Abstract

REWERTS, SCOTT D. Aerodynamic Applications of Incompressible Navier-Stokes Flows at High Reynolds Number Simulations Using the Immersed Boundary Method. (Under the direction of Dr. Jack R. Edwards.)

A method for unsteady simulation of incompressible viscous flows is presented and applied to aerodynamic applications at high Reynolds numbers. The Immersed Boundary (IB) method is integrated into an existing incompressible unsteady Navier-Stokes (NS) solver. Prescribed surface velocity and a near-body interpolation procedure constrain the fluid behavior near surfaces. An Artificial Compressibility (AC) term, which tends to zero after iteration between discrete physical time levels, is added to the continuity equation to keep the continuity and momentum equations strongly coupled during the flow field's evolution. Finite-volume spatial fluxes are reconstructed with upwinding using the Low-Diffusion Flux Splitting Scheme (LDFSS). Oscillations are prevented from building up during high-order variable extrapolation using Total-Variation-Diminishing (TVD) and Weighted Essentially Non-Oscillatory (WENO) limiters. Turbulent effects are included using the Edwards variant of the one-equation Spalart-Allmaras (SA-E) model. The solution is advanced in time using a dual-stepping scheme. Extrapolations of flow field conditions such as pressure back to the surface are applied. A novel extension to thin baffle surfaces is applied. Sparsely populated surface pressures are extended to the full collection of nodes, and various calculations of forces and moments are implemented.

Steady flows considered in this study include a symmetric single-element airfoil with and without a lower surface trailing-edge aerodynamic device, and a cambered single-element airfoil with and without leading edge contamination from ice accretion. An unsteady flow is also investigated by deploying the slat and flap of a cambered multiple-element airfoil.
over a brief amount of time. All of these geometries consist of a stretched Cartesian grid defining the fluid volume, and a collection of nodes and normal vectors defining surface shape, orientation and velocity. Additionally, as a proof-of-concept for the thin-baffle extension, a steady three-dimensional case was run simulating airflow through each a complex and simple duct (both around a turning vane).

This method’s results for pressure distribution, forces and moments are compared against published experimental data. Although the method seems to under predict the magnitude of potentially compressible suction and suction peaks on bodies generating high values of lift at high Reynolds numbers, good agreement in trend is demonstrated with experimental data, and overall agreement is reasonable enough for coarse, rapid design work and analysis.
Simulating Aerodynamic Applications at High Reynolds Numbers Using the Immersed Boundary Method

by
Scott David Rewerts

A thesis submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the degree of Master of Science

Aerospace Engineering

Raleigh, North Carolina

2010

APPROVED BY:

____________________                                          ____________________
Dr. Hong Luo                                                  Dr. Ashok Gopalarathnam

____________________
Dr. Jack R. Edwards, Jr.
Chair of Advisory Committee
**Biography**

Scott David Rewerts was born November 5th, 1980 to David G. Rewerts and Barbara L. Rewerts in Lexington, NC. He grew up in and around the town of his birth, attending Lexington Senior High School while swimming competitively and playing piano. Scott began undergraduate studies in Aerospace Engineering at North Carolina State University in Raleigh in the fall of 1999. In May of 2003, he earned a Bachelor’s of Science degree with a minor in piano performance, and started graduate school briefly thereafter under advisor and professor Dr. Jack R. Edwards, Jr. On July 24th, 2004, Scott was married to Katherine Paige Clement.

In the summer of 2005, Scott completed his graduate classes. He accepted a job in Albuquerque, NM at Eclipse Aviation as an Aerodynamics Engineer, in hopes of contributing to a shift away from hub-and-spoke travel for middle-class travelers in the US. While employed at Eclipse, Scott helped the company win a Collier Trophy award and achieve type certification, a production certificate, RVSM and FIKI supplemental certifications, and a high degree of success in a drag reduction program, all on the Eclipse 500 very light jet.

In early 2008, Scott returned to Greensboro, NC to take a similar position on the Honda Jet program at Honda Aircraft Company, Inc. He remains there at the time of completion of this work. After completing his degree, Scott hopes to remain academically involved, and continue developing into a more capable engineer.
Acknowledgements

Scott D. Rewerts would like to thank the following:

Dr. Jack R. Edwards, Jr. for his advice, guidance and instruction;

Dr. Ashok Gopalarathnam, for his instruction and service on my thesis committee;

Dr. Hong Luo, for serving on my thesis committee;

Corvid Technologies (previously Aletheon Technologies), for originally funding this research;

My family and friends, for their support;

All of the above, for patience and providing Scott the opportunity to continue his studies whenever he could pursue them.

The Free and Open Source Software (FOSS) movement, without which completing this work at home would have been much more difficult. FOSS software utilized through the course of this work included compilers (G95 and GCC), libraries (ANN and MPICH2), image editing & office software (GIMP and OpenOffice), plotting (ParaView), and operating system (Ubuntu Linux).
# Table of Contents

List of Figures..........................................................iv

1 - Introduction...........................................................................1

2 - Methodology.................................................................3

2.1 - Model Equations As A Subset of General Governing Equations........3

2.2 - Artificial Compressibility Method........................................6

2.3 - Vector Format of Model Equations & Cell-Centered Finite Volume Method

Construction from Differential Formulation..........................8

2.4 – Interface Flux Vector Splitting, Upwinding, and High-Order Extension....13

2.5 - Immersed Boundary Method As Boundary Condition....................19

2.6 - Extension of Immersed Boundary Method to Zero-Thickness Thin-Baffles.29

2.7 - Solution Advancement Through Pseudo-Time and Time.................30

2.8 – Edwards Variant of Spalart-Allmaras Turbulence Model..................33

2.9 - Force and Moment Calculation...........................................36

2.10 - Grid Generation of Immersing Grid and Surface Nodes...............45

3 - Results.............................................................................46

3.1 - Ducts With Thin-Baffle Turning Vane..................................46

3.2 - Single-Element Symmetric NACA0012 Airfoil and Trailing Edge Device....51

3.3 - Single-Element Cambered NACA23012 Airfoil and Leading Edge Ice

Contamination...........................................................................62

3.4 - Multi-element Airfoil With High-Lift Devices Extending Over Time....70

4 - Conclusions........................................................................81

5 - Recommendations for Future Research....................................83

6 - References..........................................................................84
List of Figures

