examine five Reynolds numbers,  $Re = 40, 80, 120, 160, 200$ , and compare with the computational results of Magnaudet [34] and Tomboulides [35], and to the experimental results of Pruppacher [36] and Taneda [37].

Plots of the Strand FC data compared with the literature are shown in Figure 3.8 for the three features. The high-order flux correction algorithm shows excellent agreement in all cases. Slightly larger errors are observed in the recirculation length and vortex center location at the higher Reynolds numbers, likely due to the fact that at higher Reynolds numbers, the vortex migrates farther away from the sphere into a region of coarser mesh. While this effect has a minor effect on the accuracy of the flux correction scheme, the second-order scheme suffers greatly from it. The second-order scheme underpredicts the recirculation length and x-coordinate of the vortex center.

### 3.4.3 Unsteady Flow over a Sphere

The case we present is unsteady flow over a sphere at  $M = 0.2$  at a  $Re = 600$ . An

Table 3.2: Order of accuracy of inviscid, viscous and source terms combined ( $Re = 100,000$ ) (x-momentum).

Mesh Nodes	Error	Order
2,048	1.028919706e-07	-
16,384	1.281676554e-08	3.183
131,072	1.455144065e-09	3.166
1,048,576	1.930668590e-10	3.058



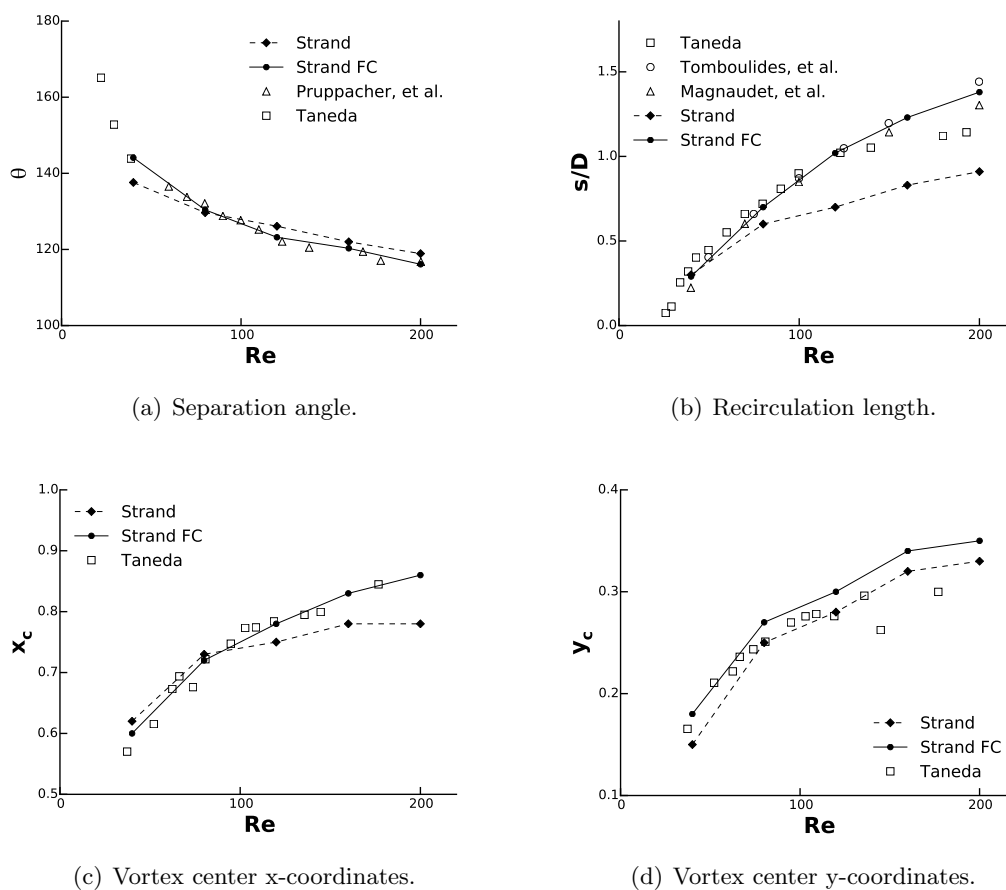


Fig. 3.8: Comparison of computed high-order strand results with experimental data measuring various flow characteristics.

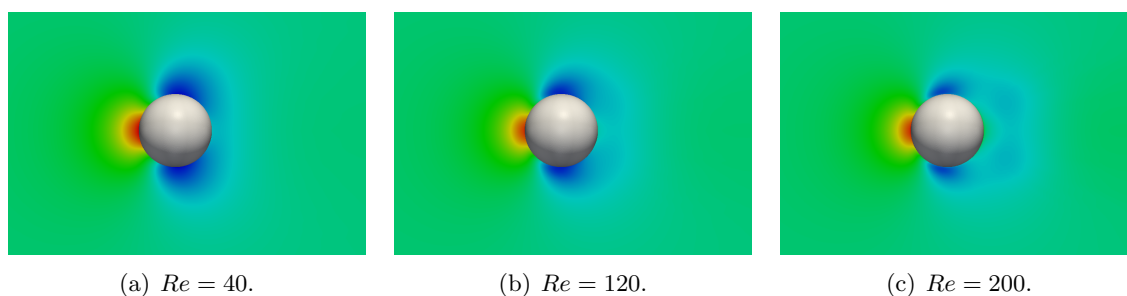


Fig. 3.9: Field plot of pressure for steady flow over a sphere for  $M = 0.2$  at various Reynolds numbers.

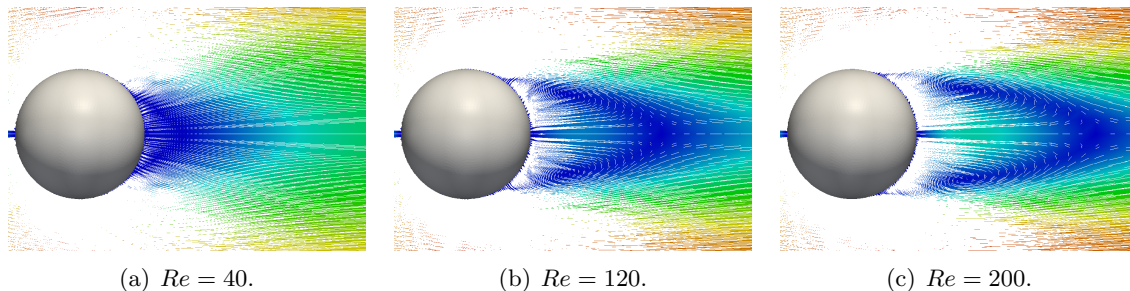
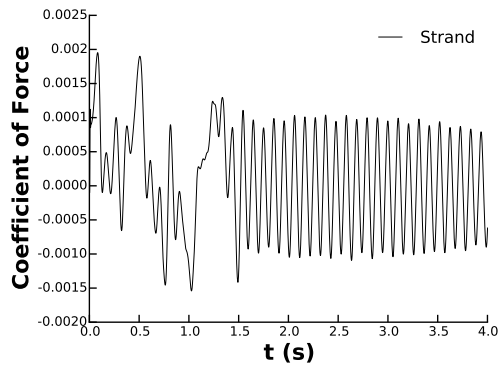


Fig. 3.10: Velocity vectors for steady flow over a sphere for  $M = 0.2$  at various Reynolds numbers.

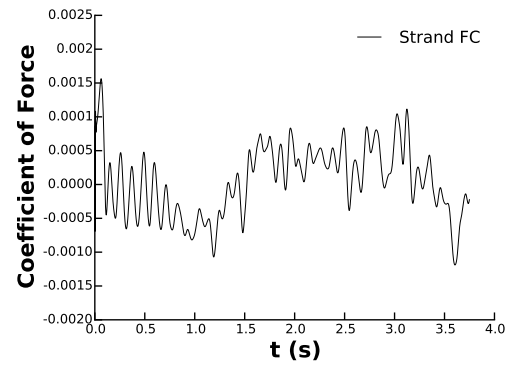
identical mesh configuration to the steady case is used, as shown in Figure 3.7, where the surface mesh is shown in Figure 3.7(a), and the volume mesh with a quarter cut-out is shown in Figure 3.7(b). Solutions were obtained using the flux correction scheme over a physical time span of 3.75 seconds using a physical time step of  $1 \times 10^{-3}$  seconds. At run time, we partition the grid using 512 cores.

Figure 3.11 shows the coefficient of force in the z-directions over time for both the second-order strand solver and the high-order flux correction solver. From the figures, it is clear that once the initial transience has dissipated, the second-order solver shows very regular shedding, whereas flux correction shows highly irregular shedding throughout the entire time period examined. It is well known that when the Reynolds number exceeds 480 over a sphere, an irregular mode is reached, and the shedding of hairpin vortical structures becomes uneven [38, 39]. Thus, we see that the flux correction actually shows the more accurate vortex shedding over the sphere. We suspect that the second-order scheme suffers from numerical dissipation, thus effectively reducing the Reynolds number below 480 where the shedding is regular.

Figure 3.12 shows plots of unsteady flow visualization at a physical time of 1.4 seconds. The early development of a hairpin vortex directly on the lower half of the sphere is shown in Figure 3.12(b), while another hairpin vortex has been shed further down the wake. Coupling the strand mesh with an off-body adaptive Cartesian mesh will greatly increase wake resolution far downstream of the sphere, where vortices were not captured by the

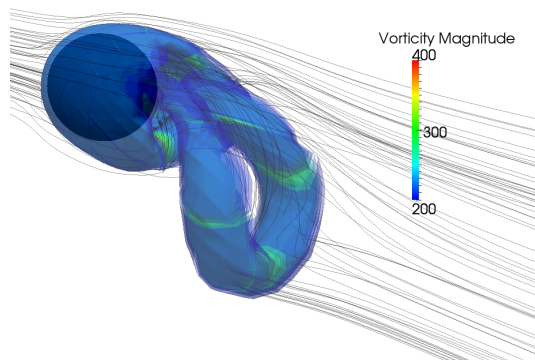


(a) Strand solver.

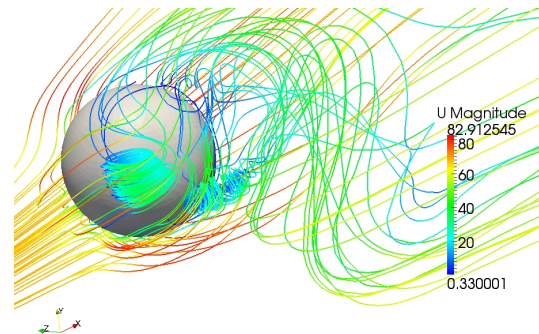


(b) Flux correction strand solver.

Fig. 3.11: Coefficient of force in the z-direction over time.



(a) Isosurface of vorticity magnitude with streamlines.



(b) Close up of unsteady shedding.

Fig. 3.12: Unsteady flow visualization of a sphere at  $M = 0.2$  and  $Re = 600$ .

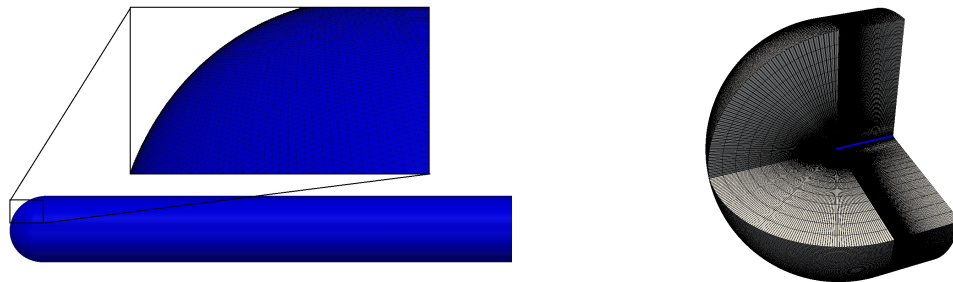
strands in regions where the grid is very coarse. Despite this, some vortex shedding has been captured quite clearly in Figure 3.12(a), where a rendered three-dimensional isosurface of the vorticity magnitude shows the shedding of a hairpin vortex formation. The opacity of the isosurface has been reduced and streamlines have been added to aid in the visualization.

#### 3.4.4 Hemisphere-Cylinder

The final validation case we present is a hemisphere-cylinder configuration to test the SA turbulence model over a smooth body of revolution in three-dimensions. Geometry is taken from work by Hsieh [40], where the cylinder has a radius of 0.5 and a length of 10. Flow conditions 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 degrees. Further details and grids may be found on the NASA-Langley turbulence resource website [41].

Figure 4.14 shows the hemisphere-cylinder surface and volume mesh configuration. The surface mesh, shown in Figure 4.14(a) is tessellated with 19050 **fourth-order surface elements**, totaling of approximately 152701 surface nodes. The strand grid volume mesh, shown in Figure 4.14, consists of 64 nodes along the strands extending for a distance of 40 diameters, totaling in a total mesh size of 9.5 million nodes. At run time, we partition the grid using 2048 cores.

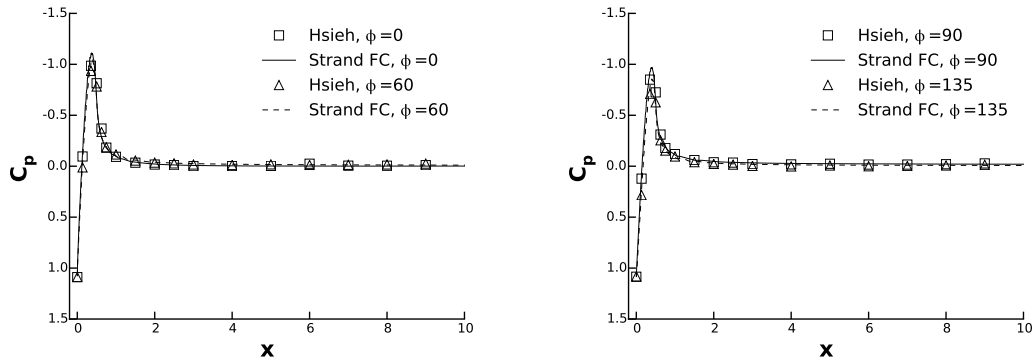
We compare the coefficient of pressure calculated on the surface along the length of



(a) Hemisphere-cylinder surface mesh with quartic elements.

(b) Hemisphere-cylinder volume mesh.

Fig. 3.13: Hemisphere-cylinder mesh configuration.

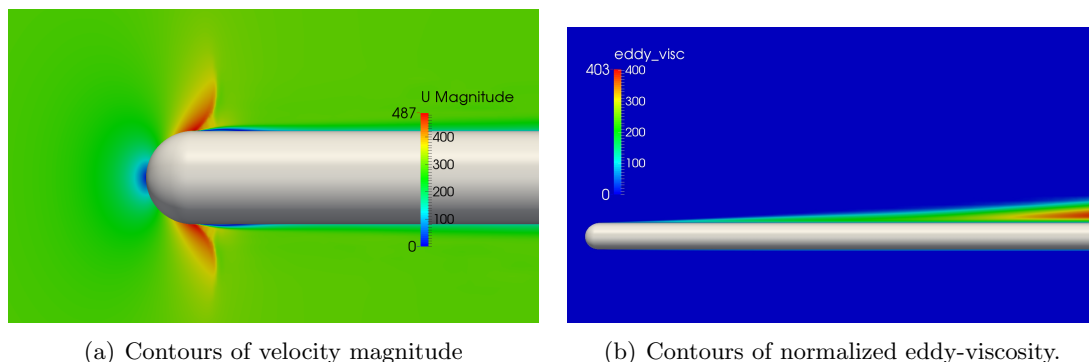


(a) Coefficient of pressure for  $\phi = 0$ degrees and  $\phi = 60$ degrees (b) Coefficient of pressure for  $\phi = 90$ degrees and  $\phi = 135$ degrees

Fig. 3.14: Surface coefficient of pressure comparison of experimental and Strand FC for various  $\phi$ .

the body with the results found by Hsieh [40]. We consider specific positions around the azimuth of the body, namely the 0 degrees, 60 degrees, 90 degrees, 135 degrees positions, with 0 degrees corresponding to the leeside of the body, pointing away from the incoming wind. Plots containing the data computed by the strand algorithm in comparison with experimental results are shown in Figure 3.14. It can be observed that the pressure coefficients calculated by the flux correction algorithm are in excellent agreement with the pressure data collected by Hsieh. Further examination of the pressure coefficients towards the front of the body shows the accuracy of the flux correction algorithm in calculating the pressure coefficient in locations where large gradients in the pressure coefficient can be observed. Field plots of velocity magnitude and normalized eddy-viscosity may be found in Figure 4.15.

Figure 3.16 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 for the second-order strand scheme and flux correction strand scheme to reach the same level of convergence is nearly identical, as shown in Figure 3.16(a). It is noteworthy that in terms of walltime, the flux correction strand scheme requires less than 30% more walltime than the second-order strand scheme to reach



(a) Contours of velocity magnitude

(b) Contours of normalized eddy-viscosity.

Fig. 3.15: Field plots of the hemisphere-cylinder at  $M = 0.6$ .

an identical level of convergence, as shown in Figure 3.16(b). This appears to be a small price to pay for the increased accuracy observed by adding the flux correction terms.

### 3.5 Conclusions

A novel high-order flux correction method for computing turbulent flows on strand grids has been presented. The method uses a combination of summation-by-parts operators in the strand direction and flux correction in the unstructured plane to achieve high-fidelity solutions of compressible turbulent flows. Low-order truncation errors are cancelled with accurate flux and solution gradients in the flux correction method, thereby achieving a formal order of accuracy of 3, although higher orders are often obtained, especially for highly viscous flows. We note that this paper extends previous two-dimensional laminar work on flux correction [10, 11] to three dimensions and turbulent flow. For turbulence simulations, a robust version of the Spalart-Allmaras turbulence model was employed that accommodates negative values of the turbulence working variable.

Fundamental verification studies were conducted for both laminar and turbulent viscous flows. Generally, fourth-order and higher accuracy was observed for viscous terms, and third-order accuracy for inviscid flow. Combined inviscid and viscous orders of accuracy (and source terms when applicable), yield a global order of accuracy around 3.3 and 3 for Reynolds numbers of 10 and 100,000, respectively. On an irregular sphere grid, it was shown that the flux correction method maintained its high order of accuracy. When applied to a

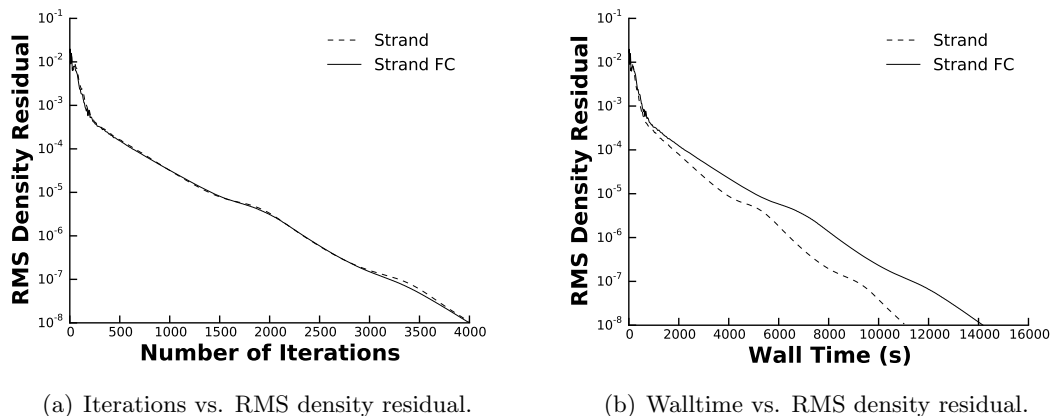


Fig. 3.16: Iterations and walltime vs. RMS density residual for the Strand FC and Strand schemes.

three-dimensional steady laminar flow over a sphere for a range of Reynolds numbers, the flux correction method accurately predicted the center and length of recirculation vortices for each Reynolds number examined and showed an excellent comparison to experimental data. Significant improvements over the second-order algorithm were observed. When applied to three-dimensional unsteady laminar flow over a sphere, the flux correction method showed qualitatively the capability to resolve vortex shedding and propagation. The ability of the flux correction algorithm to resolve turbulent flows was demonstrated using a hemisphere-cylinder configuration. The surface coefficient of pressure was accurately predicted for all azimuths at an angle of attack of 5 degrees. When analyzing the convergence for this case, it was shown that the number of iterations required for second-order and high-order scheme were nearly identical. The high-order scheme required only 30% more walltime than the second-order scheme to achieve the same level of convergence.

Future work will focus on obtaining solutions to transonic and supersonic flows in the presence of shock waves. We anticipate an advantage of our scheme to capture shocks by using well-established limiting techniques. Additionally, we plan to extend this work to more geometrically complex cases involving multiple bodies in motion. This will require coupling with off-body Cartesian grids. Various strategies are under consideration for the

best way to generate surface grids and subsequent strand grids for geometry with sharp corners and ridges.

### **3.6 Acknowledgments**

Development was performed with the support of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office, by the Army Research Office Fluid Dynamics Program directed by Dr. Frederick Ferguson, and by the Office of Naval Research Sea-Based Aviation program directed by Dr. Judah Milgram and Mr. John Kinzer.

### **3.7 Compliance with Ethical Standards**

Funding: Development was performed with the support of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office and by the Office of Naval Research Sea-Based Aviation program (grant no. N000141310827) directed by Dr. Judah Milgram and Mr. John Kinzer.

Conflict of Interest: The authors declare that they have no conflict of interest.



**REFERENCES**

- [1] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [2] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [3] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA paper* 2009-3792, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [4] Katz, A., Wissink, A., Sankaran, V., Meakin, R., and Sitaraman, J., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 6512–6530.
- [5] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.
- [6] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [7] Lee, Y.-L. and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [8] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.

- [9] Wissink, A., “Helios Solver Developments Including Strand Meshes,” Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.
- [10] 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.
- [11] Tong, O., Work, D., and Katz, A., “High-Order Methods for Turbulence Using Strand Grids,” Tech. Rep. ICCFD8-0215, 8th International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 2014.
- [12] 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.
- [13] 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.
- [14] Kreiss, H. and Scherer, G., “Finite element and finite difference methods for hyperbolic partial differential equations,” *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. D. Boor, Academic Press, 1974.
- [15] Strand, B., “Summation by parts for finite difference approximation for  $d/dx$ ,” *Journal of Computational Physics*, Vol. 110, 1994, pp. 47–67.
- [16] Carpenter, M. H., Gottlieb, D., and Abarbanel, S., “The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes,” *Journal of Computational Physics*, Vol. 108, No. 2, 1993, pp. 272 – 295.
- [17] 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.

- [18] Fernandez, D. C. D. R. and Zingg, D., “High-Order Compact-Stencil Summation-By-Parts Operators for the Second Derivative with Variable Coefficients,” Tech. Rep. ICCFD7-2803, 7th International Conference on Computational Fluid Dynamics (IC-CFD7), Big Island, HI, July 2012.
- [19] Carpenter, M., Gottlieb, D., and Abarbanel, S., “Time-Staggle Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High-Order Compact Schemes,” ICASE Report 93-9, ICASE, Hampton, VA, March 1993.
- [20] Svård, M., Carpenter, M., and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, far-field boundary conditions,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1020–1038.
- [21] Svård, M. and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, No-slip wall boundary conditions,” *Journal of Computational Physics*, Vol. 227, 2008, pp. 4805–4824.
- [22] Nordström, J., Forsberg, K., Adamsson, C., and Eliasson, P., “Finite Volume Methods, Unstructured Meshes and Strict Stability for Hyperbolic Problems,” *Applied Numerical Mathematics*, Vol. 45, 2003, pp. 453–473.
- [23] Barth, T. J. and Frederickson, P., “Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction,” *AIAA paper* 1990-0013, AIAA 28th Aerospace Sciences Meeting, Reno, NV, January 1990.
- [24] Delanaye, M. and Liu, Y., “Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids,” *AIAA paper* 1995-3259, AIAA 14th CFD Conference, Norfolk, June 1999.
- [25] Ollivier-Gooch, C., Nejat, A., and Michalak, K., “On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes,” *AIAA paper* 2007-4464, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.

- [26] Allmaras, S., Johnson, F., and Spalart, P., “Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model,” Tech. Rep. ICCFD7-1902, 7th International Conference on Computational Fluid Dynamics, July 2012.
- [27] Spalart, P. and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *Recherche Aerospatiale*, Vol. 1, 1994, pp. 5–21.
- [28] Solin, P., Segeth, K., and Dolezel, I., *Higher-Order Finite Element Methods*, 2003.
- [29] Bassi, F. and Rebay, S., “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics*, Vol. 138, December 1997, pp. 251–285.
- [30] Diener, P., Dorband, E., Schnetter, E., and Tiglio, M., “Optimized High-Order Derivative and Dissipation Operators Satisfying Summation by Parts, and Applications in Three-dimensional Multi-block Evolutions,” *Journal of Scientific Computing*, Vol. 32, 2007, pp. 109–145.
- [31] Visbal, M. and Gaitonde, D., “On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes,” *Journal of Computational Physics*, Vol. 181, 2002, pp. 155–185.
- [32] Roache, P., “Code Verification by the Method of Manufactured Solutions,” *Transactions of the ASME*, Vol. 124, 2002, pp. 4–10.
- [33] Roy, C., “Review of Code and Solution Verification Procedures for Computational Simulation,” *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.
- [34] 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.
- [35] Tomboulides, A., Orszag, S., and Karniadakis, G., “Direct and Large Eddy Simulations of Axisymmetric Wakes,” *AIAA paper 93-0546*, 1993.

- [36] Pruppacher, H., Clair, B. L., and Hamielec, A., “Some Relations Between Drag and Flow Pattern of Viscous Flow Past a Sphere and Cylinder at Low and Intermediate Reynolds Numbers,” *Journal of Fluid Mechanics*, Vol. 44, 1970, pp. 781–791.
- [37] Taneda, S., “Experimental Investigation of Wake Behind a Sphere at Low Reynolds Numbers,” *J. Physical Society of Japan*, Vol. 11, 1956, pp. 1104–1108.
- [38] 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.
- [39] Taneda, S., “Visual observations of flow past a sphere at Reynolds numbers between  $10^4$  and  $10^6$ ,” *Journal of Fluid Mechanics*, Vol. 85, 1978, pp. 187–192.
- [40] 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.
- [41] Rumsey, C., “NASA Langley Turbulence Modeling Resource (<http://turbmodels.larc.nasa.gov>),” 2015.

CHAPTER 4  
HIGH-ORDER STRAND GRID METHODS FOR SHOCK TURBULENCE  
INTERACTION

#### 4.1 Abstract

In this work, we examine the flux-correction method for three-dimensional transonic turbulent flows on strand grids. Building upon previous work, we treat flux derivatives along strands with high-order summation-by-parts operators and penalty-based boundary conditions. A finite volume-like limiting strategy is implemented in the flux-correction algorithm in order to sharply capture shocks. To achieve turbulence closure in the Reynolds-Averaged Navier-Stokes equations, a robust version of the Spalart-Allmaras turbulence model is employed that accommodates negative values of the turbulence working variable. Validation studies are considered which demonstrate the flux-correction method achieves a high degree of accuracy for turbulent shock interaction flows.

#### 4.2 Introduction

Obtaining high-fidelity solutions for high-Reynolds number viscous flows over multi-body complex geometry presents a unique set of challenges in computational fluid dynamics (CFD) algorithms today. First, high-quality viscous meshes around multi-body complex geometry can take meshing experts days or even weeks to complete the gridding process using current mesh practices. Improvements in hardware reduce the simulation time for complex simulations. Thus, the percentage of time devoted to mesh generation under current practices will only continue to increase relative to simulation time. Second, traditional numerically diffuse second-order schemes often display accuracy limitations for complex unstructured multi-body grid configurations, however, these schemes are still in use by the majority of CFD practitioners despite the prominence of high-order methods in research.

---

Co-Authors: Yushi Yanagita, Robert Schaap, Shaun Harris, and Aaron Katz.

High-order methods for unstructured grids are not yet at a production level, while standard second-order methods for unstructured grids are easily accessible. Complex systems require ever-increasing mesh sizes, for which scalability becomes a greater issue. Automating viscous mesh generation, preserving spatial and temporal accuracy, and maintaining computational efficiency are currently among the greatest research challenges in CFD today.

The strand-Cartesian approach has shown great potential to alleviate many of these difficulties [1–4]. Strand and Cartesian grids allow the possibility of fully automatic volume grid generation while enhancing scalability and the potential for high-order accuracy. Near solid bodies, 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 6.1(a). Away from the solid bodies, adaptive Cartesian grids shown in Figure 6.1(b) resolve vertical shedding and wake features with efficient high-order algorithms. Due to the robust and automatic nature of the strand-Cartesian grid generation process, the technique is easily extensible to moving-body problems for which the grid can readily be regenerated at each time step. Strand and Cartesian grids communicate through implicit overset interpolation [5–7], which is facilitated by the compact grid representation of the strand-Cartesian mesh system. A typical three-dimensional strand-Cartesian grid system may be stored on each processor in a parallel computation, allowing for self-satisfying domain connectivity [1] and reducing the percentage of time needed for intergrid communication [8].

The primary goal of this work is to demonstrate improved near-body accuracy and efficiency for transonic turbulent flows through high-order finite difference and flux-correction methods in three dimensions. We focus on the strand grid region because even advanced off-body high-order solution techniques may become ineffective if excessive near-body discretization error causes non-physical diffusion of vortical flow features, as shown in recent work by Wissink et al. [9]. While high-order wake-capturing in the off-body is critical for capturing vortex-body interactions, near-body accuracy is critical to properly capture the initial generation of these structures from the surface and to obtaining meaningful body

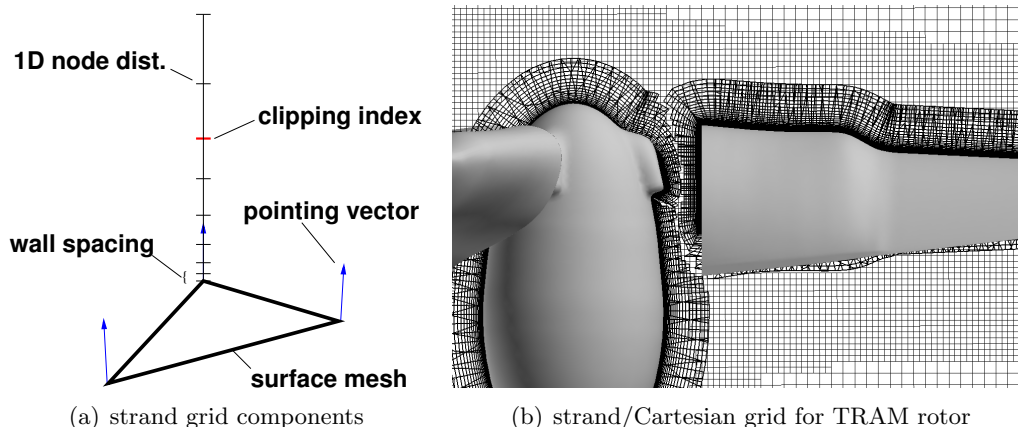


Fig. 4.1: Strand grid elements and example strand/Cartesian grid system for the TRAM rotor.

force computations, especially drag. Thus, this work focuses only on the flux-correction method applied to strand grids. Coupling with an overset Cartesian grid is an eventual goal. Figure 6.1(b) is an example of how strands will eventually be applied with a Cartesian mesh.

The flux-correction method is novel approach of obtaining better than third-order accuracy on strand grids. Previous work on flux-correction [10, 11], paved the way for more recent work [11–13]. The method involves correction of the flux in the unstructured plane, combined with stable summation-by-parts (SBP) operators [14, 15], implemented as source terms to approximate flux derivatives along strands. Unlike most high-order methods under investigation today, the flux-correction method uses a node-centered finite volume method as a starting point to which truncation error-canceling terms are added to increase accuracy. The method requires no additional flux quadrature or second derivatives in the solution reconstruction like quadratic finite volume schemes [16–18]. Two-dimensional and three-dimensional studies indicate the resulting scheme is nearly fourth-order accurate and requires minimal computational overhead beyond second-order schemes [11, 13]. Because the flux correction formulation begins with a finite volume scheme, robust limiting strategies may be employed with ease. This paper aims to examine turbulent transonic flows with the flux-correction method in three-dimensions to demonstrate advanced shock capturing



and limiting capabilities.

The paper is outlined as follows: First, we provide details of the high-order strand grid discretization scheme, including flux-correction and high-order summation-by-parts operators for first derivatives and second derivatives with variable coefficients. Here, we also provide a description of the limiting techniques applied in this work. Next, we present results for a variety of three-dimensional cases. Finally, we draw conclusions based on the existing results and suggest future research directions for the final paper.

### 4.3 High-Order Strand Grid Discretization

In this work, we solve the compressible Reynolds-Averaged Navier-Stokes (RANS) equations in three-dimensions. The “negative” Spalart-Allmaras model [19], which admits negative values of the turbulence working variable, is used to achieve turbulence closure. Negative values of  $\tilde{\nu}$  are often admitted on under-resolved grids, and at the edge of boundary layers and wakes. The combined RANS-SA equations may be expressed as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S, \quad (4.1)$$

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

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho \tilde{\nu} \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ \rho h u_j \\ \rho \tilde{\nu} u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ \frac{\eta}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_j} \end{pmatrix}, \quad (4.2)$$

and the vector of source terms,  $S$ , is defined as

$$S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathcal{P} - \mathcal{D} + c_{b2}\rho \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \end{pmatrix}. \quad (4.3)$$

The turbulent source term for the SA model consists of a production term,  $\mathcal{P}$ , and a destruction term,  $\mathcal{D}$ . Here,  $\rho$  is the density,  $u_j$  is the  $j^{\text{th}}$  component of the fluid velocity,  $p$  is the pressure,  $e$  is the total energy per unit mass,  $h \equiv e + p/\rho$  is the total enthalpy per unit mass,  $\tilde{\nu}$  is the turbulence working variable,  $\sigma_{ij}$  is the deviatoric stress tensor,  $q_j$  is the  $j^{\text{th}}$  component of the heat flux vector, and  $\frac{\eta}{\sigma}$  is the turbulent diffusion coefficient. The stress tensor is defined as

$$\sigma_{ij} = 2(\mu + \mu_T) \left( S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (4.4)$$

where  $\mu$  is the dynamic viscosity,  $\mu_T$  is the turbulent viscosity, and  $S_{ij}$  is the rate of strain tensor, defined as

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (4.5)$$

The heat flux vector is obtained with Fourier's Law, and defined as

$$q_j = -C_p \left( \frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \frac{\partial T}{\partial x_j}, \quad (4.6)$$

where  $T$  is temperature,  $C_p$  is the specific heat,  $Pr$  is the Prandtl number, and  $Pr_T$  is the turbulent Prandtl number. In addition, Sutherland's Law is utilized to relate dynamic viscosity and temperature, and the ideal gas equation of state is used.

The ‘‘negative’’ SA model is designed to use the standard SA model when the turbulent working variable,  $\tilde{\nu}$ , is positive, and adds modifications to the standard model when the turbulent working variable is negative. The details of the standard SA model, including the well-known definitions of the production and destruction terms, may be found in the original work by Spalart and Allmaras [20].

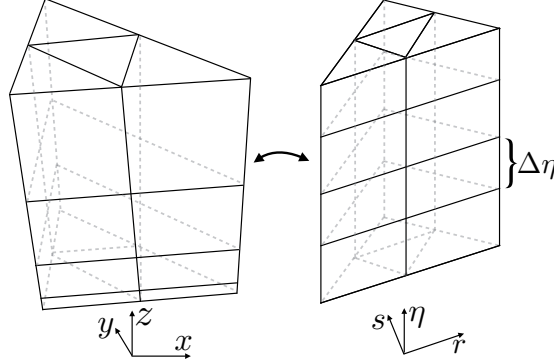


Fig. 4.2: Mapping of strand stack from physical space to computational space.

In this work, Equation 5.1 is solved on strand grids consisting of an unstructured triangular surface tessellation extruded along straight lines (strands) away from solid bodies. Each stack of prismatic cells emanating from the surface in the physical space may be mapped to a standard computational space as shown in Figure 5.2. To facilitate high-order algorithms, the triangular base of each prismatic element may be divided into equally spaced sub-triangles in the  $r$ - $s$  plane. In this work, we investigate up to fourth-order surface elements. Quadratic surface elements are shown in Figure 5.2. The stretched distribution of nodes along each strand in the physical space is mapped to an equally spaced distribution in the  $\eta$ -direction in the computational space, where  $\eta \in (0, 1)$ . In the computational space, the strand spacing is  $\Delta\eta = 1/(N-1)$ , where  $j = 1, \dots, N$  is the strand node numbering beginning with the node on the surface. To avoid confusion, the triangular-shaped elements forming the various levels in the prism stack will be referred to as “surface elements,” while the three-dimensional elements formed by an equally-spaced stacking of the surface elements in the  $\eta$ -direction will be referred to as “volume elements.” Additionally, the triangles formed from sub-dividing each surface element will be referred to as “sub-triangles.”

Upon transformation to the computational space, Equation 5.1 becomes

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} + \frac{\partial \hat{H}}{\partial \eta} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} - \frac{\partial \hat{H}^v}{\partial \eta} = \hat{S}, \quad (4.7)$$

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

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

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

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

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

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

Here,  $J$  is the Jacobian of the transformation,  $\hat{F}_j$  and  $\hat{F}_j^v$  are the transformed inviscid and viscous fluxes, and partial differentiation is denoted with a subscript (e.g.  $\partial x / \partial s = x_s$ ).

In previous work, we explain in detail the novel manner in which a high-order discretization is carried out on strand grids [11]. Here, we focus on significant improvements and refinements to the original method. The general strategy is to perform a high-order finite-volume flux balance in the  $r$ - $s$  plane, known as “flux correction,” and to use high-order finite-differences based on SBP operators in the  $\eta$ -direction. To facilitate the flux-correction algorithm, two-dimensional median-dual control volumes are constructed around each node in each high-order surface element in the  $r$ - $s$  plane. The sub-triangles and median-dual con-

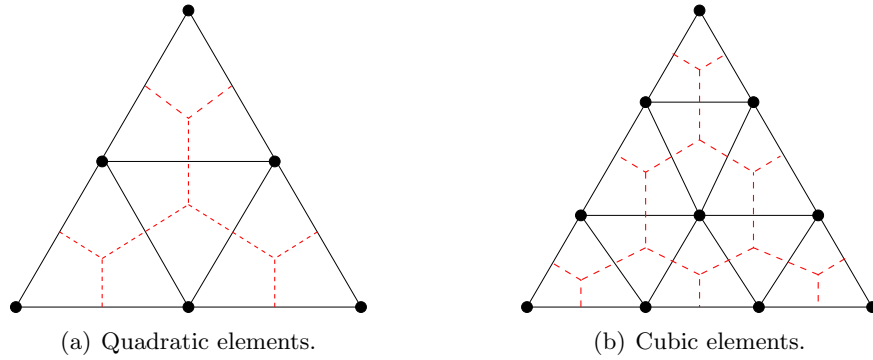


Fig. 4.3: Element mappings used for gradient reconstruction.

control volumes in a single surface element are shown as the black solid lines and red dashed lines in Figure 5.3 for quadratic and cubic surface elements, respectively. While not shown in Figure 5.3, quartic surface elements are used for all applications in this work. It is known that in order for high-order schemes to deliver high orders of accuracy, high-order curved boundary elements are essential [21, 22], which we employ in this work.

A critical aspect of the method is to treat the  $\eta$ -derivatives in Equation 5.8 with a particular source term discretization which preserves the accuracy of the flux-correction procedure in the  $r$ - $s$  plane. Therefore, the  $\eta$ -derivatives and physical time derivative are moved to the right-hand side and treated as source terms:

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} = \tilde{S}, \quad (4.8)$$

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

A pseudo-time derivative is added on the left-hand side of Equation 5.9 to facilitate a semi-implicit time-marching solution [12]. As long as each term in  $\tilde{S}$  is computed to at least second-order accuracy in terms of the truncation error, the corrected flux balance in the  $r$ - $s$  plane will retain desirable truncation error properties, resulting in a high-order accurate scheme. Examining Equation 5.9, the three-dimensional equations essentially reduce to a two-dimensional problem in the  $r$ - $s$  plane at each layer in the strand grid. The layers are coupled together through the new source term,  $\tilde{S}$ , which contains  $\eta$ -derivatives of the flux. Treatment of the  $\eta$ -derivatives is accomplished with SBP operators, along with penalty-based boundary conditions, and is discussed at length elsewhere [14, 15, 23–28], and in our previous work [11]. Below, we focus on the high-order flux correction algorithm as it relates to our new limiter strategy for transonic flows.

Equation 5.9 requires a two-dimensional discretization in the  $r$ - $s$  plane suitable for unstructured grids with a source term. The discretization used for this purpose here is the flux-correction scheme of Katz and Sankaran [10]. Recently, Pincock and Katz [29] extended the original scheme to include the viscous terms in the Navier-Stokes equations. A defining

feature of the flux-correction scheme is its ability to retain high-order accurate truncation error on general simplex grids, which results in third-order discretization error for inviscid fluxes, and fourth-order discretization error for viscous fluxes.

The flux-correction procedure differs from a conventional finite volume scheme in the definition of the numerical fluxes. Following previous work on flux-correction [10], the inviscid numerical fluxes between nodes  $a$  and  $b$  are computed as

$$\hat{\mathcal{F}}_{ab,j} = \frac{1}{2} \left( \hat{\mathcal{F}}_R + \hat{\mathcal{F}}_L \right) - \frac{1}{2} \left| \hat{\mathcal{A}}(Q_R, Q_L) \right| (Q_R - Q_L). \quad (4.9)$$

The unique aspect of the flux-correction scheme is the use of reconstructed fluxes, along with high-order gradient information. The left and right reconstructed fluxes are computed as

$$\hat{\mathcal{F}}_L = \hat{\mathcal{F}}_{a,j} + \frac{1}{2} \Delta \mathbf{r}_{ab}^T \left( \nabla_{rs}^h \hat{\mathcal{F}} \right)_{a,j}, \quad \hat{\mathcal{F}}_R = \hat{\mathcal{F}}_{b,j} - \frac{1}{2} \Delta \mathbf{r}_{ab}^T \left( \nabla_{rs}^h \hat{\mathcal{F}} \right)_{b,j}, \quad (4.10)$$

where  $\Delta \mathbf{r}_{ab}^T = (r_b - r_a, s_b - s_a)$ , and  $\nabla_{rs}^h$  is an estimate of the gradient in the  $r$ - $s$  plane computed to at least second-order accuracy, such that

$$\nabla_{rs}^h = \nabla_{rs} + O(h^q), \quad q \geq 2.$$

The strategy to compute such gradients is to compute the derivatives of a local polynomial representation within each surface element, and then volume-average among the shared representations in neighboring elements. The averaging is critical to centering the gradients and to the overall stability of the scheme. The requirement that  $q \geq 2$  may be satisfied by using at least quadratic surface elements shown in Figure 5.3. Further details of the gradient procedure used for the flux and solution may be found in our previous work [11].

With this background established, the new focus of this paper is to assess the ability of the high-order strand grid scheme to capture shocks present in high-Reynolds number turbulent flows. While simple and robust limiting schemes have been developed for second-order CFD methods, discontinuous flows continue to present significant challenges for high-order methods due to Gibbs oscillations. Many approaches have been proposed

to handle these flows in a high-order context. The most popular of these approaches include specialized limiting techniques [30] and artificial dissipation [31,32]. Both approaches require substantially different formulations than traditional monotonicity-preserving limiters used in second-order schemes. In the case of artificial dissipation, often an additional equation for artificial viscosity is added, which increases the expense and complexity of the scheme. Overall, there appears to be no consensus among the community regarding the best approach for capturing shocks with a high-order method.

One advantage of our flux-correction high-order scheme is that it is possible to implement traditional finite volume limiters to locally introduce first-order dissipation at shocks. Here, we explore such an approach. Just as in finite volume schemes, the form of artificial dissipation in Equation 5.10 includes a difference of left and right states,  $Q_L$  and  $Q_R$ . For smooth flows, we reconstruct these states using gradients with second order accuracy or better. To handle discontinuous flows, we multiply these gradients by a limiter, which locally switches the left and right states to the nodal values, resulting in a first-order dissipation term. The reconstructed states with limiter,  $\phi$ , may be expressed as,

$$Q_L = Q_{a,j} + \frac{\phi_{ab}}{2} \Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{a,j}, \quad Q_R = Q_{b,j} - \frac{\phi_{ab}}{2} \Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{b,j}. \quad (4.11)$$

Here,  $\phi_{ab}$  is the limiter at the edge in the  $r$ - $s$  plane connecting nodes  $a$  and  $b$ . Similar to the symmetric limited positive (SLIP) scheme of Jameson [33,34], we compute the limiter as,

$$\phi_{ab} = 1 - \left| \frac{u - v}{\max(|u| + |v|, \epsilon)} \right|^3, \quad u = 2\Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{b,j} - \Delta Q_{ab}, \quad v = 2\Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{a,j} - \Delta Q_{ab}, \quad (4.12)$$

where  $\epsilon$  is a small number to avoid limiting at smooth extrema or dividing by zero in freestream flow, and  $\Delta Q_{ab} = Q_b - Q_a$  is the solution difference across the edge. The  $r$ - $s$  gradients in Equation 5.16 are calculated using the same method used to obtain the other flux and solution gradients in the FC scheme. Therefore, no additional MPI communication is required, and little extra computational effort or complexity is added.

To date, we have successfully computed several shocked flows with the above method, as described in the next section. Further tests are required to assess the limiter in terms of robustness and accuracy. In particular, we need to evaluate the limiter for cases in which discontinuities lie perpendicular to strands, such as for a bow shock. In this case, it may be necessary to detect discontinuities along the strand direction in addition to the  $r$ - $s$  plane, as described by Equation 5.16. Nonetheless, the success in computing turbulent shocked flows with a high-order method thus far is significant, especially in light of the relatively simple SLIP formulation afforded by the flux-correction scheme.

## 4.4 Results

The ability of the high-order flux-correction method to accurately capture shocks for high Reynolds number turbulent flow using strand grids is investigated in this section. Three configurations are examined for validation with extensive use of the NASA-Langley turbulence modeling resource [35]: a bump in a channel, an infinite wing, and a hemisphere-cylinder configuration. Each case is run at subsonic conditions first, followed by transonic conditions. The cases examined make use of strand grids only. When available, results from independent compressible CFD codes (CFL3D, FUN3D) are used to aid validation. Experimental data is also used for comparison when available.

### 4.4.1 Bump-in-Channel

The first case we examine is a bump-in-channel. For this geometry, we examine two different flow conditions. First, we run a subsonic turbulent flow case to validate the solver using the NASA-Langley turbulence [35] as a means of comparison. Next, we run a qualitative transonic turbulent flow study to examine the shock capturing capabilities over the bump-in-channel. The body reference length is 1.5 units, where the lower wall is a viscous-wall bump extending from  $x = 0$  to 1.5. The maximum bump height is  $y = 0.05$ . The upstream and downstream far-field extends 25 units from the viscous-wall, with symmetry boundary conditions imposed on the lower wall between the far-field and the solid wall. The upper boundary is a distance of  $y = 5.0$  high, and is set as an inviscid plane.



Each side of the channel is also set as an inviscid plane.

Fourth-order elements are used to create sub-triangles within each parent element. The surface grid has a size of 14,473 nodes, with 96 nodes in the strand direction, totaling approximately 1.4 million nodes in the volume. The surface mesh is shown in Figure 4.4(a), and the volume mesh is shown in Figure 4.4(b). To avoid internal crossing issues over the bump, strands are fixed vertically, and are not smoothed or skewed.

### Subsonic-Turbulent Flow

First, we run a Mach number of  $M = 0.2$  and a Reynolds number of  $Re = 3 \times 10^6$  based on a grid length of unity. A further description and layout of the case may be found on the NASA-Langley turbulence website [35]. Contour plots of velocity and pressure are shown in Figure 4.5. Stream-wise velocity and turbulent viscosity profiles are shown for two locations downstream on the bump, and are over-plotted with FUN3D and CFL3D results in Figure 4.6. Excellent agreement is seen in both cases. The computed drag coefficient is shown in Table 5.3. The drag coefficient falls within the range predicted by the established codes.