Figure 2.5.1 – Expanding Off-Body Interpolation Stencil......................................................24
Figure 2.5.2 – Various Two-Dimensional Power-Law Velocity Profiles................................28
Figure 2.9.1 – Lift History on NACA0012 ($\alpha=16^\circ$ & Rec = 2e6)..................................44
Figure 3.1.1 – 3D Complex Rectangular Duct Grid Block, Surface Nodes & Vectors...........47
Figure 3.1.2 – Complex Duct Sparse Grid, Distance Function Contour and Poor Blanked
Heaviside Representation of Smooth Surface...........................................................................48
Figure 3.1.3 – Poor Tangency Resulting from Poor Grid Representation of Surface..........49
Figure 3.1.4 – Good Tangency with Proper Grid Representation of Surface.......................50
Figure 3.1.5 – Contours of Velocity in the Direction of the Length of the Duct..................50
Figure 3.1.6 – Straight Duct Pressure Contours.................................................................50
Figure 3.2.1 – Surface Nodes comprising NACA0012 with 1% Gurney Flap on Lower
Surface Trailing Edge........................................................................................................52
Figure 3.2.2 – Normal Vector Orientation of Surface Nodes comprising NACA0012 with
1% Gurney Flap on Lower Surface Trailing Edge.................................................................52
Figure 3.2.3 – Far field view of Stretched Cartesian Grid Containing NACA0012..........53
Figure 3.2.4 – Near-body view of Stretched Cartesian Grid containing NACA0012
Surface Nodes.........................................................................................................................53
Figure 3.2.5 – Leading Edge Zoom of Stretched, Blanked Cartesian Grid for Clean
NACA0012 Airfoil....................................................................................................................54
Figure 3.2.6 – Trailing Edge Zoom of Stretched, Blanked Cartesian Grid for Clean
NACA0012 Airfoil....................................................................................................................54
Figure 3.2.7 – Nondimensional Velocity Magnitude Contour Plot of 1% Gurney Flap
Configuration at Zero Angle of Attack...............................................................................55
Figure 3.2.8 – Calculated Pressure Coefficient Distributions for Clean NACA0012 and
1% Chord Gurney Flap Configurations at Zero Angle of Attack........................................56
Figure 3.2.9 – Calculated Pressure Coefficient Distributions for Clean NACA0012 and 1% Chord Gurney Flap Configurations at Four Degrees Angle of Attack..................56
Figure 3.2.10 – Calculated and Published Experimental Section Lift Coefficient versus Angle of Attack for both NACA0012 Configurations..................................................57
Figure 3.2.11 – Calculated and Published Experimental Section Drag Coefficient versus Angle of Attack for both NACA0012 Configurations..................................................58
Figure 3.2.12 – Calculated and Published Experimental Section Quarter-Chord Moment Coefficient versus Angle of Attack for both NACA0012 Configurations...........58
Figure 3.2.13 – Streamline Flow Visualization of Immersed Boundary Code Results Behind 1% Chord Gurney Flap At Zero Angle of Attack..................................................61
Figure 3.2.14 – Streamline Flow Visualization of Immersed Boundary Code Results behind 1% Chord Gurney Flap at 10 Degrees Angle of Attack.................................61
Figure 3.3.1 – Surface Nodes for NACA23012 With Leading Edge Contamination......63
Figure 3.3.2 – NACA23012 Vector Orientation with Leading Edge Contamination......63
Figure 3.3.3 – Clean & Ice-Contaminated NACA23012 with Grid Representation........64
Figure 3.3.4 – Streamline Flow Visualization and Velocity Magnitude Contours Predicted by Immersed Boundary Code At Zero Angle of Attack Iced NACA23012 ..............65
Figure 3.3.5 – Streamline Flow Visualization and Velocity Magnitude Contours Predicted by Immersed Boundary Code At $\alpha = 8^\circ$ Iced NACA23012.................................65
Figure 3.3.6 – Clean & Iced Pressure Distributions at Zero Degrees Angle of Attack ....66
Figure 3.3.7 – Clean & Iced Pressure Distributions at Four Degrees Angle of Attack....66
Figure 3.3.8 – Clean & Iced Pressure Distributions at Eight Degrees Angle of Attack....67
Figure 3.3.9 – Section Lift Coefficient Curves for Clean & Iced LE NACA23012...........68
Figure 3.3.10 – Section Drag Coefficient Curves for Clean and Iced LE NACA23012...68
Figure 3.3.11 – Section Quarter-Chord Pitching Moment Coefficient Curves for Clean and Iced Leading Edge NACA23012.................................................................69
Figure 3.4.1 – Surfaces & Normal Vectors (0, 20, 40, 65 & 100% of Extension).........73
Figure 3.4.2 – Blanked Distance Function & Streamlines for Moving-Body Multi-
Element Airfoil Case at 0, 10, 40, 70 and 100% of Extension Motion.............................74
Figure 3.4.3 – Velocity Magnitude Contours for Moving-Body Multi-Element Airfoil at
0%, 10%, 20%, 30%, 60% and 100% of Extension Motion..................................................76
Figure 3.4.4 – Pressure Coefficient Contours for Moving-Body Multi-Element Airfoil at
0%, 10%, 20%, 30%, 60% and 100% of Extension Motion..................................................77
Figure 3.4.5 – Experimental and Computational Surface Pressure Coefficient Contours
on Multi-Element Airfoil at In Steady State after Extension...............................................78
Figure 3.4.6 – Lift Coeff. Time History During Multi-Element Extension Motion.........79
Figure 3.4.7 – Drag Coeff. Time History During Multi-Element Extension Motion......80
Figure 3.4.8 – Pitching Moment Coeff. History During Extension Motion...............80
1 - Introduction

In computational fluid dynamics (CFD) studies, grid generation for both steady and unsteady problems can be tedious and time-consuming. This may be particularly true on complicated surface geometries, situations involving design work (with rapid and significant geometry changes), and simulations with moving bodies or developing geometries (where the grid may need to be generated at each discrete time step). Furthermore, the inherent accuracy of discretizations of the fluid flow equations on simple Cartesian grids must be restored on non-Cartesian grids by high-order schemes and grid refinement near surfaces. Therefore, a technique which could ease grid generation and allow the use of Cartesian grids for viscous problems at arbitrary Reynolds numbers could reduce project time spans and possibly enhance formulation accuracy away from surfaces in the flow field.

Existing methods in widespread use implement solutions to portions of the challenges described above. Cut-cell type Euler flow solvers with coupled boundary layer solvers have been used in industry for a number of years [2, 4]. Typical unstructured methods allow mostly automated, robust grid generation and adaptation around complex geometries; however, they inherently sacrifice computational efficiency, and a current lack of established grid guidelines may impact viscous solution accuracy [46, 49]. Overset structured methods ease meshing around complicated geometries and allow mesh
motion, but primarily rely on injection (rather than conservation) of properties at overlapping boundaries [33]. Spring-analogy mesh deformations allow mesh movement, but are limited to small motions and may highly skew the resulting grids [6, 41].

The Immersed Boundary method has potential to relieve all of these requirements simultaneously. This is accomplished by allowing grids of arbitrary orientation to exist inside an arbitrarily shaped moving body, and only solving the equation set on the exterior. This allows stretched Cartesian grids to reasonably capture complex geometries, eliminating the need for curved, highly distorted and skewed body-fitted grids. With proper treatment, it may be possible to require only moderate refinement near surfaces.

This work extends Fadlun’s direct-forcing variant [21] of Peskin’s original Immersed Boundary technique [36], from its previous formulation and applications in low Reynolds number flows to relatively high Reynolds number aerodynamic flows. Essentially, forcing functions are applied to the momentum equations near surfaces in an iterative fashion until the nearby surface velocity is effectively prescribed upon the local fluid elements. The required forces and moments for achieving the prescribed movement can then be computed. In most applications in this work, that surface velocity is defined by the simple stationary non-slip condition.

The simulation methods detailed in this work have previously been applied to a variety of both classical test cases and modern applications chosen outside of the aerospace industry. Select examples include elementary flows around a cylinder and sphere and both flow and transport results involving basic geometric elements or complex
human shapes disturbing and interacting with stagnant air containing various dilute-phase particle fields [34, 16]. Examples of existing aerospace results included preliminary studies carried out on each a steady single-element symmetric airfoil [34, 16] and a pitching cambered airfoil [32].

This work extends the application to several problems which are typical of internal and external computational investigations in the aerospace industry. Applications include a three-dimensional thin turning vane design, an analysis of a trailing edge device's effects upon a two-dimensional single-element symmetric airfoil, an examination of the effects of leading edge ice contamination on a single-element cambered airfoil, and an investigation of the presence of an unsteady response from rapid deployment of high-lift segments of a multi-element airfoil.

2 - Methodology

2.1 - Model Equations As A Subset of General Governing Equations

A solution is sought to a subset of the more general fluid mechanic conservation equations. The ensuing simulations in this work pertain to incompressible, viscous flows of Newtonian fluids which are calorically perfect, single-phase inert gases in thermodynamic equilibrium.

Starting from a more broad viewpoint, the continuum fluid flow equations for conservation of mass, momentum and energy are reproduced below from one of many
available references [51], excluding equations of state and transport models. These equations are given in compact shorthand (Einstein's summation notation with tensor analysis operators as described in [51]), in a Cartesian coordinate system:

\[ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \]  
Eq. (1)

\[ \frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = - \frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} + \rho f_i \]  
Eq. (2)

\[ \frac{\partial}{\partial t} \left( \rho \left( e + \frac{u_i u_i}{2} \right) \right) + \frac{\partial}{\partial x_j} \left( \rho u_j \left( h + \frac{u_i u_i}{2} \right) \right) = \frac{\partial}{\partial x_j} (u_i \tau_{ji}) - \frac{\partial q_j}{\partial x_j} \]  
Eq. (3)

By definition from one of many references on thermodynamic relationships [3], enthalpy and internal energy in equation (3) above are by definition further related by:

\[ h = e + \frac{P}{\rho} \]  
Eq. (4)

By excluding pressure-related terms, the remaining fluid stress terms are the deviatoric portion, which may be stated for a Newtonian fluid as follows [47]:

\[ \tau_{ij} = 2\mu S_{ij} + \lambda \delta_{ij} \frac{\partial u_k}{\partial x_k} \]  
Eq. (5)

\[ S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \]  
Eq. (6)

Stokes first proposed the following hypothesis for the viscosity coefficients of Newtonian fluids:

\[ \lambda = - \frac{2}{3} \mu \]  
Eq. (7)
This relation holds completely for a monatomic gas, and it is very commonly extended during numerical simulation to other Newtonian fluids in thermodynamic equilibrium with good accuracy [35, 51].