### Transonic-Turbulent Flow

Next, we run a Mach number of  $M = 0.8$ , at a Reynolds number of  $Re = 3 \times 10^6$  based on a grid length of unity. Contour plots of velocity and pressure are shown in Figure 4.8. Shock-induced separation and recirculation is evident behind the bump in Figure 4.8(a). The lack of validation or experimental data for these flow conditions makes further comparisons difficult. However, qualitatively we can see in both contour plots that the shock has been captured clearly with no overshoots, indicating the successful functioning

Table 4.1: Comparison of computed drag coefficients for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

Data Source	$C_d$
Strand FC	3.58300E-3
FUN3D	3.56106E-3
CFL3D	3.57238E-3

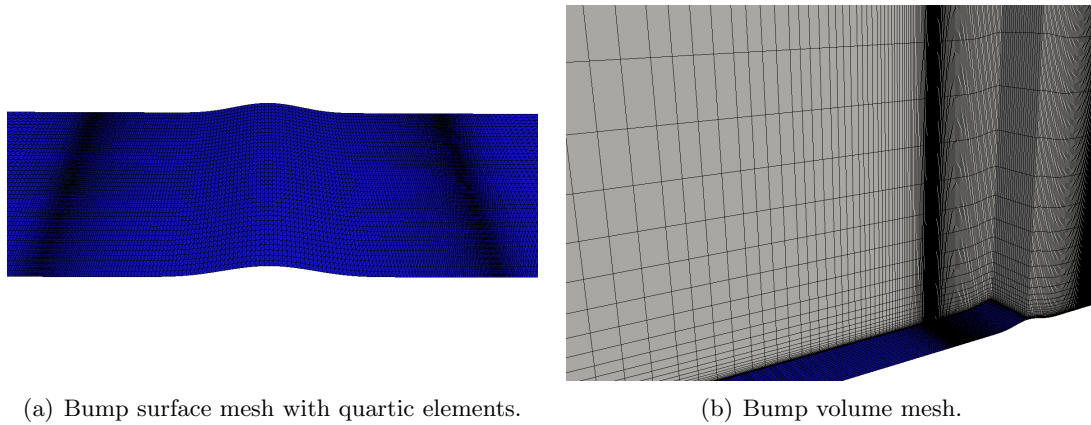
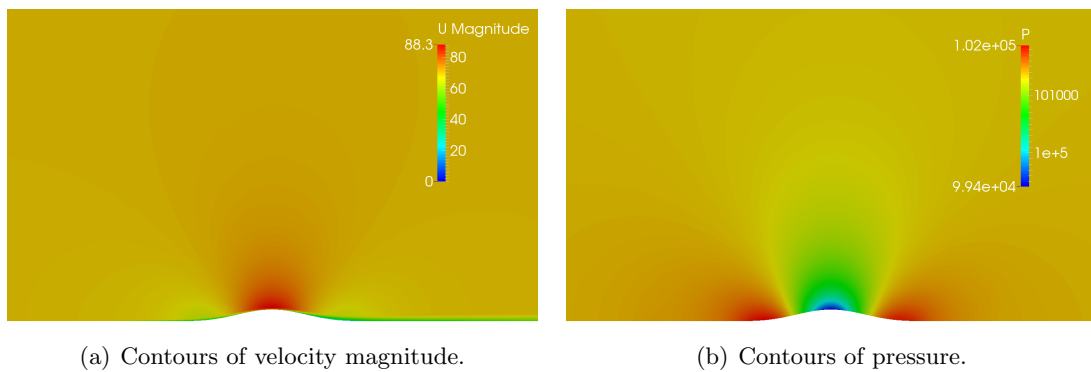
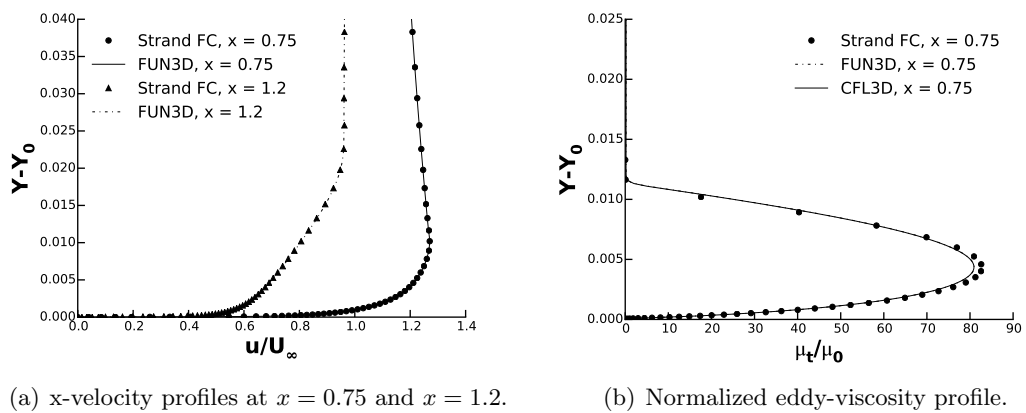
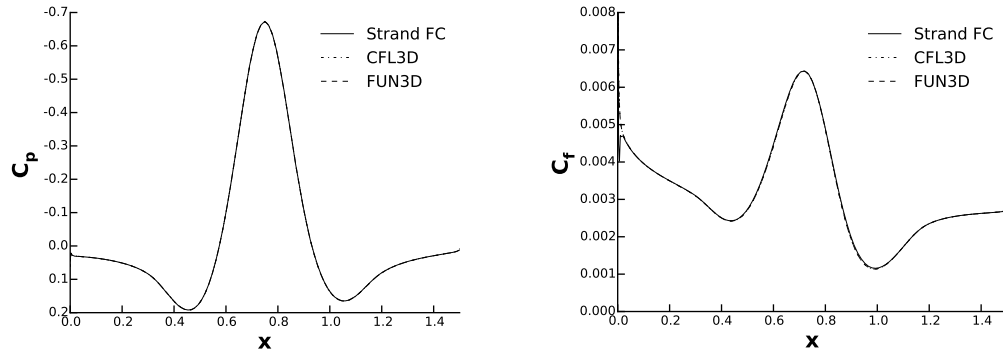


Fig. 4.4: Bump-in-channel mesh configuration.

Fig. 4.5: Field plots of the bump-in-channel at  $M = 0.2$ .Fig. 4.6: Comparison of stream-wise velocity and turbulent viscosity profiles for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .



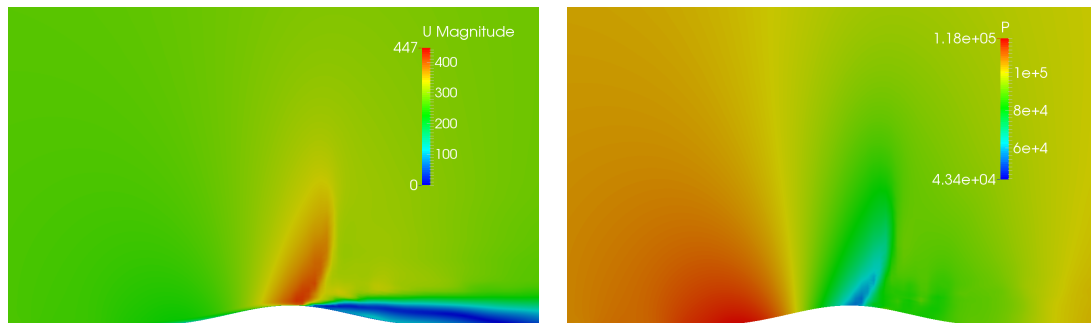
(a) Coefficient of pressure along the surface. (b) Coefficient of friction along upper surface.

Fig. 4.7: Coefficient of pressure and friction for flow over the bump at  $M = 0.2$ .

of the limiter.

#### 4.4.2 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 mesh skewing. Again, we examine two flow regimes: subsonic and transonic turbulent flow. For this case, only strands are used, and no overset Cartesian meshing is employed. The surface mesh consists of 1,280 fourth-order elements and 16,064 nodes, shown in Figure 4.9(a). We set 64 nodes



(a) Contours of velocity magnitude.

(b) Contours of pressure.

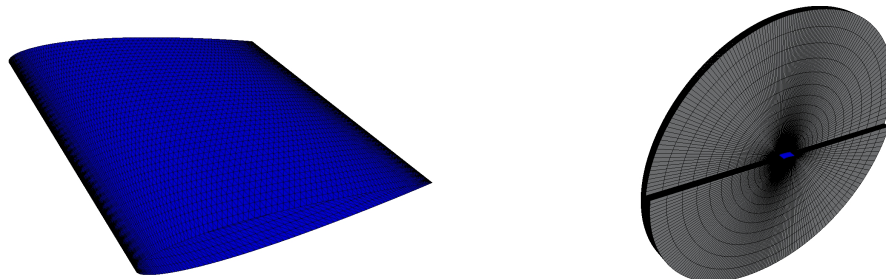
Fig. 4.8: Field plots of the bump-in-channel at  $M = 0.8$ .

along each strand, which extend 10 chords, resulting in a volume mesh with approximately 1 million nodes, shown in Figure 4.9(b). The wing has a chord length of  $c = 1$  and a span of  $S = 1$ . For both cases studied, the the boundary conditions consist of a far-field boundary condition applied on the outer boundary, wall conditions on the surface of the airfoil, and symmetry planes on the sides of the wing.

### Subsonic-Turbulent Flow

We first validate the flux-correction method for subsonic turbulent flow over this airfoil, as described in the NASA Langley Turbulence Modeling Resource. 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 [36], and Gregory and O'Reilly [37] are used for validation. A further description and layout of the case may be found on the NASA-Langley turbulence website [35].

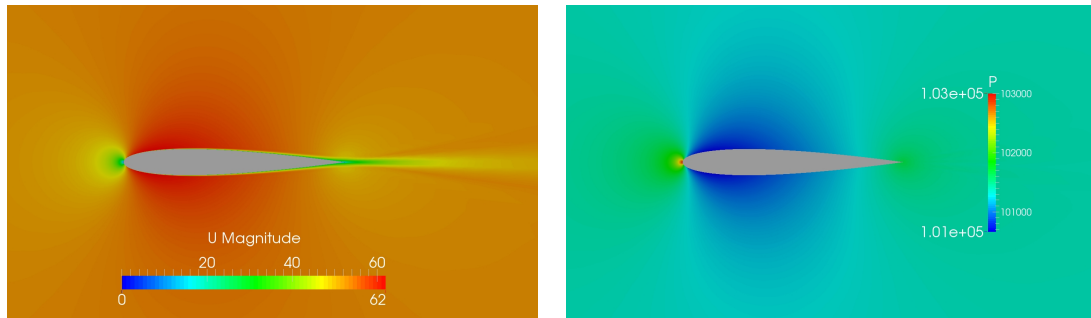
Figure 4.10 shows contours of velocity and pressure around the airfoil. The surface pressure coefficient and coefficient of friction for  $\alpha = 0$  degrees are shown in Figure 4.11, and compared with the experimental data of Gregory and O'Reilly [37], as well as data from CFL3D [35]. The Gregory 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 solver use a very fine two-dimensional C-grid, totaling 274,329 nodes. In



(a) NACA 0012 surface mesh with quartic elements.

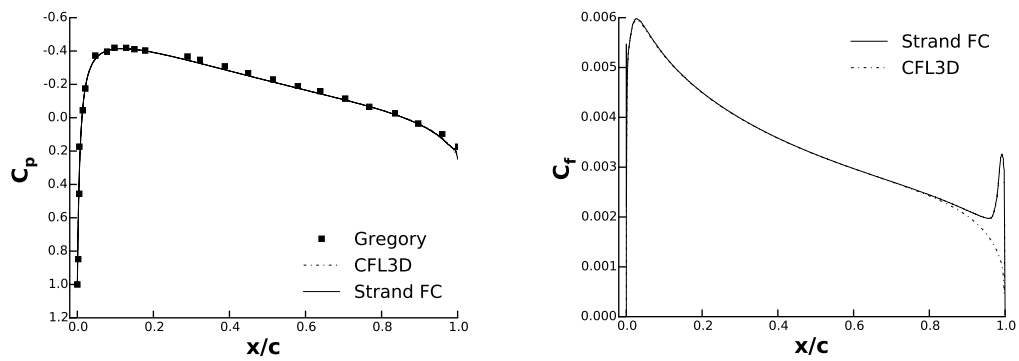
(b) NACA 0012 volume mesh.

Fig. 4.9: NACA 0012 mesh configuration.



(a) Contours of velocity magnitude.

(b) Contours of pressure.

Fig. 4.10: 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. 4.11: Coefficient of pressure and friction for flow over a NACA 0012 airfoil at  $M = 0.15$  and  $Re = 6 \times 10^6$  at  $\alpha = 0$  degrees.

contrast, the three-dimensional grid used for this problem does not make specific refinements for the wake. The calculated drag from the case falls close to the data provided from the NASA Langley turbulence resource [35], and to the experimental data of Ladson [36]. A likely reason for the minor discrepancy between the Strand FC drag and the experimental is the lack of an off-body wake-refining Cartesian grid. Inaccuracies of this sort using strand grids were also observed previously by Work et al. [38]. This is demonstrated in Figure 4.11(b), where the coefficient of friction follows the curve of the CFL3D result until the trailing edge where the  $C_f$  exhibits a peak.

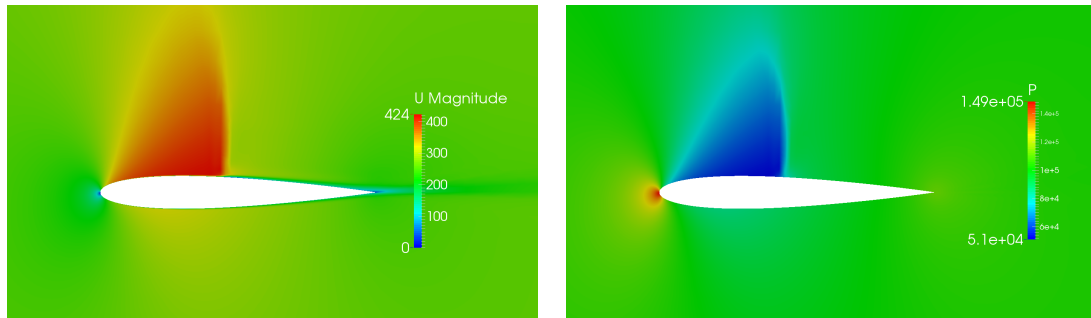
### Transonic-Turbulent Flow

With the NACA 0012 subsonic turbulent case validated, we now test the limiting techniques within in the flux-correction method to predict turbulent shock interaction over an infinite NACA 0012 wing. We use a case set up by McDevitt and Okuno [39], consisting of flow at  $M = 0.759$  and  $Re = 6.3 \times 10^6$  at an angle of attack of  $\alpha = 2.05$  degrees.

Figure 4.12 shows contours of velocity and pressure around the airfoil. The contours of velocity magnitude show a shock induced separation of the boundary layer, effectively altering the shape to the airfoil, similar to the upward deflection of a trailing edge flap, as shown in Figure 4.13(b). This behavior concurs with that observed by McDevitt and Okuno [39]. The calculated flux-correction scheme surface pressure coefficient with the limiter is shown in Figure 4.13(a), and compared with the experimental results from McDevitt and Okuno [39]. With the limiter, good agreement is observed, specifically at the shock location on the upper surface. Moreover, the shock is captured sharply with no overshoots.

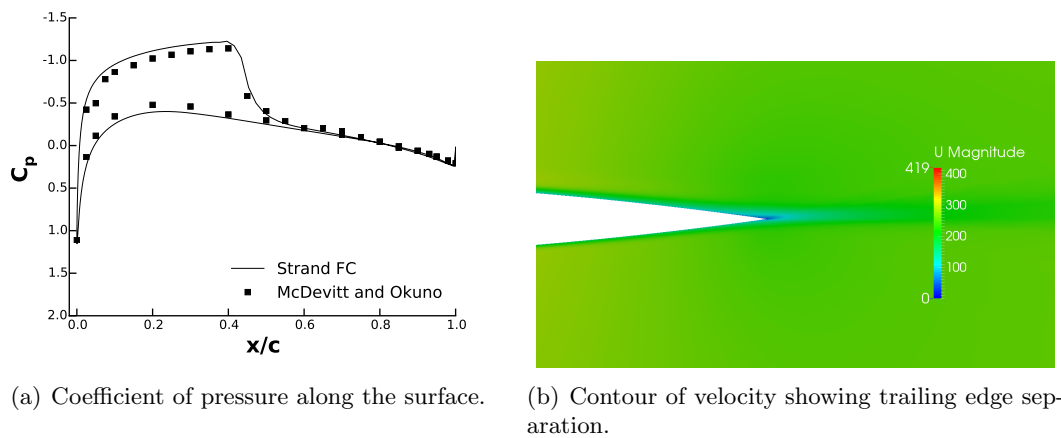
Table 4.2: Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at  $M = 0.15$ ,  $\alpha = 0$  degrees, and  $Re = 6 \times 10^6$ .

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



(a) Contours of velocity magnitude.

(b) Contours of pressure.

Fig. 4.12: Field plots of the hemisphere-cylinder at  $M = 0.759$ .

(a) Coefficient of pressure along the surface.

(b) Contour of velocity showing trailing edge separation.

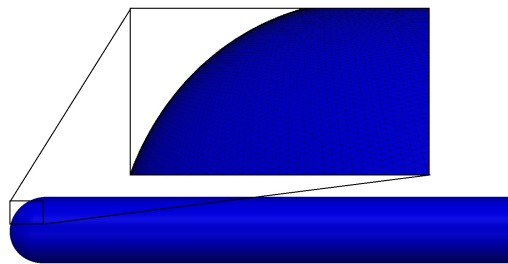
Fig. 4.13: Coefficient of pressure and trailing edge separation for a NACA 0012 airfoil at  $M = 0.759$  and  $Re = 6.3 \times 10^6$  at  $\alpha = 2.05$  degrees.

### 4.4.3 Transonic Hemisphere-Cylinder

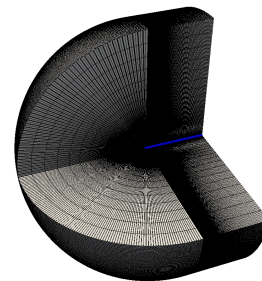
The final case study we present is transonic flow over a hemisphere-cylinder configuration. Geometry is taken from work by Hsieh [40], where the cylinder has a radius of 0.5 and a length of 10. In recent work, Tong et al. [13] validated the flux-correction method for this case with subsonic turbulent flow conditions. Flow conditions consist of  $M = 0.85$  and unit length Reynolds number of  $Re = 4 \times 10^5$  at an angle of incidence of 0 degrees. Further details and grids may be found on the NASA-Langley turbulence resource website [35] and in Hsieh [40].

Figure 4.14 shows the hemisphere-cylinder surface and volume mesh configuration. The surface mesh, shown in Figure 4.14(a) is tessellated with 19,050 quartic elements, totaling of approximately 152,701 surface nodes. The strand grid volume mesh, shown in Figure 4.14, consists of 64 nodes along the strands extending for a distance of 40 diameters, bringing the total mesh count to around 9.5 million nodes. At run time, we partition the grid using 192 cores.

Field plots of velocity magnitude and pressure may be found in Figure 4.15. An oblique shock along with a separation bubble can be seen in Figure 4.15(a) which shows velocity magnitudes surrounding the hemisphere-cylinder body. The formation of the shock and separation bubble are consistent with the experimental results found by Hsieh [40]. Next, we examine non-dimensional velocity profiles at various locations along the hemisphere



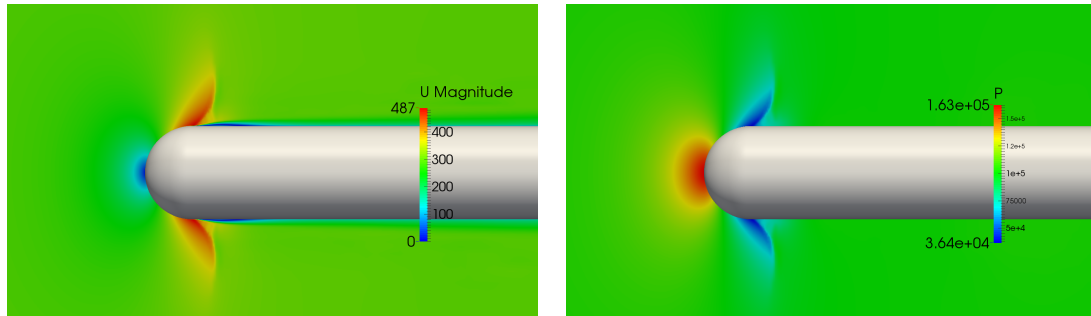
(a) Hemisphere-cylinder surface mesh with quartic elements.



(b) Hemisphere-cylinder volume mesh.

Fig. 4.14: Hemisphere-cylinder mesh configuration.





(a) Contours of velocity magnitude.

(b) Contours of pressure.

Fig. 4.15: Field plots of the hemisphere-cylinder at  $M = 0.85$ .

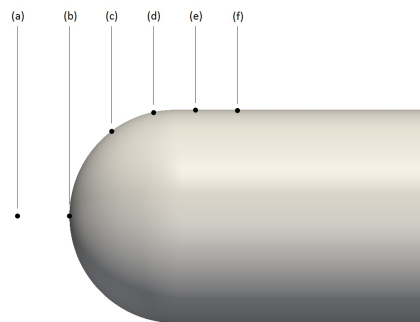


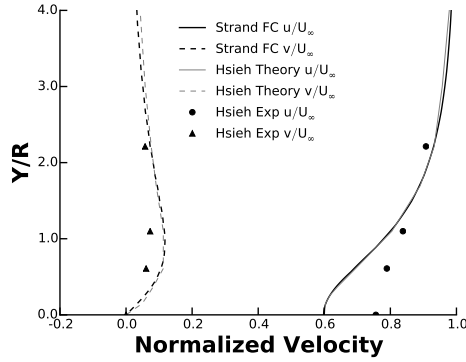
Fig. 4.16: Location of velocity profiles along the hemisphere-cylinder.

cylinder, shown by letter location in Figure 4.16. Each of the lettered locations corresponds to the lettered subfigures in Figure 4.17, which show the velocity profiles. Here, we compare the flux-correction method to experimental results [40], and to theoretical results obtained with potential flow theory [41]. 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 [40] claims that this deviation is due to particle lag from the laser doppler velocimetry method used to obtain the results. Nonetheless, the flux correction results generally agree well with the theory here.

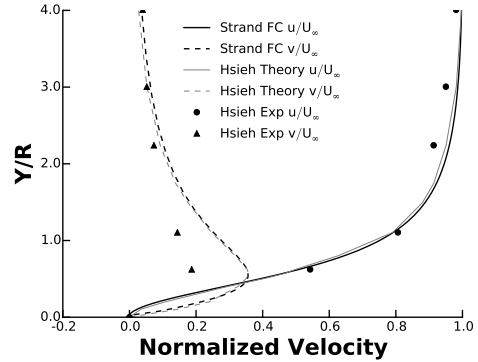
#### 4.5 Conclusions and Future Work

A novel high-order algorithm for strand grids has been presented for turbulent shock interaction flows. The method uses a combination of summation-by-parts operators in the strand direction and flux-correction in the unstructured layers to achieve high-fidelity solutions of compressible turbulent flow. Low-order truncation errors are cancelled via accurate flux and solution gradient terms in the flux-correction method, thereby achieving a formal order of accuracy of three, although higher orders are often observed, especially for highly viscous flows. To achieve turbulence closure in the RANS equations, a robust version of the Spalart-Allmaras turbulence model was employed that accommodates negative values of the turbulence working variable. A major advantage of our high-order formulation is the ability to implement traditional finite volume-like limiters to cleanly capture shocks. We explored this approach in this work via a symmetric limited positive (SLIP) limiter. The ability of our high-order method to handle shocks via SLIP limiters is the major focus of the paper.

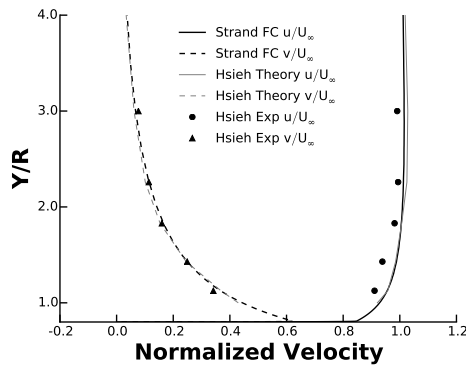
When applied to a three-dimensional bump in a channel, the flux-correction method



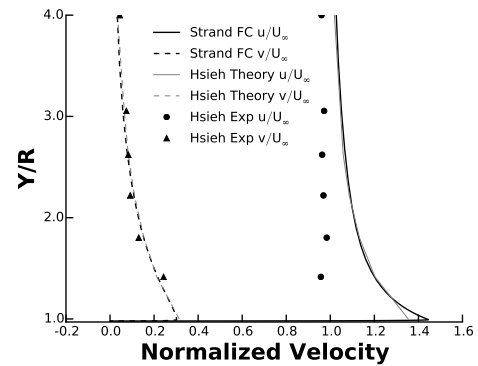
(a) Velocity profile in y-direction at  $z = -0.5$ .