Assuming fluid incompressibility negates the need for accounting for energy conservation, so equation (3) is neglected. Additionally for an incompressible fluid, density does not vary spatially or temporally, so equations (1) and (2) may be reduced by treating it as a constant. Therefore, we are left with the viscous, incompressible equations of fluid mechanics for a Newtonian gas in thermodynamic equilibrium:

$$\frac{\partial u_i}{\partial x_i} = 0$$  \hspace{1cm} \text{Eq. (8)}

$$\rho \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j u_i) = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} + \rho f_i$$  \hspace{1cm} \text{Eq. (9)}

Finally, with equation (8) above and the symmetric property of fluid stresses for a Newtonian fluid, we can see that regardless of Stokes' closure of the second viscosity coefficient, the second term of equation (5) is zero under the incompressible assumption, leaving only the following simplified expression for deviatoric portions of the fluid stresses:

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$  \hspace{1cm} \text{Eq. (10)}
2.2 - Artificial Compressibility Method

The continuity equation provided by Eq. (8) does not contain a temporal derivative. Chorin [15] originally suggested adding a temporal derivative to Eq. (8) for steady-state problems, which would tend to zero as the solution ceased evolving.

\[
\frac{\partial \tilde{\rho}}{\partial t} + \frac{\partial \tilde{u}_i}{\partial x_i} = 0 \quad \text{Eq. (11)}
\]

This term essentially prevents decoupling from the velocity (advanced partially by the momentum equations but also strongly by the continuity equation) and the pressure (advanced solely by the simultaneous momentum equations, otherwise).

Chorin's work [15] suggests:

\[
\tilde{\rho} = \delta \tilde{P} \quad \text{Eq. (12)}
\]

Differentiating with respect to a pseudo-variable analogous to time:

\[
\frac{\partial \tilde{\rho}}{\partial \tilde{t}} = \delta \frac{\partial \tilde{P}}{\partial \tilde{t}} \quad \text{Eq. (13)}
\]

Aside, from the familiar relationships for a thermally perfect gas [3]:

\[
P = \rho RT \quad \text{Eq. (14)}
\]

\[
a = \sqrt{\gamma RT} \quad \text{Eq. (15)}
\]

Squaring both sides of Eq. (15) and dividing both sides by the specific heats ratio:

\[
\frac{a^2}{\gamma} = RT \quad \text{Eq. (16)}
\]

Then, plugging into Eq. (14) and solving for density:
\[ \rho = P \left( \frac{\gamma}{\alpha^2} \right) \]  

Eq. (17)

Now, examining Eq. (17) with respect to Eq. (12), it is observable that the following analogy may be made:

\[ \delta = \left( \frac{\gamma}{\alpha^2} \right) \]  

Eq. (18)

For the remainder of discussion, an alternative wave speed will be defined:

\[ \delta = \left( \frac{1}{\beta^2} \right) \]  

Eq. (19)

Accepting the pressure relation to artificial density as acceptable for the actual pressure (but only for the interim solution which is transient in pseudo-time), the modified continuity equation term becomes:

\[ \left( \frac{1}{\beta^2} \right) \frac{\partial P}{\partial \tilde{t}} \]  

Eq. (20)

The option of adding a corresponding pseudo-time term to the momentum equation [45] is exercised, such that a pseudo-time solution may be sought for it as well; this term is combined with the physical time derivative.

It may be shown for artificial compressibility methods in general [45] that while the equation system's convective eigenvalues are unmodified, the equation system's pressure wave speed eigenvalues are augmented by this alternative wave speed. In the limit that the alternative wave speed becomes large, the ad-hoc temporal term added to the continuity equation will decrease until it is effectively zero even during transient
evolution. Borrowing from Chorin’s description of the alternative wave speed as a relaxation parameter, this will cause the remaining equations to become increasingly stiff recreating the original problem of decoupling. In the limit that the alternative wave speed becomes too low, increasing amounts of numerical error are introduced.

Regardless, the steady state solution is not of exclusive interest. By sub-iterating through pseudo-time at a constant level of physical time, the pseudo-time term in the continuity equation vanishes. Therefore, the system of governing incompressible equations we are solving through pseudo-time is then:

\[
\left(\frac{1}{\beta^2}\right)\frac{\partial \tilde{P}}{\partial \tilde{t}} + \frac{\partial u_i}{\partial x_i} = 0 \tag*{Eq. (21)}
\]

\[
\rho \frac{\partial u_i}{\partial \tilde{t}} + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \tag*{Eq. (22)}
\]

2.3 - Vector Format of Model Equations & Cell-Centered Finite Volume Method

Construction from Differential Formulation

It is now more convenient to place the model equations in vector form. Expanding the Eq. (8) and each expression represented by Eq. (9) as they pertain to physical time:

\[
\frac{\partial (u)}{\partial x} + \frac{\partial (v)}{\partial y} + \frac{\partial (w)}{\partial z} = 0 \tag*{Eq. (23)}
\]

\[
\rho \frac{\partial (u)}{\partial \tilde{t}} + \frac{\partial (\rho uu)}{\partial x} + \frac{\partial P}{\partial x} \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial (\rho uv)}{\partial y} \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial (\rho uw)}{\partial z} \frac{\partial \tau_{xz}}{\partial z} = \rho(f_x) \tag*{Eq. (24)}
\]
\[
\rho \frac{\partial(v)}{\partial t} + \frac{\partial(\rho u v)}{\partial x} - \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial(\rho v v)}{\partial y} + \frac{\partial P}{\partial y} - \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial(\rho v w)}{\partial z} - \frac{\partial \tau_{yz}}{\partial z} = \rho f_y
\]
Eq. (25)

\[
\rho \frac{\partial(w)}{\partial t} + \frac{\partial(\rho w u)}{\partial x} - \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial(\rho v v)}{\partial y} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial(\rho w w)}{\partial z} - \frac{\partial \tau_{zz}}{\partial z} = \rho f_z
\]
Eq. (26)

Expanding the accompanying Eq. (10):

\[
\tau_{xx} = 2 \mu \frac{\partial u}{\partial x}
\]
Eq. (27)

\[
\tau_{xy} = \tau_{yx} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)
\]
Eq. (28)

\[
\tau_{xz} = \tau_{zx} = \mu \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)
\]
Eq. (29)

\[
\tau_{yy} = 2 \mu \frac{\partial v}{\partial y}
\]
Eq. (30)

\[
\tau_{yz} = \tau_{zy} = \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)
\]
Eq. (31)

\[
\tau_{zz} = 2 \mu \frac{\partial w}{\partial z}
\]
Eq. (32)

Similar derivative operands may be collected such that this may be expressed:

\[
\frac{\partial \tilde{U}}{\partial t} + \frac{\partial \tilde{F}_x}{\partial x} + \frac{\partial \tilde{F}_y}{\partial y} + \frac{\partial \tilde{F}_z}{\partial z} = \tilde{f}
\]
Eq. (33)

\[
\tilde{U} = \begin{bmatrix} 0 \\ \rho u \\ \rho v \\ \rho w \end{bmatrix}
\]
Eq. (34)
\[ \vec{F}_x = \begin{bmatrix} u \\ \rho uu + P - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \end{bmatrix} \]  
Eq. (35)

\[ \vec{F}_y = \begin{bmatrix} v \\ \rho vv - \tau_{xy} \\ \rho vw - \tau_{yz} \end{bmatrix} \]  
Eq. (36)

\[ \vec{F}_z = \begin{bmatrix} w \\ \rho ww - \tau_{xz} \\ \rho wz - \tau_{zz} \end{bmatrix} \]  
Eq. (37)

\[ \vec{f} = \begin{bmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho f_z \end{bmatrix} \]  
Eq. (38)

Briefly, the rows of terms given by Eq. (35), (36) and (37) must be put into a more compact form using the unit vectors for each Cartesian direction [45]:

\[ \vec{F} = \vec{F}_x \hat{i} + \vec{F}_y \hat{j} + \vec{F}_z \hat{k} = \langle \vec{F}_x, \vec{F}_y, \vec{F}_z \rangle \]  
Eq. (39)

Then, Eq. (33) becomes:

\[ \frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{F}}{\partial y} + \frac{\partial \vec{F}}{\partial z} = \vec{f} \]  
Eq. (40)

Recognizing the combination of spatial derivatives in Eq. (40) as the gradient operating upon the vector defined by Eq. (39), Eq. (40) may be rewritten as:

\[ \frac{\partial \vec{U}}{\partial t} + \nabla \cdot \vec{F} = \vec{f} \]  
Eq. (41)
Since the differential formulation discussed above holds for every point within the flow, it may be integrated over an arbitrary control volume. By integrating each of the terms of the differential form of the governing equations over the volume of each of these discrete computational cells, the integral form of the equations will remain. Integrating Eq. (33) over an arbitrary volume:

\[
\iiint_V \left( \frac{\partial \vec{U}}{\partial t} \right) dV + \iiint_V \left( \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{F}}{\partial y} + \frac{\partial \vec{F}}{\partial z} \right) dV = \iiint_V (\rho \vec{f}) dV
\]

Eq. (42)

The spatial derivatives in the second integral term may be transformed using the Green theorem (in two spatial dimensions) or the Gauss Divergence Theorem (in three spatial dimensions). This effectively replaces area integral terms or volume integral terms of the potentially discontinuously varying solution with path integral terms or surface integral terms of the fluxes of conserved variables in two or three dimensions, respectively [45]:

\[
\iiint_V (\nabla \cdot \vec{F}) dV = \iint_S (\vec{F} \cdot \vec{n}) dS
\]

Eq. (43)

Then, Eq. (42) becomes:

\[
\iiint_V \left( \frac{\partial \vec{U}}{\partial t} \right) dV + \iint_S (\vec{F} \cdot \vec{n}) dS = \iiint_V (\rho \vec{f}) dV
\]

Eq. (44)

Now, the equations may be discretized using the finite volume method. Considering a fluid domain which spatially stretches far from the region of interest, the domain may be broken up into a number of small, discrete computational cells (with n faces) for which an equivalent average value for the primitive solution variables may be
found. By assuming that some approximately exact values for solution variables may be
reconstructed at the boundaries of these computational cells for the surface integral terms
above, and that the lumped cell properties and volumetric forces are sufficient when
applying the volume integral terms above, Eq. (44) may be rewritten as a discretized
finite-volume formulation. The surface integrals are then interpreted as the net change in
conserved property fluxes across the cell boundaries, while the volume integrals are
interpreted as an accumulation or decrease in the conserved average properties contained
in the cells:

\[
\iiint_V \left( \frac{\partial \bar{U}}{\partial t} \right) dV + \iint_S (\bar{F} \cdot \bar{n}) dS - \iiint_V (\rho \bar{f}) dV \approx \frac{\partial}{\partial t} V \bar{U} + \sum_{k=1}^n \bar{F}_k \cdot \bar{n}_k A_k - \Psi \bar{f} \quad \text{Eq. (45)}
\]

Plugging in for the vector F from Eq. (39):

\[
\frac{\partial}{\partial t} V \bar{U} + \sum_{k=1}^n (\bar{F}_x \bar{i} + \bar{F}_y \bar{j} + \bar{F}_z \bar{k}) \cdot \bar{n}_k A_k = \Psi \bar{f} \quad \text{Eq. (46)}
\]

Here, the components of vector F are only denoted as vectors themselves to represent
rows of the equation system. Completing the dot product inside of the sum:

\[
\frac{\partial}{\partial t} V \bar{U} + \sum_{k=1}^n (\bar{F}_x n_x + \bar{F}_y n_y + \bar{F}_z n_z) A_k = \Psi \bar{f} \quad \text{Eq. (47)}
\]

Eq. (47) above holds for two-dimensional problems by setting fluxes to zero in the third
dimension, and using constant lengths in the third dimension; volumes apply on a “per
dept” basis (as cell area).
2.4 – Interface Flux Vector Splitting, Upwinding, and High-Order Extension

The fluxes from a given interface $k$ (formulated as the parenthesis from one term of the sum in Eq. (47)) may be rewritten as the following by recalling Eq. (35) through Eq. (37), replacing density back into the continuity equation, and pulling density and the velocity component normal to the cell interface out of the row vectors:

$$\left(\vec{F}_x n_x + \vec{F}_y n_y + \vec{F}_z n_z\right)_k = \rho \begin{bmatrix} u \\ v \\ w \end{bmatrix} + P \begin{bmatrix} 0 \\ n_x \\ n_y \\ n_z \end{bmatrix} - \begin{bmatrix} 0 \\ \tau_{xx} + \tau_{xy} + \tau_{xz} \\ \tau_{yx} + \tau_{yy} + \tau_{yz} \\ \tau_{zx} + \tau_{zy} + \tau_{zz} \end{bmatrix}_k$$

Eq. (48)

Here, the total cell interface flux is divided into convective, pressure and viscous contributions respectively; the sum of the convective and pressure contributions form the total inviscid contribution. Upwinding formulations modify the inviscid terms, and split them into a biased sum of portions originating from either side of the interface.

Standard central difference discretizations for most of the spatial derivative terms lack adequate dissipation to smooth out the discretizations' inherent instabilities [45]; upwinding schemes force information to propagate through the flow field in a manner consistent with fluid physics by accounting for effects such as flow direction and Mach number, introducing an additional amount of dissipation due to their formulation. Although most upwinding schemes are not rigorously derived, they can still be shown to exactly capture various types of discontinuities.

A simplified, incompressible-capable version [10] of Edwards' Low-Diffusion
Flux Splitting Scheme (LDFSS) [19] provides upwinding and the framework for flux vector splitting in this work. LDFSS is based on a modification to Van Leer’s flux splitting scheme; it retains monotonic capturing (including moving and non-grid aligned discontinuities), but is distinguished from many schemes by its ability to exactly capture a stationary contact wave. LDFSS exhibits the telescoping property and disallows expansion-shock type solutions. Using LDFSS, the split analogue for Eq. (48) applied to the X-direction flux at the “i+1/2” interface (between cell “i” and cell “i+1”) is:

$$\vec{F}_{i+1/2} = \vec{F}_{i+1/2}^+ + \vec{F}_{i+1/2}^-$$  \hspace{1cm} \text{Eq. (49)}$$

$$\vec{F}_{i+1/2}^+ = \rho a^{-1}_{i+1/2} \begin{pmatrix} 1 \\ u \\ v \\ w \end{pmatrix} + C_{E,c}^- \begin{pmatrix} 1 \\ u \\ v \\ w \end{pmatrix}$$  \hspace{1cm} \text{Eq. (50)}$$

$$\vec{F}_{i+1/2}^- = C_{E,p} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$  \hspace{1cm} \text{Eq. (51)}$$

$$\vec{F}_{i+1/2}^\tau = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \end{pmatrix}$$  \hspace{1cm} \text{Eq. (52)}$$

Defining convective reference speed as a maximum flow velocity magnitude as the scalar reference speed of the flow, limited by a unity Courant number condition, and pressure reference speed as simply the maximum flow velocity magnitude:
\[ V_{\text{ref},e} \equiv \max \left( V_{ijk} \right) \min \left( \frac{\Delta x_{ijk}, \Delta y_{ijk}, \Delta z_{ijk}}{\Delta t} \right) \forall i, j, k \]  
Eq. (53)

\[ V_{\text{ref},p} \equiv \max \left( V_{ijk} \right) \forall i, j, k \]  
Eq. (54)

When determining fluxes along interfaces oriented normal to a given computational direction, the formulation requires adopting subscripts “L” for the “left” (or “minus”) side of that interface, and the “R” for the “right” (or “plus”) side of that same cell interface. To clarify, for cell interface “i + 1/2”:

\[ u_L \equiv u_i \]  
Eq. (55)

\[ u_R \equiv u_{i+1} \]  
Eq. (56)

The same left and right conventions apply for interfaces dividing cells in any of the computational directions (i, j or k), and for any of the velocity components (u, v or w), reference speeds or pressure (P).

Then, several interface properties are defined as simply the average of the cell properties in the cells adjacent to the interface:

\[ u_{i} \equiv \frac{1}{2} (u_L + u_R) \]  
Eq. (57)

Interface properties for all three velocity components (u, v or w), both reference speeds and pressure (P) are calculated for interfaces dividing cells in each computational direction (i, j, or k).