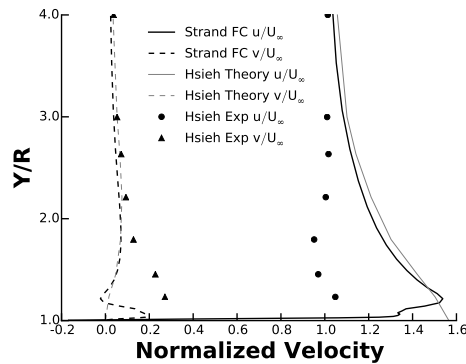
(b) Velocity profile in y-direction at  $z = 0.0$ .



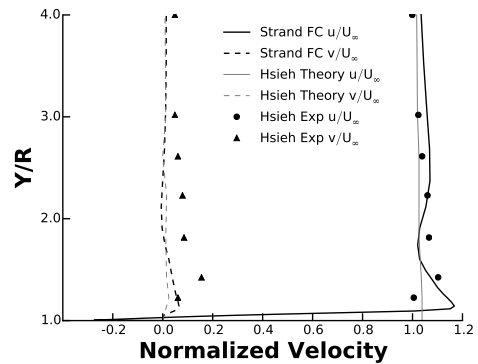
(c) Velocity profile in y-direction at  $z = 0.4$ .



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



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



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

Fig. 4.17: Velocity field about hemisphere-cylinder at various  $z$  locations.

accurately predicted the drag for turbulent subsonic flow. When applied to turbulent transonic flow, the flux-correction method qualitatively showed the ability to capture shocks cleanly. When applied to turbulent flow over a NACA 0012 airfoil, the flux-correction method accurately predicted the drag for turbulent subsonic flow. When applied to transonic flow, the coefficient of pressure along the surface of the airfoil was accurately predicted, including the shock location. Again, sharp and clean shocks were observed in the results. The ability of the flux-correction algorithm to resolve turbulent shock interaction flows was demonstrated using a hemisphere-cylinder configuration. The velocity profiles extending radially at six different locations along the cylinder showed similar profiles to those found theoretically and experimentally by Hsieh [40].

While this work appears promising in terms of accurately computing complex turbulent flows, the test cases presented here are rather simple geometrically. Future work will focus on extending our high-order methods to geometries of practical interest. Two research areas are envisioned in this effort. First, we intend to couple the strand grids to off-body Cartesian grids in an overset fashion. While overset grid methods are certainly not new, specific challenges may arise in terms of stability and accuracy with our high-order schemes. The effect of high-order interpolation and energy stable interface conditions will be investigated. Second, we intend to investigate novel ways of representing and handling complex surface geometry. The geometry surface serves as the starting point for strand growth and off-body mesh generation. Therefore, the ability to handle any geometry with complex combinations of convex, concave, and saddle regions, is of prime interest. One possible approach would be to consider surface geometry as the zero level set of a higher-dimensional function. Along with variable strand lengths, this could facilitate small surface modifications and adaptations that would allow for the robust treatment of any geometry with the strand method.

## **Acknowledgments**

Development was performed with the support of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by

the U.S. Department of Defense HPC Modernization Program Office, and by the Office of Naval Research Sea-Based Aviation program directed by Dr. Judah Milgram and Mr. John Kinzer.

**REFERENCES**

- [1] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [2] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [3] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA paper* 2009-3792, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [4] Katz, A., Wissink, A., Sankaran, V., Meakin, R., and Sitaraman, J., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 6512–6530.
- [5] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.
- [6] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [7] Lee, Y.-L. and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [8] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.

- [9] Wissink, A., “Helios Solver Developments Including Strand Meshes,” Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.
- [10] 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.
- [11] 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.
- [12] Tong, O., Work, D., and Katz, A., “High-Order Methods for Turbulence Using Strand Grids,” Tech. Rep. ICCFD8-0215, 8th International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 2014.
- [13] Tong, O., Katz, A., Yanagita, Y., Casey, A., and Schaap, R., “High-Order Methods for Three-Dimensional Strand Grids,” *Journal of Scientific Computing*, Vol. 64, 2015.
- [14] 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.
- [15] Fernandez, D. C. D. R. and Zingg, D., “High-Order Compact-Stencil Summation-By-Parts Operators for the Second Derivative with Variable Coefficients,” Tech. Rep. ICCFD7-2803, 7th International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, HI, July 2012.
- [16] Barth, T. J. and Frederickson, P., “Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction,” *AIAA paper* 1990-0013, AIAA 28th Aerospace Sciences Meeting, Reno, NV, January 1990.
- [17] Delanaye, M. and Liu, Y., “Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids,” *AIAA paper* 1995-3259, AIAA 14th CFD Conference, Norfolk, June 1999.

- [18] Ollivier-Gooch, C., Nejat, A., and Michalak, K., “On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes,” *AIAA paper* 2007-4464, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [19] Allmaras, S., Johnson, F., and Spalart, P., “Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model,” Tech. Rep. ICCFD7-1902, 7th International Conference on Computational Fluid Dynamics, July 2012.
- [20] Spalart, P. and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *Recherche Aerospaciale*, Vol. 1, 1994, pp. 5–21.
- [21] Solin, P., Segeth, K., and Dolezel, I., *Higher-Order Finite Element Methods*, 2003.
- [22] Bassi, F. and Rebay, S., “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics*, Vol. 138, December 1997, pp. 251–285.
- [23] Kreiss, H. and Scherer, G., “Finite element and finite difference methods for hyperbolic partial differential equations,” *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. D. Boor, Academic Press, 1974.
- [24] Strand, B., “Summation by parts for finite difference approximation for  $d/dx$ ,” *Journal of Computational Physics*, Vol. 110, 1994, pp. 47–67.
- [25] Carpenter, M. H., Gottlieb, D., and Abarbanel, S., “The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes,” *Journal of Computational Physics*, Vol. 108, No. 2, 1993, pp. 272 – 295.
- [26] Carpenter, M., Gottlieb, D., and Abarbanel, S., “Time-Staggle Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High-Order Compact Schemes,” ICASE Report 93-9, ICASE, Hampton, VA, March 1993.



- [27] Svård, M., Carpenter, M., and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, far-field boundary conditions,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1020–1038.
- [28] Svård, M. and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, No-slip wall boundary conditions,” *Journal of Computational Physics*, Vol. 227, 2008, pp. 4805–4824.
- [29] 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.
- [30] Wang, Z., “Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids,” *Journal of Computational Physics*, Vol. 178, 2002, pp. 210–251.
- [31] Nguyen, N. and Peraire, J., “An Adaptive Shock-Capturing HDG Method for Compressible Flows,” *AIAA paper* 2011-3060, AIAA 20th CFD Conference, Honolulu, June 2011.
- [32] Persson, P., “Shock Capturing for High-Order Discontinuous Galerkin Simulation of Transient Flow Problems,” *AIAA paper* 2013-3061, AIAA 21st CFD Conference, San Diego, June 2013.
- [33] 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.
- [34] 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.
- [35] Rumsey, C., “NASA Langley Turbulence Modeling Resource (<http://turbmodels.larc.nasa.gov>),” 2015.

- [36] 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.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] 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.

CHAPTER 5  
ASSESSMENT OF A TWO-EQUATION TURBULENCE MODEL IN THE  
HIGH-ORDER FLUX CORRECTION SCHEME

### 5.1 Abstract

In this work, we examine a two-equation turbulence model in the flux correction method for three-dimensional turbulent flows on strand grids. Building upon previous work, flux derivatives along strands are treated with high-order summation-by-parts operators and penalty-based boundary conditions. To achieve turbulence closure in the Reynolds-Averaged Navier-Stokes equations, the two-equation Menter SST  $k$ - $\omega$  turbulence model is employed. Oscillations caused by the large specific dissipation rate viscous wall boundary condition are damped with selected techniques and the symmetric limited positive scheme of Jameson. Verification and validation studies are considered and demonstrate the flux correction method achieves a high degree of accuracy for the Menter SST turbulence model without any oscillations in the solution. High-order turbulent flux correction results demonstrate improvements in accuracy with minimal computational and algorithmic overhead over traditional second-order algorithms.

### 5.2 Introduction

In computational fluid dynamics (CFD), obtaining high-fidelity solutions for high-Reynolds number flows over complex multi-body geometry remains one of the greatest challenges. First, the gridding process for complex multi-body geometry, such as rotorcraft, can take experts days or weeks to complete using current mesh practices. This current trend will only be exacerbated by projected improvements in hardware, further increasing the percentage of time devoted mesh generation relative to total simulation time. Second, accuracy limitations are often observed due to the use of traditional numerical-diffuse second-order

---

Co-Authors: Cole Blakely, Robert Schaap, and Aaron Katz.

schemes. Despite increasing levels of research into high-order methods, these methods are not yet at a production level for unstructured grids, while second-order methods are easily accessible. Thus, the majority of CFD practitioners still adopt traditional second-order methods. Finally, complex systems require ever-increasing mesh sizes, for which scalability becomes a greater issue. Automating viscous mesh generation, preserving spatial and temporal accuracy, and maintaining computational efficiency are currently among the greatest research challenges in CFD today.

The strand-Cartesian approach has shown great potential to alleviate many of these difficulties [1–4]. Strand and Cartesian grids allow the possibility of fully automatic volume grid generation while enhancing scalability and the potential for high-order accuracy. Near solid bodies, 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 6.1(a). Away from solid bodies, adaptive Cartesian grids resolve vortical shedding and wake features with efficient high-order algorithms, shown in Figure 6.1(b). Due to the robust and automatic nature of the strand-Cartesian grid generation process, the technique is easily extensible to moving-body problems for which the grid can readily be regenerated at each time step. Strand and Cartesian grids communicate through implicit overset interpolation [5–7], which is facilitated by the compact grid representation of the strand-Cartesian mesh system. A typical three-dimensional strand-Cartesian grid system may be stored on each processor in a parallel computation, allowing for self-satisfying domain connectivity [1] and reducing the percentage of time needed for intergrid communication [8].

The primary goal of this work is to demonstrate improved near-body accuracy and efficiency for turbulent flows through high-order flux correction methods in three-dimensions. Flux correction is a novel method of obtaining near fourth-order accuracy on strand grids. Previous methods of “flux correction” [9, 10] have paved the way for recent work investigating the method [11–14]. The high-order strand method involves correction of the flux in the unstructured plane, combined with stable summation-by-parts (SBP) operators [15, 16],

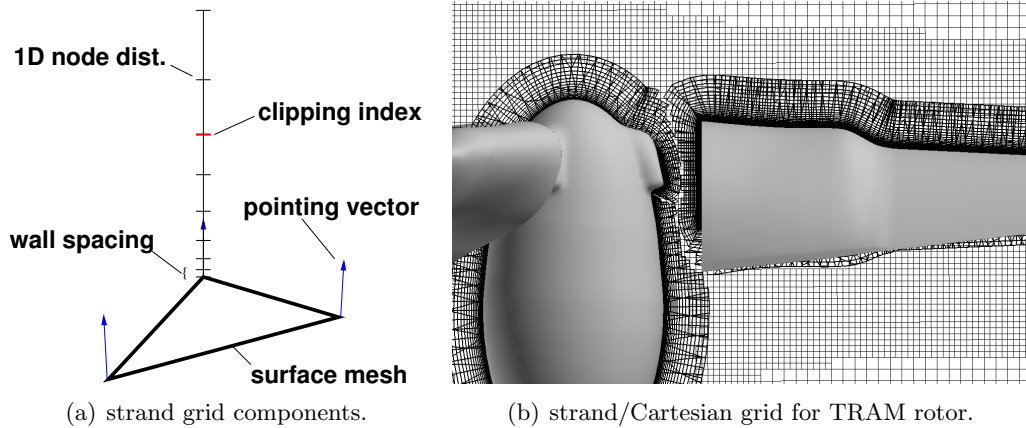


Fig. 5.1: Strand grid elements and example strand/Cartesian grid system for the TRAM rotor.

implemented as source terms to approximate flux derivatives along strands. Unlike most high-order methods under investigation today, the flux correction method uses a node-centered finite volume method as a starting point to which truncation error-canceling terms are added to increase accuracy. The method requires no additional flux quadrature or second derivatives in the solution reconstruction like quadratic finite volume schemes [17–19]. High-order near-body solution methods for turbulent flows are investigated as the accuracy of even the most advanced adaptive high-order off-body methods can be spoiled by excessive near-body discretization error, as shown by Wissink et al. [20]. Thus, in this work, we focus only on the flux correction method for strand grids. Coupling with an off-set Cartesian grid is an eventual goal and is not addressed in this work. Figure 6.1(b) shows an example of the goal we are currently working towards.

Resolving highly turbulent flows in the near-body is critical for capturing the initial generation of vortical structures, however, the choice of turbulent closure for the RANS equations can greatly affect the solution accuracy. The Spalart-Allmaras [21, 22] (SA) turbulence model is generally the model of choice due to its accuracy, simplicity, and ease of implementation, however, it is known that it fails to provide accurate prediction for highly separated flows [23]. The Menter shear stress transport (SST) two-equation turbulence model [24, 25] has shown excellent prediction of highly-separated flows, but has yet to be

embraced as a primary method for turbulence closure in high-order algorithms. While some work has been made to implement the SST model into high-order schemes [26, 27], it is generally avoided for two reasons. First, an extremely large specific dissipation rate boundary condition on viscous walls is a source of instability and oscillations in the solution. Second, the model switch at the edge of the boundary layer creates a discontinuity in the solution, leading to oscillations when a high-order scheme attempts to resolve this. This paper aims to extend the flux correction methodology to the SST turbulence model and damp oscillations with the symmetric limited positive (SLIP) scheme of Jameson [28, 29].

The paper is outlined as follows: First, we provide details of the high-order strand grid discretization scheme, including flux correction and high-order summation-by-parts operators for first derivatives and second derivatives with variable coefficients. Second, we give a brief description of implementation details and methods used to overcome any oscillatory behavior caused by the boundary conditions. Next, we present results for a variety of three-dimensional cases. Finally, we draw conclusions based on the existing results and suggest future research directions for the final paper.

### 5.3 High-Order Strand Grid Discretization

In this work, we solve the compressible Reynolds-Averaged Navier-Stokes (RANS) equations in three-dimensions, and use the Menter SST  $k-\omega$  (SST) model [24] to achieve turbulence closure. The model uses blending functions to utilize the strengths of both the  $k-\epsilon$  and  $k-\omega$  models. The RANS-SST equations may be expressed as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S, \quad (5.1)$$

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

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho k \\ \rho \omega \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ \rho h u_j \\ \rho k u_j \\ \rho \omega u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \\ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \end{pmatrix}, \quad (5.2)$$

and the vector of source terms,  $S$ , is defined as

$$S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathcal{P} - \mathcal{D}_k \\ \frac{\gamma}{\nu_t} \mathcal{P} - \mathcal{D}_\omega + 2(1 - F_1) \frac{\rho \sigma_\omega 2}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \end{pmatrix}. \quad (5.3)$$

Here,  $\rho$  is the density,  $u_j$  is the  $j^{\text{th}}$  component of the fluid velocity,  $p$  is the pressure,  $e$  is the total energy per unit mass,  $h \equiv e + p/\rho$  is the total enthalpy per unit mass,  $\sigma_{ij}$  is the deviatoric stress tensor,  $q_j$  is the  $j^{\text{th}}$  component of the heat flux vector,  $k$  is the turbulent kinetic energy,  $\omega$  is the specific rate of dissipation, and  $\gamma$ ,  $\sigma_k$ , and  $\sigma_\omega$  are blended coefficient functions. The stress tensor is defined as

$$\sigma_{ij} = 2(\mu + \mu_T) \left( S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij}, \quad (5.4)$$

where  $\mu$  is the dynamic viscosity,  $\mu_T$  is the turbulent viscosity, and  $S_{ij}$  is the rate of strain tensor, defined as

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (5.5)$$

The heat flux vector is obtained with Fourier's Law, and defined as

$$q_j = -C_p \left( \frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \frac{\partial T}{\partial x_j}, \quad (5.6)$$

where  $T$  is temperature,  $C_p$  is the specific heat,  $Pr$  is the Prandtl number,  $Pr_T$  is the turbulent Prandtl number. In addition, Sutherland’s Law is utilized to relate dynamic viscosity and temperature, and the ideal gas equation of state is used.

The turbulent source term for the SST model consists of a production term,  $\mathcal{P}$ , and a destruction term for each working variable,  $\mathcal{D}_k$ , and  $\mathcal{D}_\omega$ , respectively. In this work we apply a modification suggested by Menter [25] when working with aerodynamics flows. The production term is defined as

$$\mathcal{P} = \mu_t \Omega - \frac{2}{3} \rho k \delta_{ij} \frac{\partial u_i}{\partial x_j}. \quad (5.7)$$

The details of the standard SST model, including the well-known definitions of the production and destruction terms, as well as the original closure coefficients which are used in this work, may be found in the original work by Menter [24].

In this work, we follow the high-order strand grid discretization scheme of Katz and Work [11]. In this method, strand grids consisting of an unstructured triangular surface tessellation are extruded along straight lines (strands) away from solid bodies. Each stack of prismatic cells emanating from the surface in the physical space may be mapped to a standard computational space as shown in Figure 5.2. The triangular base of each prismatic element may be divided into equally spaced sub-triangles in the  $r$ - $s$  plane, facilitating high-order solution and flux representation. In this work, we investigate linear, quadratic, and cubic surface elements. Quadratic surface elements are shown in Figure 5.2. The stretched distribution of nodes along each strand in the physical space is mapped to an equally spaced distribution in the  $\eta$ -direction in the computational space, where  $\eta \in (0, 1)$ . In the computational space, the strand spacing is  $\Delta\eta = 1/(N - 1)$ , where  $j = 1, \dots, N$  is the strand node numbering beginning with the node on the surface. To avoid confusion, the triangular-shaped elements forming the various levels in the prism stack will be referred to as “surface elements,” while the three-dimensional elements formed by an equally-spaced stacking the surface elements in the  $\eta$ -direction will be referred to as “volume elements.” Additionally, the triangles formed from sub-dividing each surface element will be referred



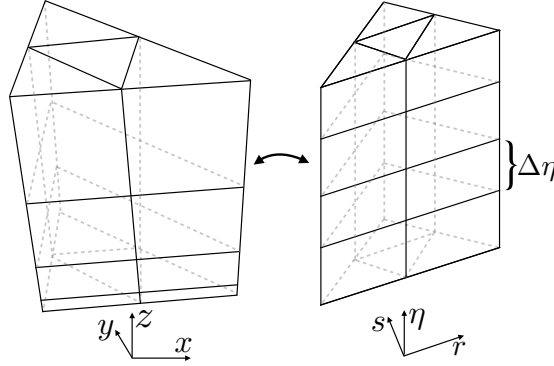


Fig. 5.2: Mapping of strand stack from physical space to computational space.

to as “sub-triangles.” While not shown in Figure 5.3, quartic surface elements are used for all applications in this work. It is known that in order for high-order schemes to deliver high orders of accuracy, high-order curved boundary elements are essential [30,31].

Upon transformation to the computational space, Equation 5.1 becomes

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} + \frac{\partial \hat{H}}{\partial \eta} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} - \frac{\partial \hat{H}^v}{\partial \eta} = \hat{S}, \quad (5.8)$$

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

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

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

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

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

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

Here,  $J$  is the Jacobian of the transformation,  $\hat{F}_j$  and  $\hat{F}_j^v$  are the transformed inviscid and viscous fluxes, and partial differentiation is denoted with a subscript (e.g.  $\partial x / \partial s = x_s$ ).

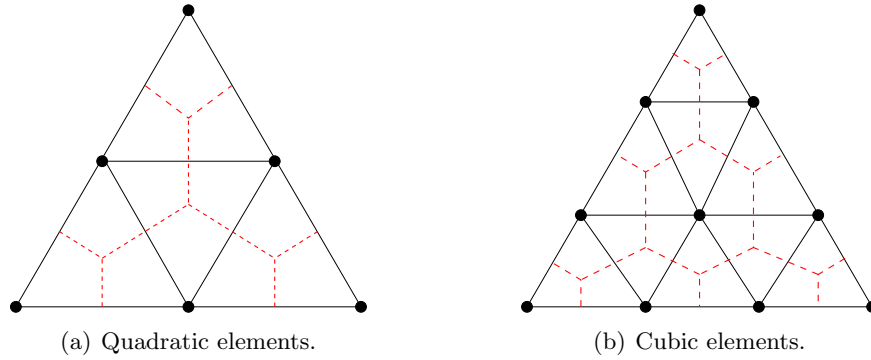


Fig. 5.3: Element mappings used for gradient reconstruction.

A unique aspect of this scheme is the novel manner in which the discretization is carried out in the  $r$ - $s$  plane and the  $\eta$  direction to obtain high-order accuracy. The general strategy is to perform a high-order finite-volume flux balance in the  $r$ - $s$  plane, known as “flux correction,” and to use high-order finite-differences based on SBP operators in the  $\eta$ -direction. To facilitate the flux correction algorithm, two-dimensional median-dual control volumes are constructed around each node in each high-order surface element in the  $r$ - $s$  plane. The sub-triangles and median-dual control volumes in a single surface element are shown as the black solid lines and red dashed lines in Figure 5.3 for quadratic and cubic surface elements, respectively.

In previous work [11] it was shown that treating the  $\eta$ -derivatives in Equation 5.8 with a particular source term discretization preserves the accuracy of the flux correction procedure in the  $r$ - $s$  plane. Therefore, the  $\eta$ -derivatives and physical time derivative are moved to the right-hand side and treated as source terms:

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial r} + \frac{\partial \hat{G}}{\partial s} - \frac{\partial \hat{F}^v}{\partial r} - \frac{\partial \hat{G}^v}{\partial s} = \tilde{S}, \quad (5.9)$$

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

A pseudo-time derivative is added on the left-hand side of Equation 5.9 to facilitate the time-marching solution procedure is described in previous work [11]. As long as each term

in  $\tilde{S}$  is computed to at least second-order accuracy in terms of the truncation error, the corrected flux balance in the  $r - s$  plane will retain desirable truncation error properties, resulting in a high-order accurate scheme. Examining equation 5.9, the three-dimensional equations essentially reduce to a two-dimensional problem in the  $r$ - $s$  plane at each layer in the strand grid. The layers are coupled together through the new source term,  $\tilde{S}$ , which contains  $\eta$ -derivatives of the flux.

Equation 5.9 requires a two-dimensional discretization in the  $r$ - $s$  plane suitable for unstructured grids with a source term. The discretization used for this purpose here is the flux-correction scheme of Katz and Sankaran [9]. Recently, Pincock and Katz [10] extended the original scheme to include the viscous terms in the Navier-Stokes equations. A defining feature of the flux-correction scheme is its ability to retain high-order accurate truncation error on general simplex grids, which results in third-order discretization error for inviscid fluxes, and fourth-order discretization error for viscous fluxes.

The flux-correction procedure differs from a conventional finite volume scheme in the definition of the numerical fluxes. Following previous work on flux-correction [9], the inviscid numerical fluxes between nodes  $a$  and  $b$  are computed as

$$\hat{\mathcal{F}}_{ab,j} = \frac{1}{2} \left( \hat{\mathcal{F}}_R + \hat{\mathcal{F}}_L \right) - \frac{1}{2} \left| \hat{A}(Q_R, Q_L) \right| (Q_R - Q_L). \quad (5.10)$$

The unique aspect of the flux-correction scheme is the use of reconstructed fluxes, along with high-order gradient information. The left and right reconstructed fluxes are computed as

$$\hat{\mathcal{F}}_L = \hat{\mathcal{F}}_{a,j} + \frac{1}{2} \Delta \mathbf{r}_{ab}^T \left( \nabla_{rs}^h \hat{\mathcal{F}} \right)_{a,j}, \quad \hat{\mathcal{F}}_R = \hat{\mathcal{F}}_{b,j} - \frac{1}{2} \Delta \mathbf{r}_{ab}^T \left( \nabla_{rs}^h \hat{\mathcal{F}} \right)_{b,j}, \quad (5.11)$$

where  $\Delta \mathbf{r}_{ab}^T = (r_b - r_a, s_b - s_a)$ , and  $\nabla_{rs}^h$  is an estimate of the gradient in the  $r$ - $s$  plane computed to at least second-order accuracy, such that

$$\nabla_{rs}^h = \nabla_{rs} + O(h^q), \quad q \geq 2.$$

The strategy to compute such gradients is to compute the derivatives of a local polynomial

representation within each surface element, and then volume-average among the shared representations in neighboring elements. The averaging is critical to centering the gradients and to the overall stability of the scheme. The requirement that  $q \geq 2$  may be satisfied by using at least quadratic surface elements shown in Figure 5.3. Further details of the gradient procedure used for the flux and solution may be found in our previous work [11]. There, other details regarding the SBP/SAT-treatment [15, 16, 32–38] of the  $\eta$ -flux derivatives are also provided.