Using interface-normal velocity components:

\[ u_{L,R} \equiv \left( (u_{L,R})_x n_x + (v_{L,R})_y n_y + (w_{L,R})_z n_z \right) \]  
Eq. (58)
\[ \vec{u}_i \equiv \frac{1}{2} (\vec{u}_L + \vec{u}_R) \]  
Eq. (59)

Although the incompressible assumption of the flow solver implies an infinite sound speed, an effective numerical interface sound speed may be calculated:

\[ a_{i+\frac{1}{2}}^\sim \equiv \sqrt{\left( \left( \frac{\vec{u}_{i+\frac{1}{2}}^2}{2} \right)^2 + 4 \left( V_{ref,c} \right)^2 \right)} \]  
Eq. (60)

Then, left and right Mach numbers may be defined, in addition to two more constants:

\[ M_{L,R} \equiv \frac{\vec{u}_{L,R}}{a_{i+\frac{1}{2}}} \]  
Eq. (60)

\[ \alpha_{L,R}^\pm \equiv \frac{1}{2} \left( 1 \pm \text{sign} \left( 1, M_{L,R} \right) \right) \]  
Eq. (62)

\[ D^\pm \equiv \frac{1}{2} \left( 1 \pm M_{L,R} \right) \]  
Eq. (63)

Van Leer’s flux splitting terms can now be determined, in addition to an interface Mach number and split interface Mach numbers:

\[ C_{VL}^\pm \equiv \pm \left( M_{L,R} \pm 1 \right)^2 \]  
Eq. (64)

\[ M_{\frac{1}{2}} = \frac{1}{2} \left( C_{VL}^+ M_L - C_{VL}^- M_R \right) \]  
Eq. (65)

\[ M_{\frac{1}{2}}^+ = M_{\frac{1}{2}} \left( 1 \mp \frac{1}{2} \left( \frac{P_L - P_R}{\rho V_{ref,c}^2} \right) \right) \]  
Eq. (66)

Now, the convective coefficient for Edwards flux splitting can be defined. An additional coefficient is also defined here to combine all of the subsonic pressure splitting terms for
the Edwards scheme into a single term:

\[
C_{E,c}^\pm \equiv C_{VL}^\pm M_1^\pm \quad \text{Eq. (67)}
\]

\[
C_{E,p}^\pm \equiv \frac{1}{2} \left( (P_L + P_R) + |D^+ - D^-| (P_L - P_R) \right) + \rho V_{ref,p}^2 \left( D^+ + D^- - 1 \right) \quad \text{Eq. (68)}
\]

Viscous fluxes are already second-order, since the flux difference occurs between stresses which are already determined using first-order differencing; since viscosity's primary effect is diffusive, no upwinding is necessary. The remaining fluxes must be extended to high-order accuracy by way of determining in each computational direction a nonlinear approximation to flow variation using an increasing number of points, over which the averages are the cell lump values. The Total Variation Diminishing (TVD) [25] and Weighted Essentially Non-Oscillatory (WENO) [31] higher-order extensions are both implemented in various versions of the methods used in this work.

Where an interpolation polynomial approach is used, the Gibbs phenomenon [23] is seen near discontinuities as a consequence of using continuous functions to approximate them. To either side of the discontinuity, overshoot occurs beyond the exact local property extrema, followed by undershoot moving away from the discontinuity, which continues to develop into growing oscillations.

Coefficients may be introduced in front of spatial gradient terms in the flux formulation properties which prevent these oscillations from growing by restricting unrealistically large contributions near discontinuities; these coefficients themselves are functions of the property slopes adjacent to an interface. Defining simple measures of the
ratio of unidirectional slopes in an arbitrary property between two adjacent cells in the positive and negative computational directions:

\[ r^+_{i+\frac{1}{2}} = \frac{u_{i+2} - u_{i+1}}{u_{i+1} - u_i} \quad \text{Eq. (69)} \]

\[ r^-_{i+\frac{1}{2}} = \frac{u_i - u_{i-1}}{u_{i+1} - u_i} \quad \text{Eq. (70)} \]

A set of nonlinear slope-limiting functions of these adjacent property slope ratios may then be defined such that oscillations either remain constant or decay, but do not grow [48]. This is paraphrased by:

\[ \Psi(r) = \begin{cases} 0 & r < 0 \\ \max \{0, \min \{r, 1\} \} & r \geq 0 \end{cases} \quad \text{Eq. (71)} \]

The TVD approach taken in this work uses the Minmod and Van Leer limiters [37]. The former is the most dissipative choice, while the latter is moderately more compressive:

\[ \Psi_{mm}(r) = \max \{0, \min \{r, 1\} \} \quad \text{Eq. (72)} \]

\[ \Psi_{vl}(r) = \begin{cases} \frac{r + |r|}{1 + r} & r \geq 0 \\ 0 & r < 0 \end{cases} \quad \text{Eq. (73)} \]

Given the slopes “a” and “b” of a property at adjacent cells:

\[ \Psi_{mm}(a, b) = \begin{cases} \frac{1}{2} \left( \text{sign}(1, a) + \text{sign}(1, b) \right) \min \{|a|, |b|\} & ab > 0 \\ 0 & ab \leq 0 \end{cases} \quad \text{Eq. (74)} \]

\[ \Psi_{vl}(a, b) = \begin{cases} \frac{1}{2} \left( \text{sign}(1, a) + \text{sign}(1, b) \right) \min \left( \frac{1}{2} |a+b|, 2 |a|, 2 |b| \right) & ab > 0 \\ 0 & ab \leq 0 \end{cases} \quad \text{Eq. (75)} \]
It has been shown [24] that applying TVD schemes in multidimensional problems results in a reduction in the order of accuracy across the entire computational domain; however, it seems generally accepted that in practice, the accuracy reduction occurs locally near discontinuities and boundaries. As limiters inherently inhibit solution evolution, TVD schemes may also experience slower convergence, and worse achievable convergence due to stabilized “bouncing” limiter behavior. Certain limiters may be “frozen” mid-calculation to slightly increase achievable steady-state convergence.

The alternative WENO [31] approach taken in this work to higher-order accuracy extension utilizes a convex combination of polynomials of various orders to approximate the interface properties between adjacent cells. This is based on the earlier ENO limiter [26] in which simply the smoothest available polynomial stencil is chosen. By evaluating different stencil sets involving the candidate interface, and automatically adapting the combination of stencils to maximize smoothness, growth of oscillations is minimized in transient solution evolution, and damping in the steady-state is strongly encouraged. Third-order and fifth-order polynomials are utilized in this work and applied unidimensionally. Further details of the WENO approach in multiple dimensions are available [40], as well as the specifics taken in this work [34].

2.5 - Immersed Boundary Method As Boundary Condition

The first requirement for this work's immersed boundary method is to divide the
immersing grid nodes into the following regions:

a) Points fully outside and away from the surface geometries

b) Points fully inside the surface geometries

c) Points bounding the surface geometries

Multiple methods were tested (below), and some continue to be refined [16]. Thick surfaces are ideally represented by consistently ordered (e.g. clockwise or counterclockwise) triangulated surfaces in three dimensions or line-segment curves in two dimensions. Thin surfaces have no such order restrictions.

The first method involved counting immersed surface intersections along a rigorous ray trace from a randomly generated point far outside the computational domain. The random point is easily generated by examining extents and size of the immersing grid. By counting the intersections between the far point and a point in question, it may be placed relative to a triangulated surface geometry point cloud without even requiring supplied normal vectors. Intersections are checked by brute force, taking the ray-plane intersection between the far point and point in question, and a triangular panel. The planar intersection point is then checked for proximity (and more stringently, inclusion) on the actual surface panel by projecting panel corner points and intersection point onto the Cartesian direction plane most closely aligned with the panel normal vector. This effectively reduces the dimension of the problem, allowing a two-dimensional then one-dimensional analogue of the original three-dimensional point placement to occur; or, the two-dimensional problem may be solved easily in a direct manner by injecting the
projected intersection point into the nearest projected panel side, and examining the sign of projected panel area change. For an “external”-type immersed boundary flow problem, an even number of intersections would imply an external point, and an odd number of intersections would imply an internal point (vice versa for an “internal-type” immersed boundary flow problem).

Although this first method exhibits good parallel scalability, it proved to be expensive, and machine precision, the random nature of the calculation, and miscounted intersections near panel corners, edges and sharp geometric edges were problematic. Assuming the number of intersections is known between a far point and point in question, it may be briefly expressed mathematically as follows, without delving further into the method for checking for intersections:

\[
\text{Raytrace} = \langle x_{pt} - x_{far}, y_{pt} - y_{far}, z_{pt} - z_{far} \rangle
\]

Eq. (76)

\[
\text{Point is } \begin{cases} \text{Same Side} & \text{mod} \left( n_{\text{ints}}, 2 \right) = 0 \\ \text{Other Side} & \text{mod} \left( n_{\text{ints}}, 2 \right) = 1 \end{cases}
\]

Eq. (77)

Consideration for the first method was not extended to multiple bodies; however, for multiple bodies, the rule provided by Eq. (77) would theoretically apply to the sum of intersections for all bodies. The first method could likely be significantly improved upon with, for example, particular choices of the arbitrary external point (which was randomly generated for this work).

The second method involved a simple sign check on the dot product between a vector from the point in question to the nearest surface point, and that nearest surface
point’s outward normal vector. The Approximate Nearest Neighbor (ANN) library [5] provided an efficient black-box capability to calculate the nearest surface point (and subsequent closest surface points) within a specified level of accuracy (ranging from approximate to exact), in many more than the sufficient number of dimensions, by using tree-sort algorithms to provide efficient search methods.

Although abrupt variation in adjacent surface normal vector directions still can prove potentially problematic for this nearest-neighbor type method, nearest-group type methods reduce and nearly eliminate these problems. The second method also exhibits good parallel scalability. The simplest version of the second method can be expressed mathematically as follows, assuming the nearest point on each surface “i” and its normal vector are known. Taking a dot product between the nearest surface point’s normal vector, and a vector connecting the point and its nearest surface point:

\[ d_{direction,i} = \vec{V}_{pt-nearest,i} \cdot \vec{n}_{outward,nearest,i} \]  
Eq. (78)

Point is \( \begin{cases} \text{Outside} & d_{direction} > 0 \\ \text{Inside} & d_{direction} < 0 \end{cases} \)  
Eq. (79)

For three-dimensional problems, the second method was deemed clearly superior based on a subjective valuation of accuracy and efficiency. For two-dimensional problems, the methods are judged equal by the same subjective valuation. Both methods benefit from “bounding box elimination” and other common-sense logical caveats.

Regardless, a signed distance function is created which is the minimum value over all immersed surfaces for the magnitude of the relative position vector in Eq. (78):
\[
\phi = \min(\left| \vec{V}_{pt-nearest, i} \right| \text{SIGN}(d_{direction, i})) \forall i \tag{80}
\]

This represents the local distance from the nearest surface, and is positive for external points. The collection of points at which the signed distance function is negative form the interior of the immersed boundary, and are differentiated as interior points.

Points lying on the surface itself (where signed distance function is zero) are classified as band points. Band points also include both 1) points adjacent to an interior point in any computational direction, and 2) points diagonal to an interior point in any combination of computational directions.

A heaviside function of the signed distance function is also defined which forms an on/off, flag-like representation of the fluid mechanic treatment of the location in this work; this flag is “off” where the governing equations are solved, and “on” where immersed boundary method is utilized. The clearest way to create the heaviside function is by examining whether a given point falls within a set of points \( \Omega_{out,far} \) which are outside of the immersed surface, and which are not band points.

\[
G(\phi_i) = \begin{cases} 
0 & x_i \in \Omega_{out,far} \\
1 & x_i \notin \Omega_{out,far}
\end{cases} \tag{81}
\]

Interpolation procedures are performed for fluid properties at band points near the immersed surface. These interpolations are performed along a local coordinate stretching from a band point's nearest surface neighbor, and along a band point's nearest surface neighbor's outward normal vector. The bounds of the interpolation are the band point's nearest neighbor on the surface, and a point defined below at a given distance \( d_i \).
away from the nearest surface neighbor along its outward normal vector.

Properties at an off-body interpolation point are reconstructed from surrounding field points using a normalized weighting procedure. The set of all possible surrounding points are chosen using a expanding stencil; by carefully ordering the stencil, it is possible to toggle the inclusion of immersing grid points in various dimensions and at various computational lengths, all by adjusting an integer controlling stencil size. Figure 2.5.1 below illustrates various stencil sizes and shapes controlled by this integer in two dimensions, for simplicity.

![Expanding Off-Body Interpolation Stencil](image)

**Figure 2.5.1 – Expanding Off-Body Interpolation Stencil**

Defining an unsigned distance from the band point $x_b$ to a point in the set of all possible stencil points $x_i$, and a measure of the coincidence of the vector between those points with the band point $x_b$'s nearest neighbor $x_n$'s outward normal vector:

$$|d_{l-b}| = |\vec{V}_{xl-xb}|$$  \hspace{1cm} Eq. (82)

$$d_{l-b,n} = \vec{V}_{xl-xb} \vec{n}_n$$  \hspace{1cm} Eq. (83)
Then, from all of the possible stencil points \( x_i \) surrounding the band point \( x_b \) with nearest neighbor \( x_n \), a subset \( x_m \) is selected which is comprised of points in the normal direction from the surface interpolation point:

\[
x_i \begin{cases} 
    \in x_m & d_{i-b,n} < 0 \\
    \notin x_m & d_{i-b,n} \geq 0
\end{cases}
\]

Eq. (84)

Maintaining the same subset of selected stencil points, the properties from this subset of points are weighted using the following function, which is self-normalized during an intermediate calculation step. This procedure yields higher weights to points closer to the band point and normal coordinate axis:

\[
\forall m : x_m, W_m \equiv \left( 10^{-12} + \sqrt{d_{i-b}^2 - d_{i-b,n}^2} \right)^2
\]

Eq. (85)

\[
\forall m : x_m, \tilde{W}_m \equiv \frac{W_m}{\sum_{1}^{m} W_m}
\]

Eq. (86)

Then, for an arbitrary property at a weighted median location of arbitrarily selected points:

\[
\bar{q} \bigg|_{d_i} = \sum_{j=1}^{m} \tilde{W}_j q_j
\]

Eq. (87)

\[
\frac{d \bar{q}}{d n} \bigg|_{d_i} = \sum_{j=1}^{m} \tilde{W}_j \left( \frac{d q}{d n} \right)_j \vec{n}_j
\]

Eq. (88)

The interpolated surface-tangential component of velocity at the band point (relative to the surface velocity) is governed by a power law with arbitrary power, which follows the form:
\[ u_{T,j}^-(n) = u_{T,j}^-(d_i) \left( \frac{n}{d_i} \right)^k \]  

Eq. (89)

This may be expanded about the interpolated point using a Taylor-series expression and product rule for differentiation, then subsequently rearranged to the following form:

\[
\bar{u}_{b,T,j} = \left( \frac{\phi_b}{d_i} \right)^k \bar{u}_{\text{interp},T,j} + \left( 1 - \frac{\phi_b}{d_i} \right) \left( k \bar{u}_{\text{interp},T,j} + \phi_b \frac{d \bar{u}_{\text{interp},T,j}}{dn} \right)
\]  

Eq. (90)

In the above expression, the following must be evaluated at the interpolation point:

\[
\bar{u}_{\text{interp},T,j} = (\bar{u}_{\text{interp},T,j} - \bar{u}_{\text{surf},T,j}) - n_j (\bar{u}_{\text{interp},j} - \bar{u}_{\text{surf},j}) n_j
\]  

Eq. (91)

Also, the dot product of the surface normal vector and the velocity gradient at the interpolation point are used to approximate the change:

\[
\frac{d(\bar{u}_{\text{interp},T,j})}{dn} \bigg|_{n=\phi_b} = \frac{\partial \bar{u}_{\text{interp},T,j}}{\partial x_j} n_j - \left[ \left( \frac{\partial \bar{u}_{\text{interp},k}}{\partial x_j} n_j \right) n_k \right] n_l
\]  

Eq. (92)

Repeated indices represent a summation.

An accompanying cubic interpolation for the normal velocity component (relative to the surface) is as follows:

\[
\bar{u}_{N,j}^- = \bar{u}_{N,j}^-(d_i) \left( \frac{n}{d_i} \right)^2
\]  

Eq. (93)

It is found similarly, and may or may not be included in the formulation. The cubic power is chosen to enforce a zero second derivative boundary condition for momentum conservation [21], in addition to the known interpolation point velocity and non-slip (or a known surface velocity) surface condition. It results in [16]:

26
\[
\tilde{u}_{b,N,l} = \left( \frac{\phi_b}{d_i} \right) \left( \bar{u}_{\text{interp},N,l} \right) + \frac{1}{2} \left( 1 - \left( \frac{\phi_b}{d_i} \right)^2 \right) \left( \bar{u}_{\text{interp},N,l} \phi_b \frac{d \tilde{u}_{\text{interp},N,l}}{dn} \bigg|_{n=\phi_b} \right)
\]
Eq. (94)

\[
\tilde{u}_{\text{interp},N,l} - \bar{n}_l \left( \tilde{u}_{\text{interp},j} - \bar{u}_{\text{surf},j} \right) \bar{n}_j
\]
Eq. (95)

\[
\frac{d(\tilde{u}_{\text{interp},N,l})}{dn} \bigg|_{n=\phi_b} = \left[ \frac{\partial \tilde{u}_{\text{interp},k}}{\partial x_j} \bar{n}_j \right] \bar{n}_k \bar{n}_l
\]
Eq. (96)

A pressure interpolation is also formulated based on the assumption of a quadratic polynomial describing the pressure deviation from its freestream value:

\[
P_{b,l} - P_{\infty} = A + Bn + Cn^2 \tag{97}
\]

Evaluating this expression at the wall shows that the pressure deviation from freestream at the wall is equal to the first constant ("A"), and that the slope of this pressure deviation from freestream at the wall is equal to the second constant ("B"). By selecting a value for \(B\), which varies in proportion to the surface's acceleration in the normal direction, considering Eq. (97) and its derivative at an arbitrary location yields two remaining equations and two remaining unknowns. Solving yields the following at a band point:

\[
P_{b,l} \bigg|_{n=\phi_b} = P_{b,n=d} + b n - \left( \frac{\phi_b}{d_i} \right)^2 \left[ \frac{d_i}{2} \left( b + \frac{dP}{dn} \bigg|_{n=d} \right) \right]
\]
Eq. (98)

Figure 2.5.2 below illustrates some first-order, two-dimensional velocity profiles for various power law values.
Schlichting reported [51] that Prandtl first suggested a power law value of 1/7 for high Reynolds number flat plate boundary layers. Far higher power law values up to a proportional relationship at unity are appropriate for laminar flow regions at low Reynolds numbers, although these may create massive flow separation at typically benign freestream conditions [16]. Values between 1/7 and 1/9 seem to produce reasonable results at high Reynolds numbers.