#### 5.4 Turbulence Model Implementation

Implementation of the Spalart-Allmaras turbulence model in high-order schemes is both straight forward and common place as the single additional equation requires little modification to the system. Previous work with flux correction has shown this to be an effective model at high-orders of accuracy [13, 14]. Source term treatment is simple, and an exact source Jacobian is utilized in order to aid convergence, as suggested by Merkle [39]. Conversely, the Menter SST model is not commonly implemented in high-order schemes due to its complexity, and oscillatory and convergence issues. In this section, we outline efforts made to implement the model into the high-order flux correction scheme.

##### 5.4.1 Oscillation Damping Treatment

The primary reason for not implementing SST into high-order methods stems from the oscillations that occur due to the infinite wall boundary condition required by specific dissipation rate,  $\omega$ . While this is approximated with a large  $\omega$  value at the wall, high-order schemes struggle with this condition and suffer from oscillations and convergence issues unless damping is applied. Bassi [26] implemented high-order SST into a high-order discontinuous Galerkin (DG) scheme, and overcame this issue by rewriting the  $\omega$  equation. Similar efforts were also made by Balakumar [27] for an essentially non-oscillatory (ENO) scheme.

In this work, we keep the original equation, however, various limiters are applied to damp the oscillatory effects of the boundary conditions and to improve convergence. First,

in order to prevent the negative non-physical turbulence working variables we apply an update limiter, as outlined by Park and Kwon [40]. After each implicit sweep,  $\Delta(\rho k)$  and  $\Delta(\rho\omega)$  are limited to half of their original value if updated value becomes negative.

$$(\rho\omega)_{new} = \begin{cases} (\rho\omega)_{old} - \Delta(\rho\omega) & \text{if } \Delta(\rho\omega) < (\rho\omega)_{old} \\ \frac{1}{2}(\rho\omega)_{old} & \text{if } \Delta(\rho\omega) \geq (\rho\omega)_{old} \end{cases} \quad (5.12)$$

Second, we impose a lower limit to  $\omega$  if it becomes unphysically low, resulting in excessive generation of turbulent stresses and higher values of  $\mu_t$ . Derivation of the limiter is given by Zheng and Liu [41], with the final result giving

$$\omega = \begin{cases} \omega & \text{if } \omega > \alpha S \\ \alpha S & \text{if } \omega \leq \alpha S. \end{cases} \quad (5.13)$$

Here,  $\alpha = \frac{5}{9}$  and  $S$  is the mean strain rate, defined as

$$S = \sqrt{2S_{ij}S_{ij} - \frac{2}{3}S_{kk}^2}, \quad (5.14)$$

where,  $S_{ij}$  is the strain rate tensor, as defined previously.

Third, in order to improve convergence, the implicit source term treatment described by Merci [42] is implemented. Here, similar to SA, an exact Jacobian of the source terms is constructed. In order to retain diagonal dominance, negative terms are treated implicitly, while positive terms are treated explicitly.

Finally, we introduce simple limiting techniques to our flux-correction high-order scheme to locally introduce first-order dissipation at discontinuities. While such a technique is generally used for shocked flows, we may implement these techniques to remove oscillations of  $k$  and  $\omega$  in the boundary layer where discontinuities occur due to the model switch. Just as in finite volume schemes, the form of artificial dissipation in Equation 5.10 includes a difference of left and right states,  $Q_L$  and  $Q_R$ . For smooth flows, we reconstruct these states using gradients with second order accuracy or better. To handle discontinuous flows, we

multiply these gradients by a limiter, which locally switches the left and right states to the nodal values, resulting in a first-order dissipation term. The reconstructed states with limiter,  $\phi$ , may be expressed as,

$$Q_L = Q_{a,j} + \frac{\phi_{ab}}{2} \Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{a,j}, \quad Q_R = Q_{b,j} - \frac{\phi_{ab}}{2} \Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{b,j}. \quad (5.15)$$

Here,  $\phi_{ab}$  is the limiter at the edge in the  $r$ - $s$  plane connecting nodes  $a$  and  $b$ . Similar to the symmetric limited positive (SLIP) scheme of Jameson [28, 29], we compute the limiter as,

$$\phi_{ab} = 1 - \left| \frac{u - v}{\max(|u| + |v|, \epsilon)} \right|^3, \quad u = 2\Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{b,j} - \Delta Q_{ab}, \quad v = 2\Delta \mathbf{r}_{ab}^T \nabla_{rs}^h Q_{a,j} - \Delta Q_{ab}, \quad (5.16)$$

where  $\epsilon$  is a small number to avoid limiting at smooth extrema or dividing by zero in freestream flow, and  $\Delta Q_{ab} = Q_b - Q_a$  is the solution difference across the edge. The  $r$ - $s$  gradients in Equation 5.16 are calculated using the same method used to obtain the other flux and solution gradients in the FC scheme. Therefore, no additional MPI communication is required, and little extra computational effort or complexity is added.

#### 5.4.2 Inviscid and Dissipation Flux Treatment

Treatment of the viscous and source terms for the RANS-SST system is relatively simple, however, inviscid and dissipation terms require some additional modification. The RANS stress tensor, defined in Equation 5.4, requires calculation of the term,  $2/3\rho k$ , referred to as the turbulent kinetic pressure (TKP). The flux correction algorithm permits only terms with first derivatives in viscous flux. Thus, for convenience, we place the TKP inside the inviscid fluxes. Furthermore, the TKP behaves similarly to the thermodynamic pressure, in that it contributes solely to the isotropic or mean stress component of the stress tensor, as opposed to the viscous stresses which are purely deviatoric. The inviscid flux Jacobian for RANS is then defined as

$$A = \begin{pmatrix} 0 & n_x & n_y & n_z & 0 & 0 & 0 \\ \gamma_1 \frac{U^2}{2} n_x - uu_n & un_x - \gamma_1 un_x + u_n & un_y - \gamma_1 vn_x & un_z - \gamma_1 wn_x & \gamma_1 n_x & 0 & 0 \\ \gamma_1 \frac{U^2}{2} n_y - vu_n & vn_x - \gamma_1 un_y & vn_y - \gamma_1 vn_y + u_n & vn_z - \gamma_1 wn_y & \gamma_1 n_y & 0 & 0 \\ \gamma_1 \frac{U^2}{2} n_z - wu_n & wn_x - \gamma_1 un_z & wn_y - \gamma_1 vn_z & wn_z - \gamma_1 wn_z + u_n & \gamma_1 n_z & 0 & 0 \\ \left(\gamma_1 \frac{U^2}{2} - h\right) u_n & hn_x - \gamma_1 uu_n & hn_y - \gamma_1 vu_n & hn_z - \gamma_1 wu_n & \gamma u_n & 0 & 0 \\ -ku_n & kn_x & kn_y & kn_z & 0 & u_n & 0 \\ -\omega u_n & \omega n_x & \omega n_y & \omega n_z & 0 & 0 & u_n \end{pmatrix}, \quad (5.17)$$

where  $U$  is the velocity magnitude,  $\gamma$  is the specific heat ratio and  $\gamma_1 = \gamma - 1$ . Modification of the speed of sound is also required. The speed of sound is based on the total pressure, and is thus related to the thermodynamic properties as well as to the turbulent kinetic energy. Thus, it can now be defined as

$$c^2 = \frac{p + \frac{2}{3}\rho k}{\rho}. \quad (5.18)$$

Next, we perform an eigenvalue decomposition on the inviscid fluxes, required for computing dissipation fluxes, given as

$$A = X\Lambda X^{-1} \quad (5.19)$$

This decomposition is used computing the dissipation fluxes. The eigenvalue matrix is changed to include the absolute value of all diagonal terms. The matrices are multiplied together to yield a modified inviscid flux Jacobian which is used to calculate the dissipation fluxes which eliminate spurious oscillations. The decomposition is also employed in penalty-based boundary conditions, used in the flux correction scheme. The eigenvalue matrix is altered to include only selective diagonal terms and some off-diagonal terms. Terms are selected such that the resulting inviscid flux Jacobian enforces characteristic relations such as a specified pressure. Selections are also designed to maintain well-posedness [36].

Here, we set our eigenvalues as  $\Lambda = \text{diag}(u_n, u_n, u_n, u_n + c, u_n - c, u_n, u_n)$ , requiring calculation of the right and left eigenvectors, given as

$$X = \begin{pmatrix} n_x & n_y & n_z & 1 & 1 & 0 & 0 \\ un_x & un_y - cn_z & un_z + cn_y & u + cn_x & u - cn_x & 0 & 0 \\ vn_x + cn_z & vn_y & vn_z - cn_x & v + cn_y & v - cn_y & 0 & 0 \\ wn_x - cn_y & wn_y + cn_x & wn_z & w + cn_z & w - cn_z & 0 & 0 \\ X_{51} & X_{52} & X_{53} & h + u_n c & h - u_n c & -\left(\frac{5}{3} - \gamma\right) & 0 \\ 0 & 0 & 0 & k & k & \gamma_1 & 0 \\ 0 & 0 & 0 & \omega & \omega & 0 & 1 \end{pmatrix} \quad (5.20)$$

where,

$$X_{51} = \frac{U^2}{2} n_x + c(vn_z - wn_y), \quad X_{52} = \frac{U^2}{2} n_y + c(-un_z + wn_x), \quad X_{53} = \frac{U^2}{2} n_z + c(un_y - vn_x), \quad (5.21)$$

and

$$X^{-1} = \begin{pmatrix} n_x(1 - b_3) - \frac{vn_z - wn_y}{c} & b_2 un_x & b_2 vn_x + \frac{n_x}{c} & b_2 wn_x - \frac{n_y}{c} & -b_2 n_x & -b_4 n_x & 0 \\ (1 - b_3) n_y - \frac{un_z - wn_x}{c} & b_2 un_y - \frac{n_x}{c} & b_2 vn_y & b_2 wn_y + \frac{n_x}{c} & -b_2 n_y & -b_4 n_y & 0 \\ (1 - b_3) n_z - \frac{un_y - vn_x}{c} & b_2 un_z + \frac{n_y}{c} & b_2 vn_z - \frac{n_x}{c} & b_2 wn_z & -b_2 n_z & -b_4 n_z & 0 \\ \frac{1}{2}(b_3 - u_n c) & \frac{1}{2}(b_2 u - \frac{n_x}{c}) & \frac{1}{2}(b_2 v - \frac{n_y}{c}) & \frac{1}{2}(b_2 w - \frac{n_x}{c}) & \frac{1}{2} b_2 & b_4 & 0 \\ \frac{1}{2}(b_3 + u_n c) & \frac{1}{2}(b_2 u + \frac{n_x}{c}) & \frac{1}{2}(b_2 v + \frac{n_y}{c}) & \frac{1}{2}(b_2 w + \frac{n_x}{c}) & \frac{1}{2} b_2 & b_4 & 0 \\ -M_t^2 U^2 & M_t^2 u & M_t^2 v & M_t^2 w & -M_t^2 & \frac{h - U^2}{c^2} & 0 \\ -\omega b_3 & \omega b_2 u & \omega b_2 v & \omega b_2 w & -\omega b_2 & -\omega b_4 & 1 \end{pmatrix} \quad (5.22)$$

where,

$$M = \frac{U}{c}, \quad M_t^2 = \frac{k}{c^2}, \quad b_2 = \frac{\gamma - 1}{c^2}, \quad b_3 = b_2 U^2, \quad \text{and} \quad b_4 = \frac{\frac{5}{3} - \gamma}{c^2}. \quad (5.23)$$

Some CFD algorithms and turbulence models, such as Spalart-Allmaras, neglect the TKP on the basis that the relatively small size of  $k$  makes its contribution to the stress tensor



negligible. While it is true that  $k$  is typically small in comparison to the thermodynamic pressure, the magnitude of the gradient of a variable is more impactful than the magnitude of the variable itself [43]. Thus in mixing regions, where the gradient of  $k$  is relatively large, the TKP can have substantial effects on the solution. The TKP is of particular importance for supersonic flow [23]. Further discussion with respect to the flux correction scheme is provided by Katz and Work [11].

## 5.5 Results

The ability of the flux correction method to accurately compute three-dimensional high Reynolds number turbulent flow within the strand grid paradigm is investigated with the Menter SST turbulence model. In this section, we present SST results, and compare these against results from established solvers, provided by the NASA-Langley Turbulence Modeling Resource (TMR) website [23]. First, we conduct rigorous accuracy tests through the method of manufactured solutions. Next, we perform a study of three-dimensional flow over a zero pressure gradient flat plate. A study of turbulent three-dimensional flow over a bump-in-channel is investigated.

### 5.5.1 Method of Manufactured Solutions

First, we rigorously test the accuracy of the high-order strand algorithm by performing grid refinement studies using manufactured solutions (MMS) [44, 45]. The use of MMS to assess accuracy of discretizations for compressible viscous flows is important due to the lack of exact solutions. Smooth trigonometric functions for x-velocity are shown in Figure 5.4(a) for a cube geometry. Similar smooth trigonometric variables are used for each of the working variables. When assessing high-Reynolds number flows, the working variables of the Menter SST model  $k$  and  $\omega$  are treated in a similar manner. Four levels of grid refinement are used, and the error between the discrete and exact manufactured solution is measured on each mesh after converging to machine precision.

The results of the MMS study are shown in Figure 5.4, where SLIP limiting is not employed over the cube grid. Here, we show the inviscid and viscous terms in isolation as

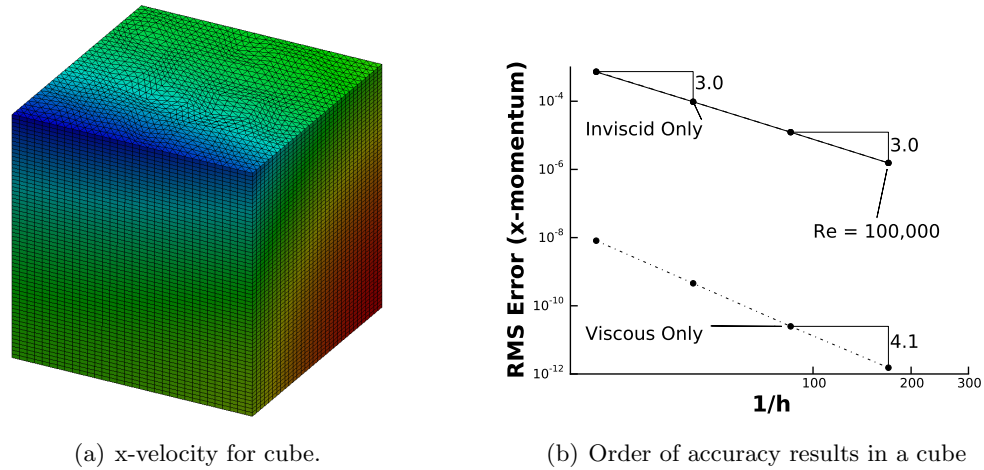
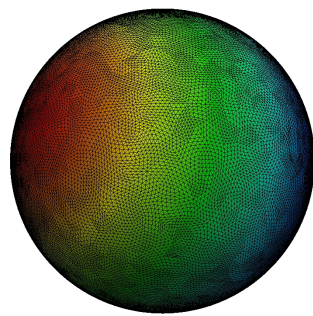
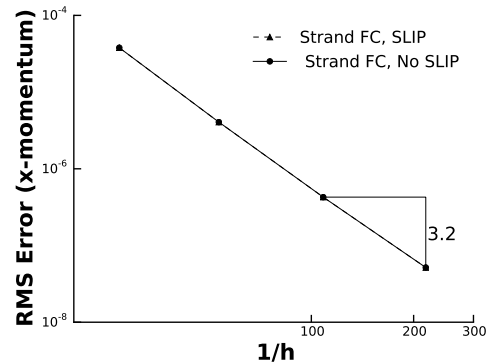


Fig. 5.4: Manufactured solution and order of accuracy results in a cube geometry.

well as combined together. The SST closure source terms are turned off when examining inviscid and viscous terms in isolation, and turned on when all the terms are combined. One advantage of verification via manufactured solutions is the ability to isolate various terms (e.g. inviscid or viscous), to assess the accuracy of each. As the figures show, the inviscid terms asymptote sharply to third-order accuracy. The inviscid curve and combined curve fall directly on top of one another. The viscous terms are significantly more accurate, generating accuracy slightly better than fourth-order. Most importantly, the combined inviscid and viscous discretizations are compatible, yielding an order of accuracy of 3 for SST. These results are valid for irregular surface meshes and smooth strand distributions with potentially very high aspect ratios. We demonstrate this with an additional MMS verification study over a sphere, shown in Figure 5.5(b), where we observe the flux correction method maintains its high-order of accuracy on an irregular sphere grid (Figure 5.5(a)), with and without the use of SLIP limiting techniques. The figure shows that the use of the SLIP limiter does not corrupt the smooth solution accuracy. Furthermore an high-order of accuracy was observed with both the limiter being used and without it. It should be noted that conventional second order schemes are easily recovered by turning off the flux correction terms and lowering the difference order along strands, which is easily accomplished with an input flag.



(a) x-velocity for sphere.



(b) Order of accuracy results for a sphere.

Fig. 5.5: Manufactured solution and order of accuracy results in a sphere geometry.

We wish to emphasize the fact that the flux correction solver is able to maintain high-order of accuracy with the of the Menter SST turbulence model (e.g.  $k$  and  $\omega$  RMS error follows a trend similar the to x-momentum error). The turbulence model within the flux correction solver is fully coupled with the RANS equations, and thus exhibits greater accuracy than many traditional RANS solvers with segregated turbulent models, or models that reduce the turbulence equation to first-order.

### 5.5.2 Zero Pressure Gradient Flat Plate

With the implementation of turbulence models verified through MMS, validation of the solver is now performed. The first case we examine is a zero pressure gradient flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ , based on a plate of length unity. Grids from the NASA-Langley turbulence modeling resource are used. However, sub-elements are generated within each

Table 5.1: Error for inviscid, viscous and source terms combined ( $Re = 100,000$ ) for a cube geometry (x-momentum).

Mesh Nodes	Error	Order
2,048	1.776615827e-04	-
16,384	1.864328530e-05	2.947
131,072	2.191720008e-06	2.976
1,048,576	2.524722826e-07	3.008

parent fourth-order element, thus giving more nodes along the surface than the original grids provide. In this work, we examine a number of different grids, starting with the coarsest grid, up to the middle grid provided. These grids are  $35 \times 25$ ,  $69 \times 49$ , and  $137 \times 97$ , where the first number corresponds to the number of points in the surface direction, and the second in the strand direction. As the flux correction scheme performs better with high-order grids, increase the number of nodes in the strand direction to match the new resolution that fourth-order grids provide. The number of nodes in the strand direction used are: 97, 192, and 384, and we denote these in the text and figures as “coarse,” “medium,” and “fine,” respectively. An example grid is shown in Figure 5.6(a). The plate leading edge begins at  $x = 0$  and extends for a length of 2. A short inviscid wall entry way beginning at  $x = -0.33$  is provided to allow for proper inflow conditions. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow. Details of the exact boundary conditions and case set up may be found on NASA-Langley turbulence resource website [23].

First, we examine the normalized x-velocity profiles in Figure 5.7 at two different locations,  $x = 0.97003$  and  $x = 1.9003$ , for increasing grid resolution. We see the high-order scheme quickly asymptotes to the results from the established solvers FUN3D and CFL3D. The turbulent viscosity field for this case is shown in Figure 5.6(b), which has been scaled by a factor of 40 vertically to facilitate visualization.

We examine the normalized  $k$  and  $\omega$  profiles in Figure 5.8 at two different locations,

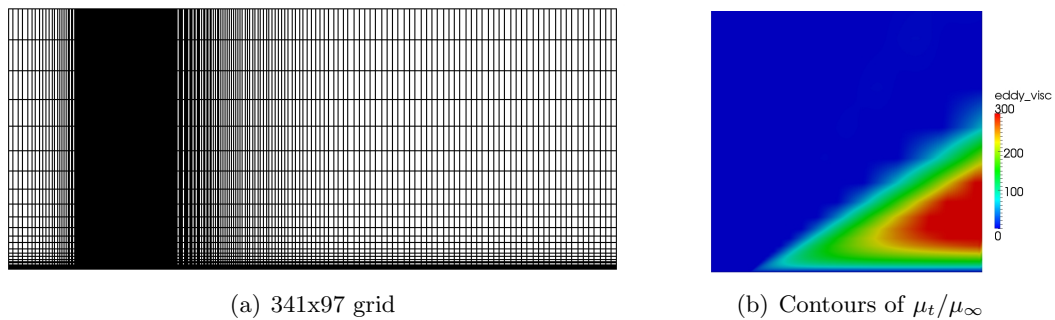
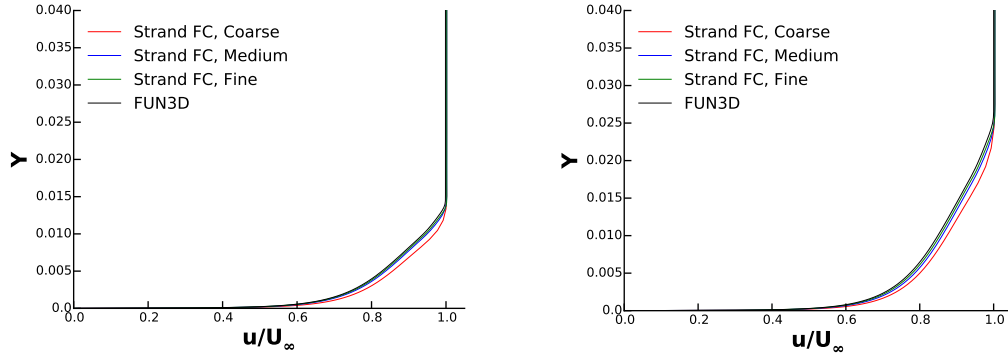


Fig. 5.6: Grid and normalized turbulent eddy-viscosity contours for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .



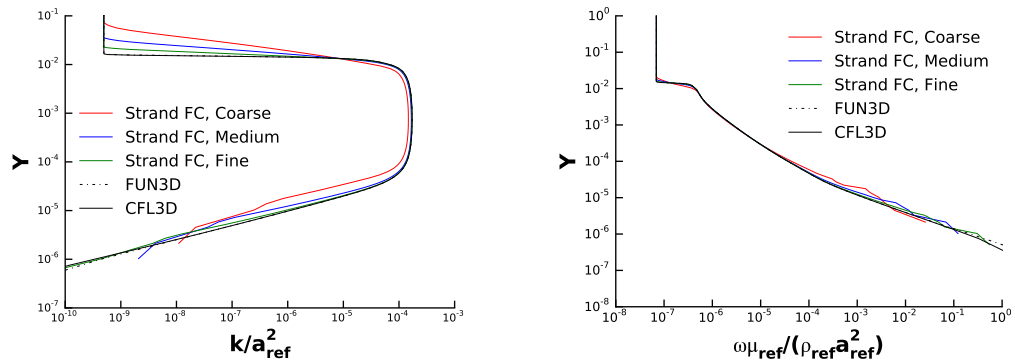
(a) Normalized x-velocity profiles at  $x = 0.97003$  (b) Normalized x-velocity profiles at  $x = 1.9003$

Fig. 5.7: Comparison of normalized stream-wise velocity profiles for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .

$x = 0.97003$  and  $x = 1.9003$ , for increasing grid resolution. Importantly, all grid resolutions show no oscillatory behavior for the high-order  $k$  and  $\omega$  profiles. Some minor discrepancies in the curves can be seen at coarser grids, however, high-order scheme asymptotes to the results from the established solvers FUN3D and CFL3D.