Illustrations demonstrating the division of the immersing grid node set, calculation of the signed distance function calculation, more advanced point placement procedures, and interpolation procedures are available in related publications [16].
2.6 - Extension of Immersed Boundary Method to Zero-Thickness Thin-Baffles

To extend the existing methodology discussed above in this work to infinitely thin immersed boundary surfaces, only a few modifications are necessary.

First, all points are external to the thin surface, and a new band width criteria must be introduced to define band points adjacent to the immersed boundary. Points at which this band width exceeds the absolute magnitude of the signed distance function are thin-baffle band points. Mathematically, for an unsigned version of the value calculated by Eq. (80):

\[ \phi_{\text{thin}} = |\phi_{\text{thick}}| \]  

Eq. (99)

Then:

\[ \left\{ \begin{array}{l}
    x_i \in \Omega_{\text{out, far}} \quad \phi_{\text{thin}} > w_{\text{band}} \\
    x_i \notin \Omega_{\text{out, far}} \quad \phi_{\text{thin}} \leq w_{\text{band}}
\end{array} \right. \]  

Eq. (100)

There is then no change to heaviside calculation. A band width just above one computational unit in the surface-normal direction (which is slightly more than the physical size of a local cell in the direction orthogonal to the surface) guarantees similar partitioning to the original, finite-thickness immersed boundary method in this work. This was determined locally using the following algorithm for a given cell:

\[ w_{\text{band}} = \frac{V}{\min(A_j)} = \max(L_j) \]  

Eq. (101)

Above, the numerator is the cell volume, and the denominator is a measure of the
maximum cell face area, resulting in a maximum dimension of cell length.

The second modification is that properties at the off-body interpolation point must be comprised only of points on the same side of the surface. Replacing Eq. (84), which defines the subset of all possible points to reconstruct the interpolation point with:

\[
\begin{align*}
  x_l \in x_m & \quad \phi_{thick} d_{i-b,n} \geq 0 \\
  x_l \not\in x_m & \quad \phi_{thick} d_{i-b,n} < 0
\end{align*}
\]

Eq. (102)

This is the only place in the thin-baffle formulation where the original signed distance function is used.

Finally, for monitoring the solution, summing forces and moments, and viewing output, it becomes necessary to store values extrapolated to the surface from each side of a thin interface.

2.7 - Solution Advancement Through Pseudo-Time and Time

The combined field-band solutions to the finite volume statement of the problem are advanced between physical time levels through a number of pseudo-time subiterations. After sufficient subiteration between time levels, theoretically 1) the field system's Artificial Compressibility (AC) terms, 2) the field system's continuity divergence, and 3) subiteration changes to the band system's forcing function terms will all approach zero, leaving an appropriate physically-based, velocity-divergence-free solution with prescribed fluid band velocities and pressures near the surfaces. At this point, the solution is considered satisfactorily advanced to become the next physical time.
step's solution. In practice, this procedure is robust and consistent enough to simply prescribe a low, constant integer number of these subiterations. For unsteady calculations, five subiterations are used in this work, while only two are used for steady-state solutions (where time accuracy and intermittent solution quality are not important).

The following nomenclature is adopted for discussion of the solution advancement. The previous physical timestep is n-1, the current physical timestep (which has already been calculated) is n, and the next physical timestep (which is being calculated) is n+1; the physical timestep varies from zero (which is defined by explicit definition in an input file) to nmax. For the next physical timestep (which is again in the process of being calculated), subiteration level k varies from 1 (which is identical to the current physical timestep which has already been calculated) to kmax (which is identical to the next physical timestep which is currently being calculated).

Applying the following backwards difference operator to the time derivative contained in the momentum equations at subiteration k and for the upcoming time level n+1, and choosing to track the following primitive variables:

\[
\frac{3}{2} \frac{U_{k,n+1} - 4U^n + U_{n-1}}{\Delta t} + O(\Delta t^2)
\]

Eq. (103)

\[
\vec{B} = \begin{pmatrix} p \\ u \\ v \\ w \end{pmatrix}
\]

Eq. (104)

Additionally, creating a new vector from the difference between the current pressure and velocity of band points, and the target interpolated band pressure and velocity:
The pseudo-time residuals of the continuity and momentum equations can then be fully defined for the combined field and band system. The field values for the residuals are composed by dividing the continuity fluxes and lumped momentum fluxes & body force terms in Eq. (47) by the cell volume, and adding the physical time difference term from Eq. (103). Using the heaviside function of Eq. (81) to toggle between the two regions, and accounting for the interpolated band pressure & velocity values:

\[
R^{n+1,k} = (1 - G(\phi^{n+1})) \left( \frac{3 U^{n+1,k} - 4 U^n + U^{n-1}}{2 \Delta t} + \left( \sum_{l=1}^{n} \left( \overrightarrow{F}_x n_x + \overrightarrow{F}_y n_y + \overrightarrow{F}_z n_z \right) - f \right) \right)
\]

\[
+ G(\phi^{n+1}) D^{n+1,k}
\]

Eq. (106)

Driving the system's evolution pseudo-time increment as a function of this combined residual, a classic linear algebra system of type \( Ax = B \) is created:

\[
\tilde{A} \left( B^{n+1,k+1} - B^{n+1,k} \right) = \tilde{A} \left( \Delta B^{n+1,k+1} \right) = -R^{n+1,k}
\]

Eq. (107)

For stability, it is desirable to frame the integration method in an implicit manner. The increased stability of an implicit method originates from the nonlinear nature of solution
changes during convergence. Additional terms are included to account for the solution change vector's differences at the future time level. Subsequently, the above system Jacobian matrix is inclusive of a measure of sensitivity of the residual vector due to the impending change in solution vector:

\[
\tilde{A} \approx A + \frac{\partial \tilde{R}}{\partial B}
\]

Eq. (108)

This may still be considered semi-rigorous due to the fact that the residual at the current time level, given by the right hand side of Eq. (106), may be thought to approximately differ from the next time level's residual by the product of said sensitivity and the impending solution vector change.

The resulting system is approximately solved for each subiteration by performing an Incomplete Lower-Upper (ILU) triangular decomposition of the system Jacobian matrix, and relaxing band properties from the current values towards those prescribed at the next timestep. This is tantamount to using a preconditioned iterative method to invert the implicit system Jacobian matrix (as opposed to direct inversion); doing so increases the efficiency of the method, and is discussed more thoroughly in related works [34].

2.8 – Edwards Variant of Spalart-Allmaras Turbulence Model

The Boussenesq assumption posits that fluctuations in turbulent eddies and molecules are similar enough such that the stresses, heat fluxes and related transport properties which result from laminar and turbulent motion may be formulated in a similar
manner [51]; this implies that the bulk viscosity coefficient could be broken down into each a well-understood laminar component, and a similar, corresponding turbulent component which is not as well understood. Due to their chaotic nature, turbulent transport properties must be modeled to achieve closure.

The one-equation turbulence model of Spalart and Allmaras is an appropriate choice for this work, given that it requires distance from the closest surface (which corresponds closely to the distance function in the immersed boundary method which must already be determined).