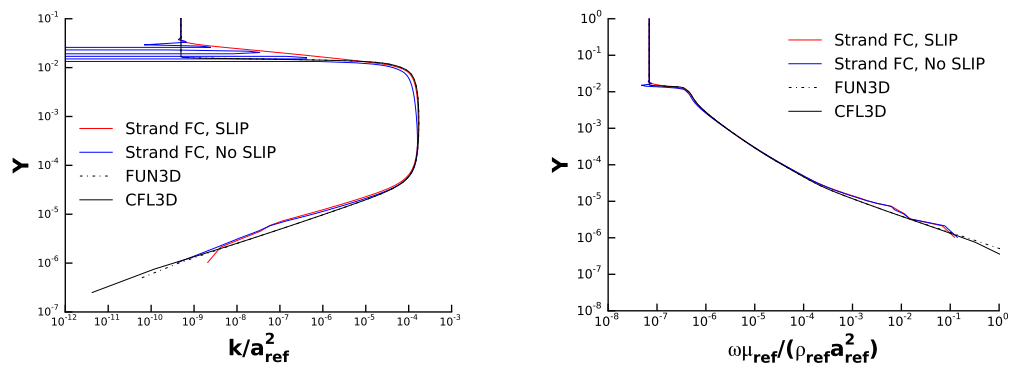
Figure 5.9 shows the effect of applying the SLIP limiting technique to the Menter SST model in a high-order scheme, and is compared the results from the established solvers, CFL3D and FUN3D. As expected, the flux correction scheme without any limiting techniques shows severe oscillations at the discontinuities in both the  $k$  and  $\omega$  profiles. Critically, the flux correction scheme with the SLIP limiting technique shows no oscillations at the discontinuity in the non-dimensional profile. It should be noted, that limiting is only applied to the  $k$  and  $\omega$  equations when needed, thus, other conserved variables such as velocity maintains its high-order of accuracy in areas where limiting is required.

The computed drag coefficient, which is entirely due to skin friction for this case, is shown in Table 5.2 for the finest grid, along with FUN3D and CFL3D results for the finest grid available on the TMR. The drag coefficient shows excellent agreement to the established codes.



(a) Normalized turbulent kinetic energy profile at  $x = 0.97003$  (b) Normalized specific dissipation rate profile at  $x = 0.97003$

Fig. 5.8: Comparison of normalized turbulent kinetic energy and normalized specific dissipation rate profiles for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .



(a) Normalized turbulent kinetic energy profile at  $x = 0.97003$  with and without SLIP limiter (b) Normalized specific dissipation rate profile at  $x = 0.97003$  with and without SLIP limiter

Fig. 5.9: Comparison of normalized turbulent kinetic energy and normalized specific dissipation rate profiles with and without a SLIP limiter for medium grid refinement over a flat plate.

### 5.5.3 Bump in Channel

Further validation is performed for turbulent flux correction by way of a bump-in-channel case. This case is conducted at a Mach number of  $M = 0.2$ , at a Reynolds number of  $Re = 3 \times 10^6$  based on a grid length of unity. The body reference length is 1.5 units, where the lower wall is a viscous-wall bump extending from  $x = 0$  to 1.5. The maximum bump height is  $y = 0.05$ . The upstream and downstream farfield extends 25 units from the viscous-wall, with symmetry boundary conditions imposed on the lower wall between the farfield and the solid wall. The upper boundary at  $y = 5.0$  is set to an inviscid plane. A further description and layout of the case may be found on the NASA-Langley turbulence website [23].

Fourth-order elements are used to create a grid with  $705 \times 97$  nodes, shown in Figure 5.10(a), and close up in Figure 5.10(b). Figures 5.10(b) and 5.10(c) are scaled by a factor of 15 vertically to facilitate visualization. To avoid internal corner issues over the bump, strands are fixed vertically to avoid crossings. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow. The turbulent viscosity field for this case is shown in Figure 5.10(c).

Stream-wise velocity profiles are shown for two locations downstream on the bump, and are over-plotted with FUN3D and CFL3D results in Figure 5.11. Good agreement is obtained for all profiles, even for the  $705 \times 96$  grid, which is approximately 7 times more coarse than the FUN3D and CFL3D results shown in the figure. Minor oscillatory behavior is observed in the specific dissipation rate curve at edge of the boundary layer in the region where the model switch occurs, however, good agreement is seen otherwise. In Figure 5.13, a plot of the surface coefficient of pressure and friction along the bump is shown, and is over

Table 5.2: Comparison of computed drag coefficients for flow over a flat plate at  $M = 0.2$  and  $Re = 5 \times 10^6$ .

Solver	$C_d$
Strand FC	2.858552E-3
FUN3D	2.844174E-3
CFL3D	2.85332E-3

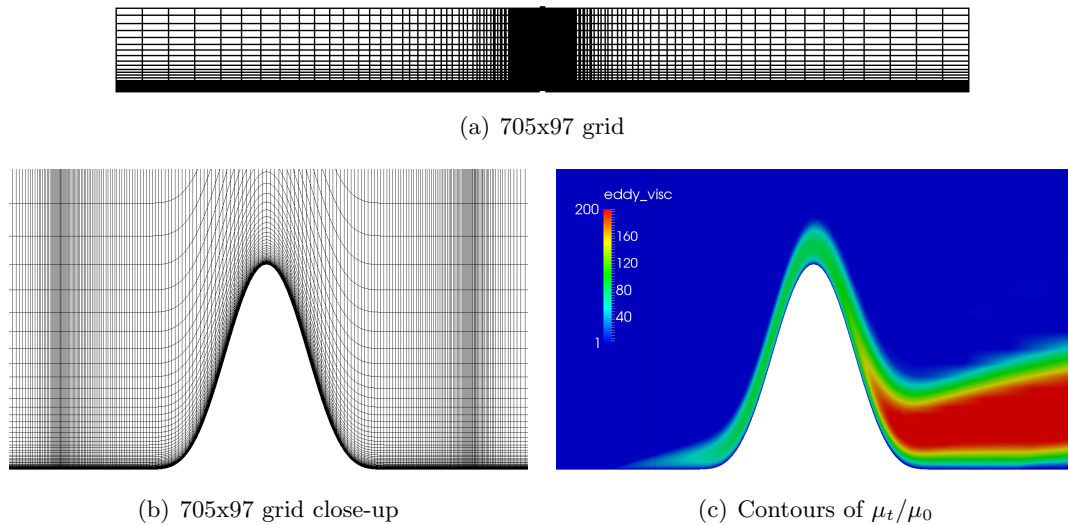


Fig. 5.10: Grid and turbulent viscosity contours for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

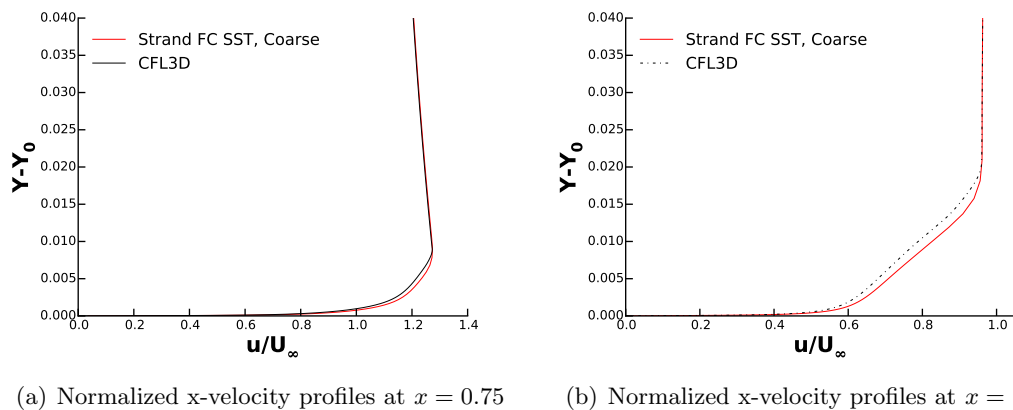
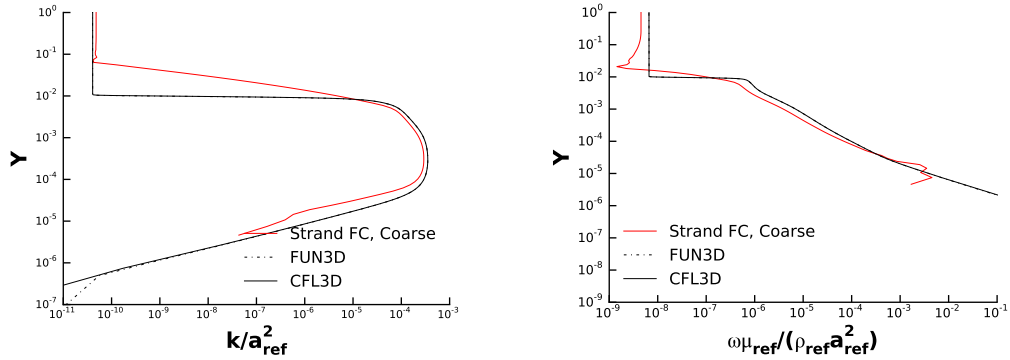


Fig. 5.11: Comparison of normalized stream-wise velocity profiles for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .





(a) Normalized turbulent kinetic energy at  $x = 0.75$ . (b) Normalized specific dissipation rate profile at  $x = 0.75$ .

Fig. 5.12: Comparison of normalized turbulent kinetic energy and specific dissipation rate profiles for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

plotted with CFL3D, FUN3D and SA results. The computed drag coefficient, is shown in Table 5.3. Drag prediction shows a minor discrepancy compared to the established codes. This is reflected in the minor discrepancies observed in the coefficient of friction curve in Figure 5.13(b). However, relatively good agreement is seen, despite containing skewed high-aspect ratio cells over the critical areas of the bump.

## 5.6 Conclusions and Future Work

An assessment of a two-equation Menter SST turbulence model in the high-order flux correction method on strand grids has been presented for high-Reynolds number viscous flows. The high-order scheme requires roughly the same computational cost as standard second-order finite volume schemes. The method combines flux correction in the unstructured plane of the strand grids and summation-by-parts finite differences to compute flux derivatives in the strand direction. The flux correction method works by canceling low-order

Table 5.3: Comparison of computed drag coefficients for flow through a bump-in-channel at  $M = 0.2$  and  $Re = 3 \times 10^6$ .

Solver	SST $C_d$
Strand FC	3.45830E-3
FUN3D	3.58605E-3
CFL3D	3.59931E-3

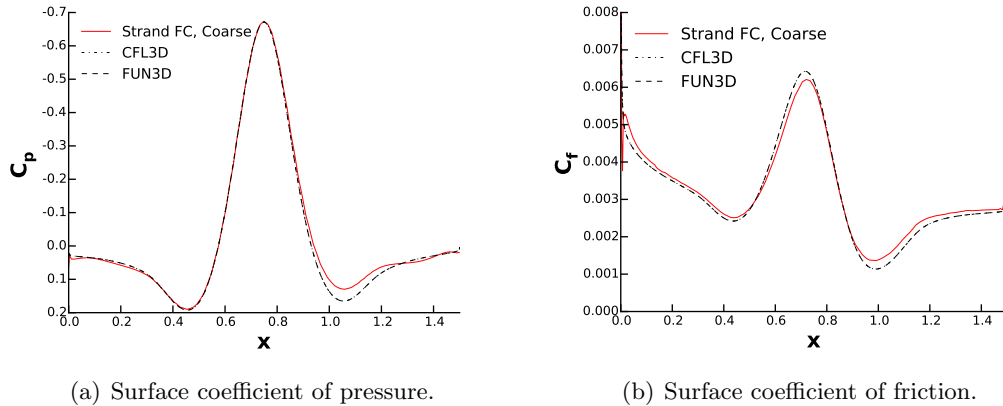


Fig. 5.13: Bump-in-channel surface coefficient of pressure and friction.

truncation errors with accurate flux and solution gradients. By treating the high-order flux derivatives in the strand direction as a source term, we were able to retain the error cancellation of the flux correction method. Limiters implemented to damp specific dissipation rate oscillations as well as treatment of inviscid and dissipation terms are briefly discussed.

Detailed verification studies with the method of manufactured solutions showed that the Menter SST model in the flux correction achieved at least third order accuracy for irregular and regular surface meshes with uniform and hyperbolic strand spacing. When applied to boundary layer flows, the high-order strand scheme produces accurate velocity profiles, even on coarse meshes with significant stretching. When applied to a three-dimensional bump in a channel, the flux-correction method accurately predicted the drag for turbulent subsonic flow. Minimal oscillations were seen in the specific dissipation rate for both cases.

While this work appears promising in terms of accurately computing complex turbulent flows, the test cases presented here are rather simple geometrically. Future work will focus on extending our high-order methods to geometries of practical interest. Two research areas are envisioned in this effort. First, we intend to couple the strand grids to off-body Cartesian grids in an overset fashion. While overset grid methods are certainly not new, specific challenges may arise in terms of stability and accuracy with our high-order schemes. The effect of high-order interpolation and energy stable interface conditions will be investigated. Second, we intend to investigate novel ways of representing and handling

complex surface geometry. The geometry surface serves as the starting point for strand growth and off-body mesh generation. Therefore, the ability to handle any geometry with complex combinations of convex, concave, and saddle regions, is of prime interest. One possible approach would be to consider surface geometry as the zero level set of a higher-dimensional function. Along with variable strand lengths, this could facilitate small surface modifications and adaptations that would allow for the robust treatment of any geometry with the strand method.

### **Acknowledgments**

Development was performed with the support of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office, by the Army Research Office Fluid Dynamics Program directed by Dr. Frederick Ferguson, and by the Office of Naval Research Sea-Based Aviation program directed by Dr. Judah Milgram and Mr. John Kinzer.

**REFERENCES**

- [1] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [2] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [3] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA paper* 2009-3792, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [4] Katz, A., Wissink, A., Sitaraman, J., and Sankaran, V., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *AIAA paper* 2010-4934, 28th AIAA Applied Aerodynamics Conference, Chicago, June 2010.
- [5] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.
- [6] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [7] Lee, Y.-L. and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [8] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.

- [9] 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.
- [10] 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.
- [11] Katz, A. and Work, D., “High-Order Flux Correction/Finite Difference Schemes for Strand Grids,” *AIAA paper*, AIAA 52nd Aerospace Sciences Meeting, National Harbor, MD, January 2014.
- [12] Tong, O., Work, D., and Katz, A., “High-Order Methods for Turbulence Using Strand Grids,” Tech. Rep. ICCFD8-0215, 8th International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 2014.
- [13] Tong, O., Katz, A., Yanagita, Y., Casey, A., and Schaap, R., “High-Order Methods for Three-Dimensional Strand Grids,” *Journal of Scientific Computing*, Vol. 64, 2015.
- [14] 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 CFD Conference, Dallas, TX, June 2015.
- [15] 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.
- [16] Fernandez, D. C. D. R. and Zingg, D., “High-Order Compact-Stencil Summation-By-Parts Operators for the Second Derivative with Variable Coefficients,” Tech. Rep. ICCFD7-2803, 7th International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, HI, July 2012.

- [17] Barth, T. J. and Frederickson, P., “Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction,” *AIAA paper* 1990-0013, AIAA 28th Aerospace Sciences Meeting, Reno, NV, January 1990.
- [18] Delanaye, M. and Liu, Y., “Quadratic Reconstruction Finite Volume Schemes on 3D Arbitrary Unstructured Polyhedral Grids,” *AIAA paper* 1995-3259, AIAA 14th CFD Conference, Norfolk, June 1999.
- [19] Ollivier-Gooch, C., Nejat, A., and Michalak, K., “On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes,” *AIAA paper* 2007-4464, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [20] Wissink, A., “Helios Solver Developments Including Strand Meshes,” Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.
- [21] Spalart, P. and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *Recherche Aerospatiale*, Vol. 1, 1994, pp. 5–21.
- [22] Allmaras, S., Johnson, F., and Spalart, P., “Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model,” Tech. Rep. ICCFD7-1902, 7th International Conference on Computational Fluid Dynamics, July 2012.
- [23] Rumsey, C., “NASA Langley Turbulence Modeling Resource (<http://turbmodels.larc.nasa.gov>),” 2015.
- [24] Menter, F. R., “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Application,” *AIAA Journal*, Vol. 32, No. 8, August 1994, pp. 1598–1605.
- [25] Menter, F. R., “Improved Two-Equation k-omega Turbulence Models for Aerodynamic Flows,” Tech. rep., NASA/TM, October 1992.
- [26] Bassi, F., Botti, L., Colombo, A., Crivellini, A., Franchina, N., Ghidoni, A., and Rebay, S., “Very High-Order Accurate Discontinuous Galerkin Computation of Transonic Turbulent Flows on Aeronautical Configurations,” *ADIGMA - A European Initiative on*

- the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Vol. 113, 2005, pp. 25–38.
- [27] Balakumar, P., “Computations of Flow over a Hump Model Using Higher Order Method with Turbulence Modeling,” Tech. rep., AIAA 43rd Aerospace Sciences Meeting, Reno, NV, January 2005.
- [28] 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.
- [29] 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.
- [30] Solin, P., Segeth, K., and Dolezel, I., *Higher-Order Finite Element Methods*, 2003.
- [31] Bassi, F. and Rebay, S., “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics*, Vol. 138, December 1997, pp. 251–285.
- [32] Kreiss, H. and Scherer, G., “Finite element and finite difference methods for hyperbolic partial differential equations,” *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. D. Boor, Academic Press, 1974.
- [33] Strand, B., “Summation by parts for finite difference approximation for  $d/dx$ ,” *Journal of Computational Physics*, Vol. 110, 1994, pp. 47–67.
- [34] Carpenter, M. H., Gottlieb, D., and Abarbanel, S., “The Stability of Numerical Boundary Treatments for Compact High-Order Finite-Difference Schemes,” *Journal of Computational Physics*, Vol. 108, No. 2, 1993, pp. 272 – 295.

- [35] Carpenter, M., Gottlieb, D., and Abarbanel, S., “Time-Staggle Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High-Order Compact Schemes,” ICASE Report 93-9, ICASE, Hampton, VA, March 1993.
- [36] Svård, M., Carpenter, M., and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, far-field boundary conditions,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1020–1038.
- [37] Svård, M. and Nordström, J., “A stable high-order finite difference scheme for the compressible NavierStokes equations, No-slip wall boundary conditions,” *Journal of Computational Physics*, Vol. 227, 2008, pp. 4805–4824.
- [38] Nordström, J., Forsberg, K., Adamsson, C., and Eliasson, P., “Finite Volume Methods, Unstructured Meshes and Strict Stability for Hyperbolic Problems,” *Applied Numerical Mathematics*, Vol. 45, 2003, pp. 453–473.
- [39] Merkle, C., Deshpande, M., and Venkateswaran, S., “Efficient Implementation of Turbulence Modeling in Computational Schemes,” Tech. rep., Proceedings of the Second U.S. National Congress on Computational Mechanics, Pergamon, Oxford, 1993.
- [40] Park, S. H. and Kwon, J. H., “Implementation of  $k$ - $\omega$  Turbulence Models in an Implicit Multigrid Method,” *AIAA Journal*, Vol. 42, 2004.
- [41] Zheng, X. and Liu, F., “Staggered Upwind Method for Solving Navier-Stokes and  $k$ - $\omega$  Turbulence Model Equations,” *AIAA Journal*, Vol. 33, 1995.
- [42] Merci, B., Steelant, J., Vierendeels, J., Riemsdagh, K., and Dick, E., “Computational Treatment of Source Terms in Two-Equation Turbulence Models,” *AIAA Journal*, Vol. 38, 2000.
- [43] Venkateswaran, S. and Merkle, C. L., “Analysis of Preconditioning Methods for the Euler and Navier-Stokes Equations,” Von Karman Institute Lecture Series 2015-0835,



University of Tennessee Space Institute, Mailstop 23, B.H. Goethert Parkway Tullahoma, TN, March 1999.

- [44] Roache, P., “Code Verification by the Method of Manufactured Solutions,” *Transactions of the ASME*, Vol. 124, 2002, pp. 4–10.
- [45] Roy, C., “Review of Code and Solution Verification Procedures for Computational Simulation,” *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.

CHAPTER 6  
ASYMPTOTIC GEOMETRY REPRESENTATION FOR COMPLEX  
CONFIGURATIONS ON STRAND GRIDS

### 6.1 Abstract

The strand-Cartesian approach provides many advantages for complex moving-body flow simulations, including fully-automatic volume grid generation, highly scalable domain connectivity, and high-order accuracy. The purpose of this work is to evaluate methods of handling small-scale features, such as sharp corners and ridges, with strand grids by smoothing the geometry, thus allowing strands to preserve orthogonality regardless of the corner or edge concavity and acuteness. Specifically, we investigate surface smoothing as a function of mesh refinement, creating an “asymptotic geometry”. Results provided qualitatively demonstrate superior strand grid meshing compared to previous methods.

### 6.2 Introduction

In computational fluid dynamics (CFD), mesh generation for complex geometry remains one of the greatest bottlenecks. This issue is only further exacerbated by multiple complex bodies in relative motion. Days or even weeks can be spent by meshing experts generating high-quality viscous meshes around complex multi-body geometry, such as rotorcraft. With the advent of reliable higher-order schemes and improvements in computational hardware, the percentage of time devoted to mesh generation with current methods will only continue to increase relative to total simulation time. Mesh automation is yet to be fully embraced by the CFD community, thus, there is a heavy burden on CFD practitioners for a wide array of complex problems, including rotorcraft and other external aerodynamic applications. Complex systems require ever-increasing mesh sizes, for which scalability becomes

---

Co-Authors: Yushi Yanagita, Shaun Harris, Dalon Work, and Aaron Katz.

a greater issue. Automating viscous mesh generation, preserving spatial and temporal accuracy, and maintaining computational efficiency are currently among the greatest research challenges in CFD today.

The strand-Cartesian approach has shown great potential to alleviate many of these difficulties [1–4]. Strand and Cartesian grids allow the possibility of fully automatic volume grid generation while enhancing scalability and the potential for high-order accuracy. Near solid bodies, 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 6.1(a). Away from solid bodies, adaptive Cartesian grids resolve vortical shedding and wake features with efficient high-order algorithms, shown in Figure 6.1(b). Due to the robust and automatic nature of the strand-Cartesian grid generation process, the technique is easily extensible to moving-body problems for which the grid can readily be regenerated at each time step. Strand and Cartesian grids communicate through implicit overset interpolation [5–7], which is facilitated by the compact grid representation of the strand-Cartesian mesh system. A typical three-dimensional strand-Cartesian grid system may be stored on each processor in a parallel computation, allowing for self-satisfying domain connectivity [1] and reducing the percentage of time needed for intergrid communication [8].

Previous work on strand grids by the authors and others generally appears promising, while certain challenges yet remain. In particular, a recent validation study of the strand mesh approach was performed with an existing general unstructured grid solver [9]. While most of the results of this study appeared favorable, the study also revealed certain difficulties handling sharp corners in the geometry. The method of sharp corner handling used in the study involved bending strands at the roots in order to achieve reasonable grid resolution and smoothness. However, root bending of strands also led to skewness and a loss of orthogonality at the surface, which resulted in an apparent loss of accuracy of the general unstructured solver used in the study. Following this, Work et al. [10] investigated an alternative method of handling sharp corners, “multi-strands,” where multiple strands

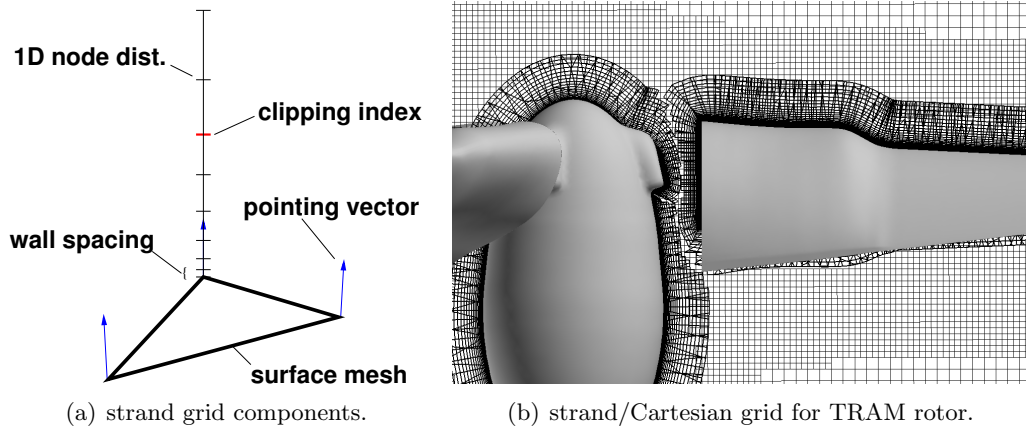


Fig. 6.1: Strand grid elements and example strand/Cartesian grid system for the TRAM rotor.

emanate from a single corner point. This method was compared to strand smoothing and telescoping Cartesian refinement. It was shown that multi-strand method creates mesh discontinuities leading to losses in accuracy. The major conclusion from this work was that for high-aspect ratio grids, smoothed strands with telescoping Cartesian refinement provides the most accurate results and the least complexity. Related to the issue of sharp corner handling is the need for greater near-body accuracy at trailing edges and wing tips in order to efficiently capture wakes and vortices. In recent work by Wissink et. al [11], it was shown how near-body inaccuracy can cause excess diffusion of flow features, despite the use of highly accurate off-body solution techniques.

In light of this recent work, an alternative method of handling small-scale features, such as sharp corners, edges, ridges and saddle points, with strand-Cartesian grids is to be investigated. While smoothing strands at the roots has shown to be an effective method of handling corners, there exists certain geometry for which there are visibility problems for prismatic grids, as detailed by Kallinderis [12] and Sharov et al [13]. In addition, highly smoothed strands may display losses in accuracy. To avoid accuracy losses in this fashion, the most obvious solution is to keep the strands normal to the surface at all times, however, this returns us to the original predicament of sufficient mesh resolution at external corners and avoiding strands crossing at internal corners. A possible solution to fit this criteria is

to smooth the geometry surface itself. In real world application, no corner is truly sharp, however, there exists some resistance among CFD practitioners to adopt this paradigm. An alternative view to preserving the initial geometry is to consider geometry a function of the grid resolution. In this sense, as we refine the mesh, we also allow the geometry to return closer to its original shape, thus, closer to retaining its true small-scale features. We define geometric manipulation in this fashion as “asymptotic geometry.” To accomplish this, the surface is defined implicitly, allowing for the surface to be maneuvered freely through the use of the level sets method [14,15]. Smoothing small-scale features allows strand orthogonality to be preserved without any mesh discontinuities. Where features are more acute, the strand length at the corners are shortened rather than clipped to preserve stencils and allow for Cartesian grids to telescope close to the body to improve resolution and accuracy of wakes and vortical flow features near sharp corners. In order to evaluate this procedure, we employ a number of quantitative and qualitative studies.

The paper is organized as follows: First, small-scale-smoothing via the level sets method is presented, along with a description of previous efforts at handling sharp small-scale features. Second, a brief overview of strand mesh generation procedures is provided, with an emphasis on the procedure for shortening the strands to the curvature. Results for a number of case studies are presented. Finally, we conclude and discuss future work.

### **6.3 Small-Scale Feature Smoothing Procedure**

In this section, we detail the smoothing procedures applied to geometry in order to smooth small-scale features and achieve an asymptotic geometry. The approach is designed to eliminate, or at least reduce, the amount of root-bending to strands in order to provide sufficient coverage at small-scale features. In the proposed method, geometry with small-scale features are smoothed to allow for increased visibility and preservation of strand orthogonality. In order to accommodate this approach to small-scale feature smoothing, we rely heavily on the level set method [14].

### 6.3.1 Level Set Representation

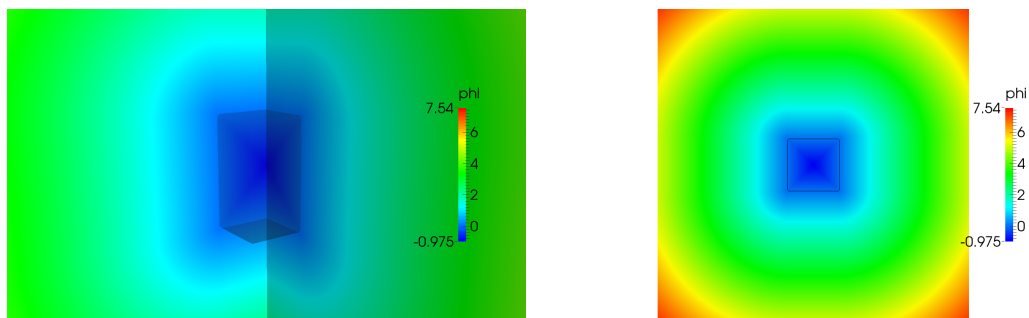
Traditionally, a geometry is represented by a set of discrete points. Should we wish to manipulate the geometry, each point must be reassigned in a rather cumbersome fashion. The method of level sets is an elegant solution that allows for the geometry to be defined implicitly. With this method, we embed our geometry as a propagating interface as the zero level set of a high-dimensional function,  $\phi$ , where

$$\phi(x, y, z, t = 0) = \pm d. \quad (6.1)$$

Here,  $d$  is the distance from an  $x, y, z$  location to our geometry, which is a closed hypersurface,  $\Gamma$ . In Figure 6.2, we provide an example of a distance function field in three-dimensions for a cube. Figure 6.2(a) shows a three-dimensional contour of the distance function. The zero level set, highlighted in black, represents our closed surface  $\Gamma$ . Figure 6.2(b) shows a two-dimensional slice of the distance function field shown in Figure 6.2(a). We denote the area or volume inside  $\Gamma$  with a negative  $\phi$ , and the area or volume outside  $\Gamma$  with a positive  $\phi$ .

The equation of motion of the level set may be represented by

$$\phi_t + F|\nabla\phi| = 0, \quad (6.2)$$



(a) Three-dimensional distance function contour for a cube. (b) Two-dimensional slice of a distance function contour for cube.

Fig. 6.2: Level set representation of a cube.

where,

$$\phi(x, y, z, t = 0) = \text{given.} \quad (6.3)$$

Here,  $F$  is defined as the speed of the flow. For our purposes,  $F$  shall be a function of the curvature,  $\kappa$ , however, it may also be a function of any other parameters desired. Thus we have a formulation to now evolve our curve,  $\Gamma$ , under a speed,  $F$ . With the notation of negative  $\phi$  inside the curve, and positive  $\phi$  outside, a speed of  $F = \kappa$  corresponds to the collapsing curvature flow, since the boundary moves in the direction of its normal with negative speed and hence inwards.

### 6.3.2 Min/Max Flow

In order to evolve our geometry in such a way that small-scale features are smooth, we employ the min/max flow method of Malladi and Sethian [16]. Originally created to remove noise and enhance images, this method may be transferred to fit our purposes by considering our initial geometry with small-scale features as a “noisy” shape.

The min/max flow method requires us to first define two flows,

$$F(\kappa) = \min(\kappa, 0) \quad (6.4)$$

and

$$F(\kappa) = \max(\kappa, 0). \quad (6.5)$$

Flow under  $F(\kappa) = \min(\kappa, 0)$  allows inward concave fingers to grow outwards, while suppressing the motion of outward convex regions. Thus, motion halts as soon as a convex hull is obtained. Conversely, flow under  $F(\kappa) = \max(\kappa, 0)$  allows outward regions to move inwards while suppressing the motion of inward concave regions. Once the shape becomes fully concave, the curvature is always positive, and the flow becomes the same as regular curvature flow; hence collapses the curve to a point.

Simply evolving the flow under either of these flow conditions will not provide us with a sufficient solution, thus, we utilize a switch function. This switch allows us to choose the

”correct” flow in order to smooth our desired features.

$$F_{min/max} = \begin{cases} \min(\kappa, 0) & \text{if } \bar{\phi} < 0 \\ \max(\kappa, 0) & \text{if } \bar{\phi} \geq 0. \end{cases} \quad (6.6)$$

Here,  $\bar{\phi}$  is the average  $\phi$  value in a disk or sphere of radius  $kh$ . We define  $h$  as the discretization size, and  $k$  as some size coefficient. This flow attempts to remove structures of width  $kh$ . By choosing a larger radius to calculate  $\bar{\phi}$ , more features are smoothed. Conversely, by choosing a smaller radius, less features are smoothed. Malladi and Sethian [16] describe this method in further detail.

### 6.3.3 Asymptotic Geometry Representations

Applying the min/max flow on a zero level set with small-scale features, we demonstrate the asymptotic geometry approach. In this section we show a number of examples with and without any smoothing to small-scale features.

#### Cube

First, we demonstrate an asymptotic geometry representation with a cube. Figure 6.3 shows the zero level set of a cube with various levels of smoothing. Here, we increased the number of cells in the grid, decreasing the discretization size, and thus, we decrease the size of what features are smoothed. Figure 6.3(a) shows the zero level set of the cube with no smoothing, while Figures 6.3(b) to 6.3(d) shows the cube with min/max flow applied to an increasing grid size.

We also examine the asymptotic representation of the cube quantitatively. Taking the RMS value of the change in Euclidean distance of the surface nodes, we may obtain an “asymptotic error,” as shown in Figure 6.4. In other words, we obtain a quantitative value for how much our new asymptotic geometry deviates from our original geometry. Similar to a flow solution, we obtain an order of accuracy, where, as we increase the grid resolution, we achieve a lower error. Here, we observe an order of accuracy of 1. A table of the asymptotic



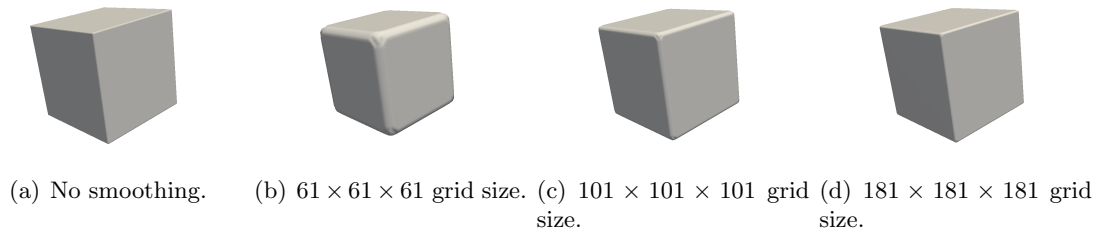


Fig. 6.3: Asymptotic geometry representation for a cube.

error for each of the grid sizes has been provided in Table 6.1.

### Cube with a Sharp Point

Figure 6.5 shows an asymptotic geometry representation applied to a cube with a sharp point. Malladi and Sethian [16] previously showed this example in two-dimensions to demonstrate the removal of the sharp point. Here, we apply a similar methodology to a three-dimensional shape. In Figure 6.5(a) we show the zero level set with no smoothing. Next in Figure 6.5(b), we show the removal of the point using min/max flow.

### Cylinder-Block

Finally in Figure 6.6 we demonstrate an asymptotic geometry representation of a cylinder with an intersecting rectangular block. This shape provides particular interest, as it is

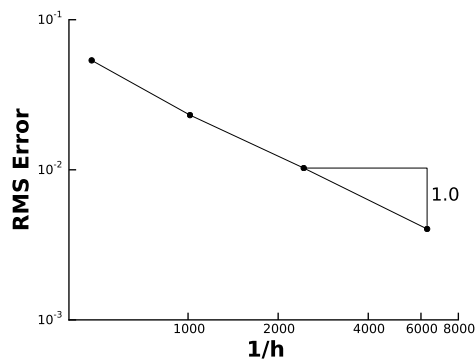
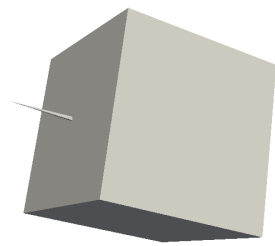
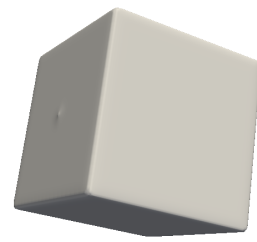
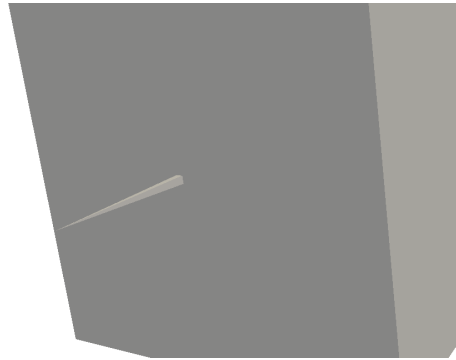


Fig. 6.4: Order of accuracy results for asymptotic geometry for a cube geometry.



(a) No smoothing.

(b)  $117 \times 156 \times 117$  grid size.

(c) No smoothing (Zoom).

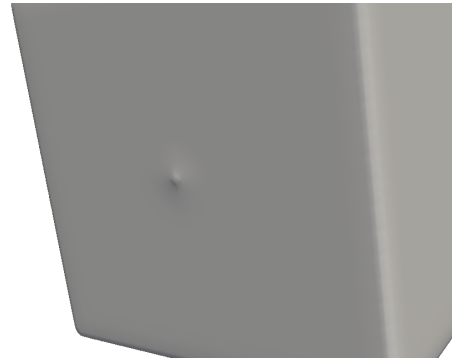
(d)  $117 \times 156 \times 117$  grid size (Zoom).

Fig. 6.5: Asymptotic geometry representation for a cube with a sharp point.

analogous to a representation of a wing-body geometry. Figure 6.12 shows the zero level set of the shape with no smoothing on a relatively coarse grid with a size of  $81 \times 81 \times 61$  cells. On the same grid we apply min/max flow and allow small-scale features to be smoothed. Figure 6.6(b) shows new zero level set. In Section 6.5 we demonstrate strand generation on the smoothed geometry and compare this to previous methods.

#### 6.4 Strand Grid Generation Procedure

In this section, we review basic strand grid generation procedures that are currently in use and provide details of a new procedure to shorten strand grids. Details of strand grid generation procedures may be found by Wissink [17]. Strands are generated from a surface tessellation of the geometry, which can readily be made through CAD or other geometry-meshing software. Figure 6.1(a) shows an illustration of the basic elements of a strand grid. Each strand within the grid consists of a surface node or root, a pointing vector derived from the faces surrounding the node, and a clipping index. The clipping index is used to prevent strands from crossing and forming negative-volume prisms. Points after the clipping index contain boundary conditions interpolated from the surrounding Cartesian mesh.

Spacing of points along a strand are based on a hyperbolic tangent stretching function, and is the same for all strands. Surface point spacing, outer boundary point spacing, and a maximum allowed stretching ratio are prescribed for use in the stretching function. The initial spacing is usually estimated as a function of the Reynolds number to produce  $y^+$  values of 1 or less. To adequately capture boundary layer effects, boundary layer theory is used to estimate the thickness of the boundary layer [18]. The strand length,  $l$ , is then expressed as a multiple,  $K$ , of the boundary layer thickness at some characteristic length

Table 6.1: Asymptotic error for a cube.

Grid Size	Number of Grid Nodes	Error	Order of Accuracy
$61 \times 61 \times 61$	226,981	5.35959e-02	-
$101 \times 101 \times 101$	1,030,301	2.3145e-02	1.110
$181 \times 181 \times 181$	5,929,741	1.026943e-02	0.929
$341 \times 341 \times 341$	39,561,821	4.03453e-03	0.983374

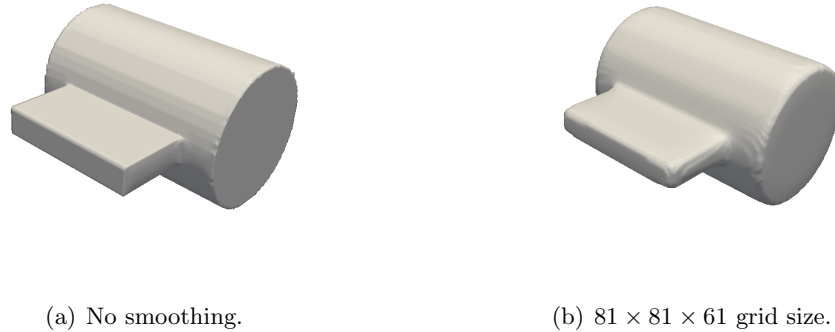


Fig. 6.6: Asymptotic geometry representation for a cylinder with an intersecting rectangular block.

of the problem,  $L$ :

$$l = K\delta(L). \quad (6.7)$$

The characteristic length,  $L$ , may vary for different components or bodies, such as a wing and fuselage [9].

An important process in the strand generation procedure is the computation of the pointing vector of each strand in the mesh. Pointing vectors are initialized as the average of the normals of the faces surrounding each node. The result of this initialization procedure are strand meshes of the type shown in Figure 6.7(a). As the figure shows, many gaps and overlaps exist around sharp corners in the geometry. To reduce these effects, strand direction smoothing is performed. This technique applies a local optimization procedure to make each pointing vector as parallel as possible to its nearest neighbors. The optimization procedure involves the method of Lagrange multipliers applied iteratively in a Jacobi fashion. Convergence of the global mesh smoothing is monitored by a RMS value showing how much the vectors have changed from one iteration to the next. The result of the smoothing procedure is shown in Figure 6.7(b).

While the smoothing procedure provides even spacing around sharp corners, the result is a loss of orthogonality of the strands with respect to the surface. Previous studies have shown that solution accuracy degrades with excessive strand-bending [9]. Efforts to main-

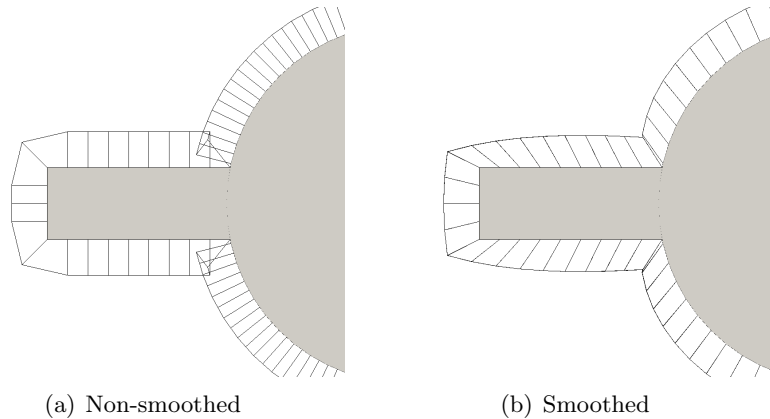


Fig. 6.7: Demonstration of strand direction vector smoothing. (a) Strands with direction vector normal to surface, no smoothing applied, (b) strands with smoothed direction vectors

tain good accuracy around sharp corners have focused on introducing the proper amount of directional smoothing to balance the competing needs for coverage and orthogonality. With too much smoothing, good coverage is obtained, but orthogonality is sacrificed. With too little smoothing, orthogonality is maintained, but coverage is inadequate. The optimal amount of smoothing is problem dependent, requiring user expertise and experience to obtain.

#### 6.4.1 Strand Grid Shortening Procedure

Traditionally, stands are clipped to allow for smooth merging with the Cartesian mesh. This approach has its deficits however. The flux-correction method [19,20] has been applied with success to the strand methodology, however, this method requires a large stencil to calculate gradients with accuracy. Thus, nodes that are clipped may clip the stencil and spoil the accuracy of the method. To counter this, we propose to simply shorten the length of that strand without any clipping, thus preserving the stencil required for high-order calculations. Shortening in this fashion serves multiple advantages over clipping. First, as already mentioned, stencil required for the flux-correction scheme is preserved. Second, by keeping the node spacing along each strand proportional, resolution around small-scale features is increased compared to simply clipping strands at the same length. Thus, less

telescoping Cartesian refinement is required compared to clipping.

To shorten each strand, a shortening factor is determined based on the local curvature of the surface, according to Equation 6.8

$$f(\kappa) = \begin{cases} 1 & \text{if } \max(|\kappa_1|, |\kappa_2|) \leq \kappa_{Threshold} \\ f_{internal} & \text{if } |\kappa_2| \geq \max(|\kappa_1|, \kappa_{Threshold}), \text{ otherwise } 1, \\ f_{external} & \text{if } |\kappa_1| > \max(|\kappa_2|, \kappa_{Threshold}), \text{ otherwise } 1 \end{cases} \quad (6.8)$$

where,  $\kappa_1$  and  $\kappa_2$  represent the maximum and minimum curvatures, respectively, and  $\kappa_{Threshold}$  is a user-defined value which determines the minimum curvature magnitude required to shorten strands at any surface node. Once the shortening factors are determined, an averaging routine is implemented to eliminate any sudden increases or decreases in strand length along the surface. This averaging routine takes the arithmetic mean of the shortening factor at a node and shortening factors of all adjacent nodes. The maximum strand length is then multiplied by the averaged shortening factor for each node to obtain the length of the strand extending from each individual node. The number of times each averaging routine should be implemented is heavily dependent on surface geometry, and surface mesh density around certain geometric features. Currently the number of times both averaging routines are implemented is taken in as user inputs, requiring some user experience and *a priori* knowledge of the geometry. In future works, methods to further automate this process will be investigated in order to automate the strand shortening process.

An example of strand-shortening over a cube geometry can be seen in Figures 6.8 and 6.9. In Figures 6.8, the strand mesh before and after the application of strand-shortening can be compared. As seen in Figure 6.8(b), the strands extending from the edges and corners of the cube have been shortened such that the resulting volume mesh geometry remains smooth. In Figure 6.9, the cross section geometry of the volume mesh, taken parallel to a surface of the cube can be seen. As shown, the strand-shortening routine has successfully shortened the strands while smoothing out any sudden changes in strand lengths, thus eliminating any sudden changes in grid size from strand to strand. Here, the

increase in mesh resolution near the surface of the edge of the cube is observed. Due to the strand-shortening, the aspect ratio of each cell is increased, thus providing better resolution for small-scale turbulent features that are generated at edges and corners. In Figures 6.9(c) and 6.9(d), a close up view of the corner has been provided. Further away from the corner in the non-shortened cube, large, long cells are created at the corners, providing poor wake refining ability, as shown in Work et al. [10]. By shortening the strands, these cells have their aspect ratio reduced, closer to 1, allowing for better wake refining capabilities in critical areas.

Next, we examine shortening strands on an interior duct-style mesh. Figure 6.10 shows a duct with and without strand-shortening. Without clipping or shortening, strands cross at the interior corners, as shown in Figure 6.10(a). While clipping alleviates this issue, the issue of disrupting the higher-order stencils remains. Figure 6.10(b) demonstrates how strand-shortening successfully eliminates crossing strands while maintaining a smooth, continuous volume mesh. No crossing is evident, and clipping is not required, thus preserving higher-order stencils. Figure 6.11 shows a two-dimensional slice for a clearer view of the strand shortening method compared to no shortening.

## 6.5 Strand Grid Generation on Asymptotic Geometry

In this section, we present a case study that highlight the advantages and disadvantages of the previously described approaches to accommodating asymptotic geometry with strand shortening. Here, we refer to “smooth” as an asymptotic geometry with strand shortening, and “non-smooth” as the original geometry with traditional root-bending.

### 6.5.1 Cylinder-Block Geometry

Here, we examine the cylinder with an intersecting block from Section 6.3.3. To obtain an asymptotic representation, the geometry was run on an  $81 \times 81 \times 61$  level set grid. Figure 6.12 shows the volume mesh of the geometry with current methods and new methods, root-bending and asymptotic geometry with strand-shortening, respectively. No root-bending was applied on the smooth geometry, while significant root-bending was applied to

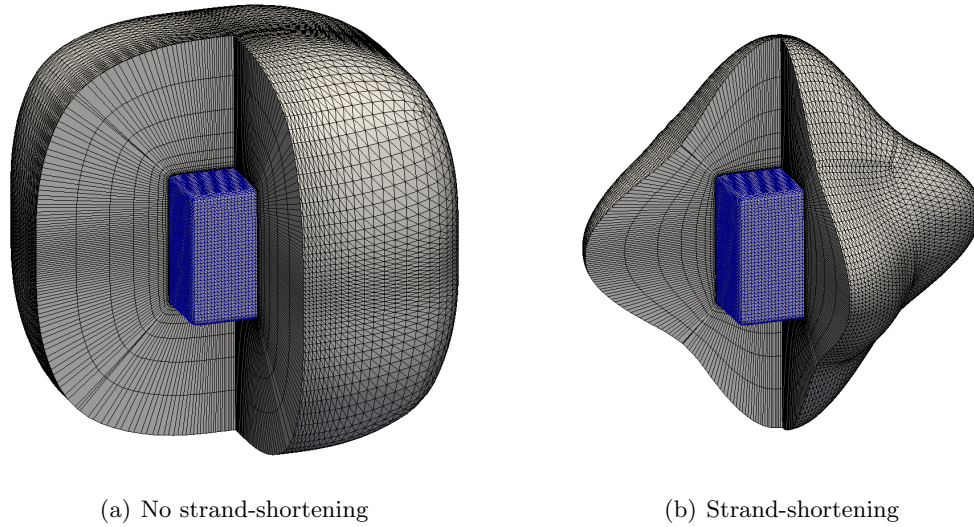


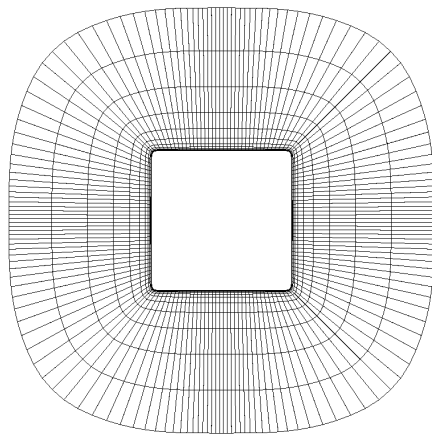
Fig. 6.8: Volume mesh of a cube with and without strand-shortening.

the non-smoothed geometry. Even with significant root-bending, crossing is evident at the internal corners. This can be seen more clearly in Figure 6.13, where two-dimensional slices of the  $z$ -plane and  $y$ -plane are presented. The asymptotic representation shows smooth transitions where small-scale features occur. Combining this with strand shortening avoids any strands crossing without the use of root-bending, and increased mesh resolution in areas with increased curvature. In both the  $y$ - and  $z$ -planes, the non-smooth geometry shows crossing at the internal corners. Further examination of the cylinder-block case can be made with a close-up of the interior corner, as shown in Figure 6.14. From the figure, it is clear the smooth geometry shows a superior mesh to the non-smooth case.

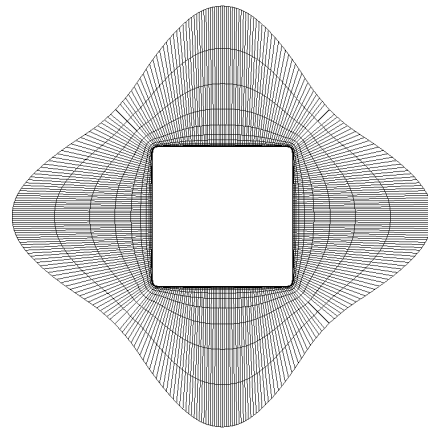
## 6.6 Conclusions and Future Work

A new method is proposed in this paper to obtain coverage around small-scale features using strand grids with the goal of maintaining orthogonality of strands with the surface. The method smoothes small-scale features via a level set method implementation, referred to in this work as “asymptotic geometry”. Small-scale feature smoothing is performed as a function of grid resolution to ensure adequate mesh resolution around these features. As

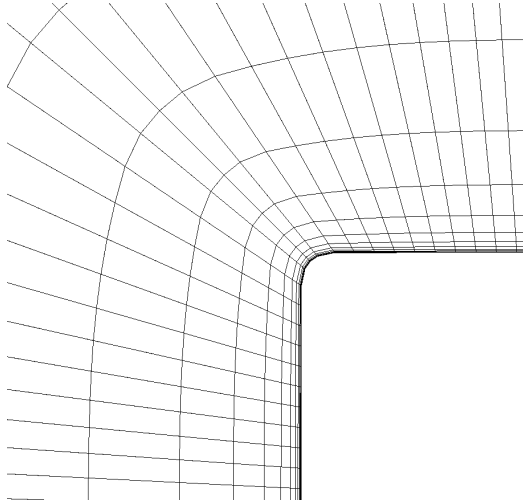




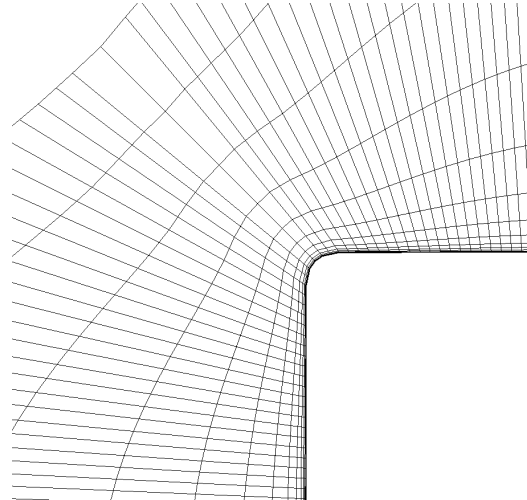
(a) No strand-shortening



(b) Strand-shortening



(c) No strand-shortening (Corner zoom)



(d) Strand-shortening (Corner zoom)

Fig. 6.9: Two-dimensional slice of a cube with and without strand-shortening.

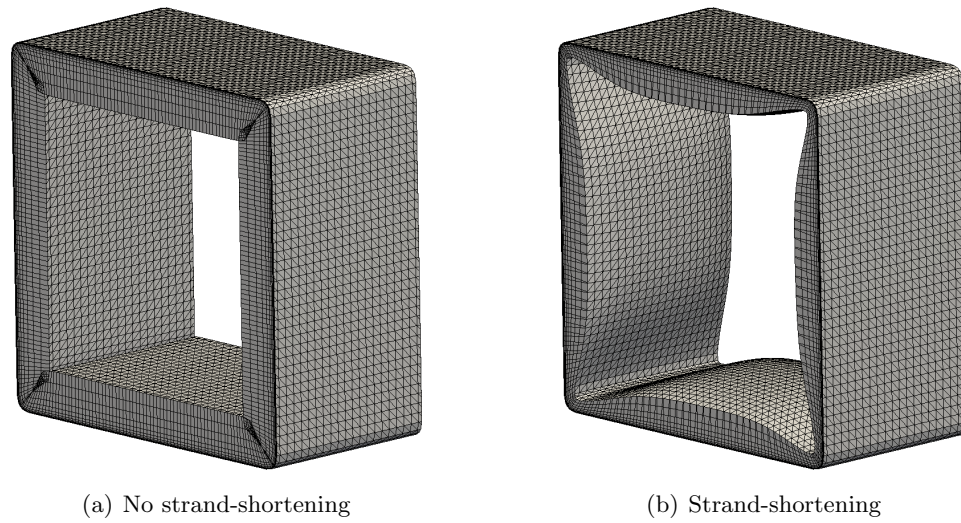
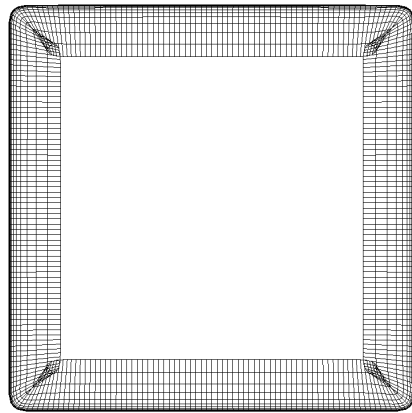


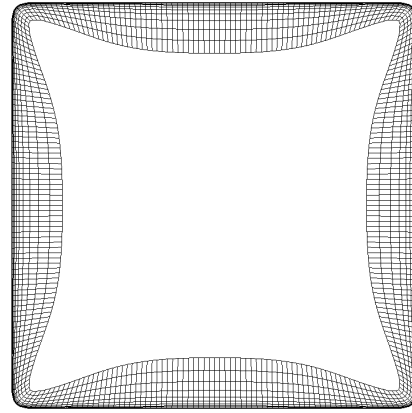
Fig. 6.10: Volume mesh of a duct with and without strand-shortening.

a result, strand direction smoothing is kept to a minimum, and orthogonality is generally preserved. Two geometries are identified, a square cylinder and a NACA 0012 airfoil, for which corners have been smoothed to demonstrate the method. In addition to smoothing corners and edges, a strand-shortening method is presented. Shortening strands in this manner preserves the large stencils required for gradient calculations in the high-order flux-correction method. Some example cases are presented.

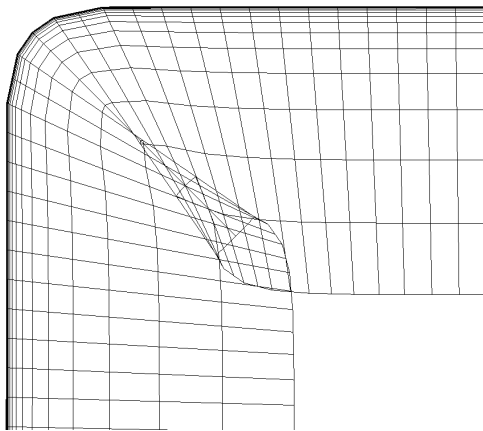
The primary focus of future work will be assessing the impact of asymptotic geometry on flow solutions. In addition, methods to reconstruct high-order surfaces will also be investigated, with a future goal of achieving adaptive strand meshing. Adaptive strand meshing will allow for more elements in critical areas, thus reducing the need to smooth strands to provide better coverage.



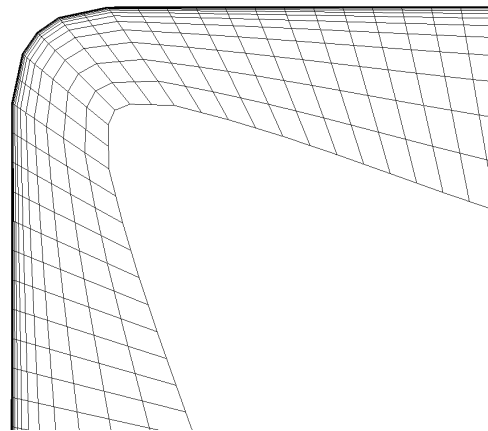
(a) No strand-shortening



(b) Strand-shortening



(c) No strand-shortening (Corner zoom)



(d) Strand-shortening (Corner zoom)

Fig. 6.11: Two-dimensional slice of a duct with and without strand-shortening.

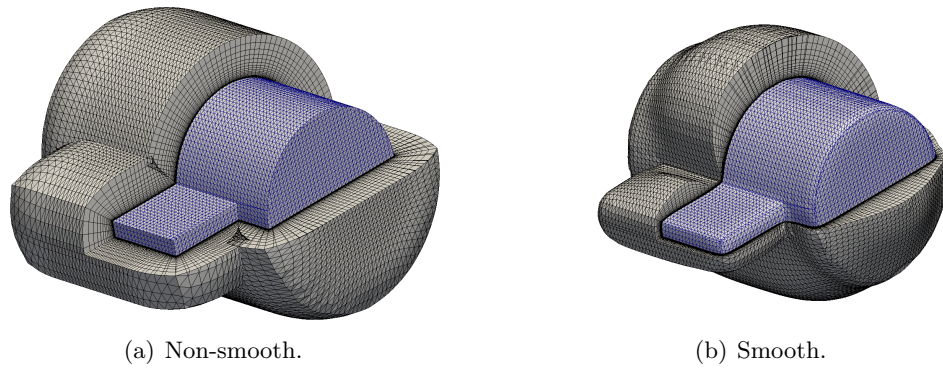


Fig. 6.12: Volume mesh of cylinder-block.

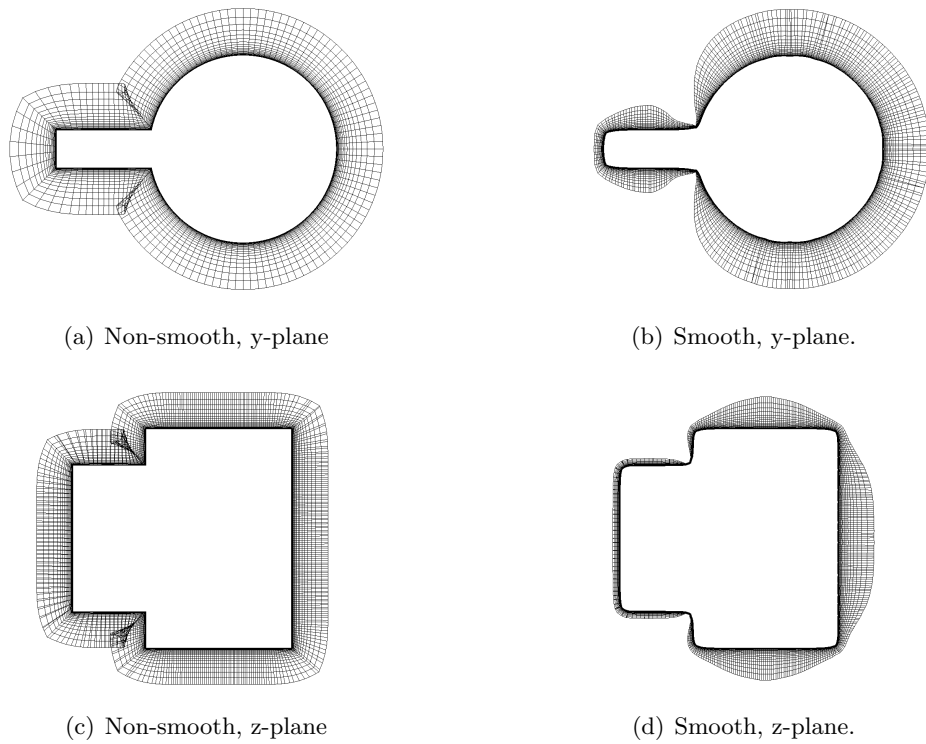


Fig. 6.13: Two-dimensional slices of cylinder-block.

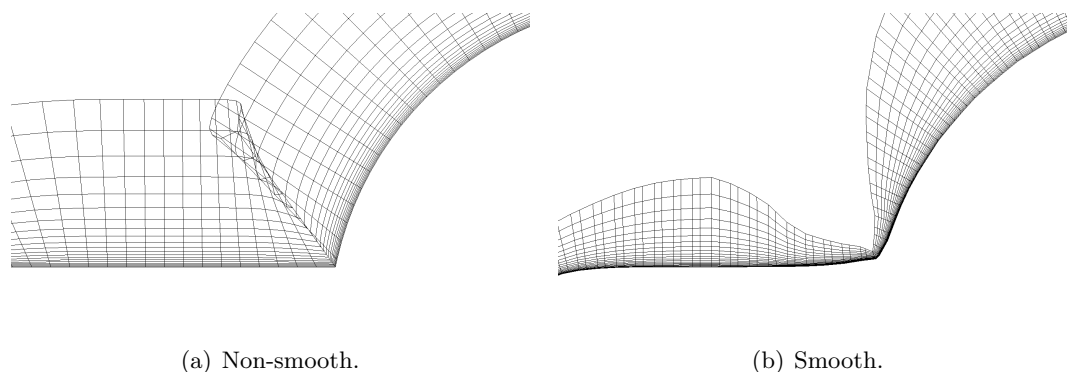


Fig. 6.14: Close-up of cylinder-block internal corner.

## REFERENCES

- [1] Meakin, R., Wissink, A., Chan, W., Pandya, S., and Sitaraman, J., “On Strand Grids for Complex Flows,” *AIAA paper 2007-3834*, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.
- [2] Wissink, A., Potsdam, M., Sankaran, V., Sitaraman, J., Yang, Z., and Mavriplis, D., “A Coupled Unstructured-Adaptive Cartesian CFD Approach for Hover Prediction,” Tech. rep., American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 2010.
- [3] Wissink, A., Katz, A., Chan, W., and Meakin, R., “Validation of the Strand Grid Approach,” *AIAA paper 2009-3792*, AIAA 19th Computational Fluid Dynamics Conference, San Antonio, TX, June 2009.
- [4] Katz, A., Wissink, A., Sitaraman, J., and Sankaran, V., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *AIAA paper 2010-4934*, 28th AIAA Applied Aerodynamics Conference, Chicago, June 2010.
- [5] Steger, J., Dougherty, F., and Benek, J., “A Chimera Grid Scheme,” Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.

- [6] Benek, J. A., Steger, J. L., and Dougherty, F. C., “A Flexible Grid Embedding Technique with Application to the Euler Equations,” *AIAA paper* 1983-1944, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.
- [7] Lee, Y.-L. and Baeder, J., “Implicit Hole Cutting – A New Approach to Overset Grid Connectivity,” *AIAA paper* 2003-4128, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
- [8] Sitaraman, J., Floros, M., Wissink, A., and Potsdam, M., “Parallel Domain Connectivity Algorithm for Unsteady Flow Computations Using Overlapping and Adaptive Grids,” *Journal of Computational Physics*, Vol. 229, 2008, pp. 4703–4723.
- [9] Katz, A., Wissink, A., Sankaran, V., Meakin, R., and Sitaraman, J., “Application of Strand Meshes to Complex Aerodynamic Flow Fields,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 6512–6530.
- [10] 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.
- [11] Wissink, A., “Helios Solver Developments Including Strand Meshes,” Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.
- [12] Kallinderis, Y. and Ward, S., “Prismatic Grid Generation for Three-Dimensional Complex Geometries,” *AIAA Journal*, Vol. 31, 1993, pp. 1850–1856.
- [13] Sharov, D., Luo, H., Baum, J., and Löhner, R., “Unstructured Navier-Stokes grid generation at corners and ridges,” *International Journal for Numerical Methods in Fluids*, Vol. 43, 2003, pp. 717–728.
- [14] Osher, S. and Sethian, J., “Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations,” *Journal of Computational Physics*, Vol. 79, 1988, pp. 12–49.
- [15] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer, 2003.

- [16] Malladi, R. and Sethian, J., “Image Processing: Flows under Min/Max Curvature and Mean Curvature,” *Graphical Models and Image Processing*, Vol. 58, 1996, pp. 127–141.
- [17] Wissink, A., Katz, A., Sitaraman, J., and Haines, B., “Strand Meshing and Solver Developments within PICASSO,” Tech. rep., 51st AIAA Aerospace Sciences Meeting, Grapevine, TX, January 2013.
- [18] Schlichting, H., *Boundary-Layer Theory (7th Edition)*, McGraw-Hill, New York, 1979.
- [19] Katz, A. and Work, D., “High-Order Flux Correction/Finite Difference Schemes for Strand Grids,” *AIAA paper*, AIAA 52nd Aerospace Sciences Meeting, National Harbor, MD, January 2014.
- [20] Tong, O., Work, D., and Katz, A., “High-Order Methods for Turbulence Using Strand Grids,” Tech. Rep. ICCFD8-0215, 8th International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 2014.

## CHAPTER 7

### CONCLUSIONS AND FUTURE WORK

Presented in this work is the development of a novel high-order flux correction method. High-fidelity solutions are obtained by combining flux correction in the unstructured plane and summation-by-parts operators in the strand direction, to achieve a unique strand grid specific method. The flux correction method cancels low-order truncation error with accurate flux and solution gradients, thereby achieving a formal accuracy of 3, however, higher orders are often obtained, especially for highly viscous flows, such as those seen in modelling rotorcraft.

The scheme was extended to high-Reynolds number computations in both two and three dimensions. Turbulence closure was achieved with a robust version of the Spalart-Allmaras turbulence model that accommodates negative values of the turbulence working variable, and the Menter SST turbulence model which blends the  $k$ - $\epsilon$  and  $k$ - $\omega$  turbulence models. A major advantage of this high-order formulation is the ability to implement traditional finite volume-like limiters to cleanly capture shocks and discontinuities. We explored this approach in this work via a symmetric limited positive (SLIP) limiter.

Extensive verification and validation was conducted in two and three dimensions to determine the accuracy and fidelity of the scheme for a number of different cases. Verification studies showed that the scheme achieved better than third order accuracy for low and high-Reynolds number flows. Cost studies show that in three-dimensions, the third-order flux correction scheme required only 30% more walltime than a traditional second-order scheme on strand grids to achieve the same level of convergence.

In order to overcome meshing issues at sharp corners and other small-scale features, a unique approach to traditional geometry, coined “asymptotic geometry,” was explored. Asymptotic geometry is achieved by filtering out small-scale features in a level set domain



through min/max flow. This approach was combined with a curvature based strand shortening strategy in order to qualitatively improve strand grid mesh quality.

While this work appears promising in terms of accurately computing complex turbulent flows in three-dimensions, the test cases presented here are rather simple geometrically. Future work will focus on extending our high-order methods to geometries of practical interest involving multiple bodies in motion, such as rotorcraft. This will require coupling with off-body Cartesian grids, an eventual goal. Various strategies are under consideration for the best way to use the method of level sets to reconstruct high-order surfaces for arbitrary geometry.

## CURRICULUM VITAE

**Oisin Tong****Education****• PhD in Mechanical Engineering**

Utah State University, Logan, UT, May 2016 (expected)

- Dissertation: Development of Three-Dimensional High-Order Strand Grid Methods
- Advisor: Dr. Aaron Katz

**• MSc in Mechanical Engineering**

Utah State University, Logan, UT, December 2013

- Thesis: Verification and Validation of the Spalart-Allmaras Turbulence Model for Strand Grids
- Advisor: Dr. Aaron Katz

**• BEng in Aeronautical Engineering (1.1)**

University of Limerick, Limerick, Ireland, May 2011

- Undergraduate Thesis: Low Mach Number Airfoil Stall
- Advisor: Dr. Andrew Niven

**Positions Held****• Graduate Research Assistant**

MAE Department, Utah State University, Logan, UT, 2012-Present

- Source code development on the high-order flux correction scheme on strand grids.
- Investigating strand grid methods for complex geometries.

- **Design Engineer**

Bombardier Aerospace, Belfast, Northern Ireland, 2009-2010

- Fuselage repair design for discrepancies and errors that occurred during aircraft assembly and manufacture.

## Teaching

- **Instructor**

MAE Department, Utah State University, Logan, UT

- Aerodynamics, Fall 2014

- **Teaching Assistant**

MAE Department, Utah State University, Logan, UT

- Computational Fluid Dynamics, Spring 2013 - 2015
- Numerical Methods, Fall 2012

## Research Interests

Computational Fluid Dynamics, Strand Grids, Reynolds-Averaged Navier-Stokes Turbulence Models, High-Order Methods, Level-Set Methods, Flux Correction

## Computational Skills

Fortran, MatLab, Python, L<sup>A</sup>T<sub>E</sub>X, C++, UNIX, Paraview, Pointwise

## Academic Honors

Dean's List of Distinguished Students

Vice President's Fellowship at Utah State University

### Professional Memberships

AIAA - Since 2011

### Leadership Activities

**Head Coach**, Utah State University Men's Rugby, Logan, UT, 2013-Present

### Published Journal Articles

- J. Thorne, A. Katz, **O. Tong**, Y. Yanagita and K. Delaney, High-Order Strand Grid Methods for Low Speed and Incompressible Flows, *In Publication*.
- **O. Tong**, A. Katz and Y. Yanagita, Verification and Validation of a High-Order Strand Grid Method for Two-Dimensional Turbulent Flows, *In Review*.
- **O. Tong**, A. Katz, Y. Yanagita, A. Casey and R. Schaap, High-Order Methods for Turbulent Flows on Three-Dimensional Strand Grids, *Journal of Scientific Computing*, 2015, DOI: 10.1007/s10915-015-0070-z.
- D. Work, **O. Tong**, R. Workman, A. Katz and A. Wissink, Strand-Grid-Solution Procedures for Sharp Corners, *AIAA Journal*, 2014, DOI: 10.2514/1.J052607.

### Published Conference Papers

- **O. Tong**, Y. Yanagita, S. Harris, D. Work and A. Katz, Asymptotic Geometry Representation for Complex Configurations on Strand Grids. *54th AIAA Aerospace Sciences Meeting*, January 2016, San Diego, CA., AIAA-2016-1584.
- **O. Tong**, C. Blakely, R. Schaap and A. Katz, Assessment of a Two-Equation Turbulence Model in the High-Order Flux Correction Scheme. *54th AIAA Aerospace Sciences Meeting*, January 2016, San Diego, CA., AIAA-2016-1582.
- Y. Yanagita, **O. Tong** and A. Katz, Critical Evaluation of Turbulence Flows for High-Order Strand Grid Methods *54th AIAA Aerospace Sciences Meeting*, January 2016, San Diego, CA., AIAA-2016-1350.

- J. Thorne, A. Katz, **O. Tong** and Y. Yanagita, High-Order Strand Grid Methods for Low Mach and Incompressible Flow. *22nd AIAA CFD Conference*, June 2015, Dallas, TX., AIAA-2015-3200.
- **O. Tong**, Y. Yanagita, R. Schaap, S. Harris and A. Katz, High-Order Strand Grid Methods for Shock Turbulence Interaction. *22nd AIAA CFD Conference*, June 2015, Dallas, TX., AIAA-2015-2283.
- **O. Tong**, A. Katz, A. Wissink and J. Sitaraman, High-Order Methods for Three-Dimensional Strand Grids. *53rd AIAA Aerospace Sciences Meeting*, Jan 2015, Kissimmee, FL., AIAA-2015-0835.
- **O. Tong**, D. Work and A. Katz, High Order Methods for Turbulence Using Strand Grids. *ICCFD8*, July 2014, Chengdu, China, ICCFD8-2014-0215.
- **O. Tong**, A. Katz, D. Work and A. Wissink, Turbulent Flow Validation in the Helios Strand Solver. *52nd AIAA Aerospace Sciences Meeting*, Jan 2014, National Harbor, MD., AIAA-2014-0083.
- D. Work, **O. Tong**, R. Workman, A. Katz and A. Wissink, Strand Grid Solution Procedures for Sharp Corners. *51st AIAA Aerospace Sciences Meeting*, Jan 2013, Dallas, TX., AIAA-2013-0800.