Defining the strain rate tensor [20], implicit Spalart-Allmaras kinematic eddy viscosity parameter [44], and surface-truncated scalar distance function:

\[
\vec{S} \equiv \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \left( \frac{\partial u_k}{\partial x_k} \right)^2 \quad \text{Eq. (109)}
\]

\[
\nu_T \equiv \bar{\nu} f_v \quad \text{Eq. (110)}
\]

\[
d \equiv \text{max} \left( 10^{-6}, \phi \right) \quad \text{Eq. (111)}
\]

The following transport equation defines the Spalart-Allmaras kinematic eddy viscosity [51]:

\[
\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{bl} \vec{S} \tilde{\nu} - c_{wft} f_w \left( \frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \frac{\partial}{\partial x_k} \left( \nu + \tilde{\nu} \right) \frac{\partial \tilde{\nu}}{\partial x_k} - \frac{c_{b2}}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_k} \frac{\partial \tilde{\nu}}{\partial x_k} \quad \text{Eq. (112)}
\]

The following relationships complete the Edwards variant (SA-Edwards) of the Spalart-Allmaras formulation used in this work [20]:

\[
\chi = \frac{\tilde{\nu}}{\nu} \quad \text{Eq. (113)}
\]
\[ f_{v1} = \frac{X^3}{X^3 + c_{v1}^3} \]  \hspace{1cm} \text{Eq. (114)}

\[ f_{v2} = 1 - \frac{X}{1 + X f_{v1}} \]  \hspace{1cm} \text{Eq. (115)}

\[ \tilde{S} = \sqrt{S} \left( \frac{1}{X} + f_{v1} \right) \]  \hspace{1cm} \text{Eq. (116)}

\[ r = \frac{\tanh \left( \frac{\tilde{v}}{\sqrt{\frac{S}{\kappa^2 d^2}}} \right)}{\tanh(1)} \]  \hspace{1cm} \text{Eq. (117)}

\[ g = r + c_{w2} \left( r^6 - r \right) \]  \hspace{1cm} \text{Eq. (118)}

\[ f_w = g \left( \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6} \]  \hspace{1cm} \text{Eq. (119)}

The Kármán and other constants ensuing provide closure for the Spalart-Allmaras model:

\[ \kappa = 0.41 \]  \hspace{1cm} \text{Eq. (120)}

\[ \sigma = \frac{2}{3} \]  \hspace{1cm} \text{Eq. (121)}

\[ c_{b1} = 0.1355 \]  \hspace{1cm} \text{Eq. (122)}

\[ c_{b2} = 0.622 \]  \hspace{1cm} \text{Eq. (123)}

\[ c_{v1} = 7.1 \]  \hspace{1cm} \text{Eq. (124)}

\[ c_{w2} = 0.3 \]  \hspace{1cm} \text{Eq. (125)}

\[ c_{w3} = 2 \]  \hspace{1cm} \text{Eq. (126)}
The original form of the Spalart-Allmaras turbulence model provides a relatively robust one-equation transport model for eddy viscosity by equating its time rate of change and convection on the left hand side of Eq. (112) with terms for production, destruction and diffusion on the right hand side, respectively. The Edwards variant enhances convergence behavior by improving numerical performance adjacent to surfaces; the correction is accomplished mainly by the modified term defined in Eq (119) using the modified Eq. (116), (117) and (118). The original terms may be found in [44]. This comes at a cost of requiring a nonzero kinematic eddy viscosity parameter.

2.9 - Force and Moment Calculation

The net force acting upon the surface can be represented diametrically as either the fluid acting upon the surface, or the surface's action upon the fluid. The representation mentioned prior is calculated using an integral of shear and pressure components over the surface; the representation mentioned latter is calculated as a surface integral of the momentum deficit on an arbitrary control volume of fluid surrounding the surface (and by extension through the Gauss theorem, a volume integral of the arbitrary control volume may be substituted for the surface integral of the fluid's momentum deficit). These different viewpoints are mathematically equivalent, as formulated below; each is implemented in the current work.
The different approaches each imply merits and drawbacks, including “uniqueness,” an impact to accuracy [13], and the abilities for instantaneous measurements, measurements of various specific subcomponents of the forces and geometry, and calculation of moments. Here, the description of “uniqueness” is intended to convey formulation for which there is no need for any arbitrary inputs.

Surface pressure and shear integrations are widely used. They are capable of providing both force and moment output on a fully unique and instantaneous basis, and also separating force and moment contributions from pressure and viscous mechanisms for various portions of an overall geometry, but they may require corrections for high accuracy (such as for farfield circulation).

From a solid surface integration perspective, with a fully populated surface solution, the resulting forces and moments acting upon the solid surface may be calculated. Starting with two arbitrary vectors fully contained in an element of surface area:

\[ \left( \mathbf{V}_{t,a}, \mathbf{V}_{t,b} \right) \parallel dS \]  
\[ \text{Eq. (128)} \]

For an element of surface area, defining an outward surface normal vector along with a surface flow-tangential vectors (which are simply a unit vector in the direction of the flow in the limit of distance to the surface approaching zero along the surface normal vector):

\[ \mathbf{n}_\perp dS \equiv \frac{\mathbf{V}_{t,a} \times \mathbf{V}_{t,b}}{|\mathbf{V}_{t,a} \times \mathbf{V}_{t,b}|} \]  
\[ \text{Eq. (129)} \]
Additionally, it becomes necessary to define an effectivity vector which describes the direction along which an element of force must be aligned to maximize its moment contribution for a given hinge and moment arm vector. Noting that each moment vector component will utilize a different hinge vector, the effectivity vectors for each moment contribution due to a given element of surface area are then chosen as:

\[
\vec{r}_{\text{eff}, x} \equiv \left( \frac{\vec{r}_{\text{arm}, i} \times \vec{i}}{|\vec{r}_{\text{arm}, i} \times \vec{i}|} \right) \times \vec{r}_{\text{arm}, i} \\
\vec{r}_{\text{eff}, y} \equiv \left( \frac{\vec{r}_{\text{arm}, i} \times \vec{j}}{|\vec{r}_{\text{arm}, i} \times \vec{j}|} \right) \times \vec{r}_{\text{arm}, i} \\
\vec{r}_{\text{eff}, z} \equiv \left( \frac{\vec{r}_{\text{arm}, i} \times \vec{k}}{|\vec{r}_{\text{arm}, i} \times \vec{k}|} \right) \times \vec{r}_{\text{arm}, i}
\]

While the effectivity vectors may technically fall in any direction contained by the plane normal to the hinge vector and passing through the point describing the given surface element, the above choices are convenient for a discrete formulation.

Lastly, accounting for dual-sided thin baffle surfaces requires that a pressure and skin friction vector sums be utilized in place of only the pressure or skin friction:

\[
(-P \vec{n} + f \vec{t})_{\text{net}} \equiv (-P_{1} + P_{2})\vec{n} + (f_{1}\vec{t}_{1} + f_{2}\vec{t}_{2})
\]

Eq. (134)

Here, subscript 1 denotes the scalar values on the side of the thin-baffle surface for which the surface normal vector points outward, and subscript 2 denotes the scalar values on the
opposite side of the thin-baffle surface. The components corresponding to subscript 2 may be simply assumed to be zero for a “thick” surface.

Then, the forces and moments resulting from a pressure and shear force distribution are mathematically expressed as:

\[ \overline{F} = \langle f_x, f_y, f_z \rangle = \iint_A \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net} \cdot \overline{i}, \overline{j}, \overline{k} \, dA \approx \sum_{i=1}^{n} \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net,i} \cdot \overline{i}, \overline{j}, \overline{k} \, A_i \]

Eq. (135)

\[ \overline{M} = \begin{pmatrix} m_x \cr m_y \cr m_z \end{pmatrix} \]

Eq. (136)

\[ m_x = \iint_A \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net} \cdot (r_{\text{eff},x}) \, dA \approx \sum_{i=1}^{n} \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net,i} \cdot (r_{\text{eff},x}) A_i \]

Eq. (137)

\[ m_y = \iint_A \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net} \cdot (r_{\text{eff},y}) \, dA \approx \sum_{i=1}^{n} \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net,i} \cdot (r_{\text{eff},y}) A_i \]

Eq. (138)

\[ m_z = \iint_A \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net} \cdot (r_{\text{eff},z}) \, dA \approx \sum_{i=1}^{n} \left\{ -P \overline{n} + f \overline{t} \right\}_\text{net,i} \cdot (r_{\text{eff},z}) A_i \]

Eq. (139)

Then, for coefficients [22]:

\[ \tilde{C}_\overline{F} = \iint_A \left\{ -C_p \overline{n} + C_f \overline{t} \right\}_\text{net} \cdot \overline{i}, \overline{j}, \overline{k} \, dA = \frac{\overline{F}}{q_\infty S_{\text{ref}}} \]

Eq. (140)

\[ \tilde{C}_\overline{M} = \begin{pmatrix} 2m_x \\ \frac{m_y}{q_\infty S_{\text{ref}} b_{\text{ref}}} \\ \frac{2m_z}{q_\infty S_{\text{ref}} c_{\text{ref}}} \end{pmatrix} \]

Eq. (141)
The assumption of an orthonormal power law velocity distribution which holds all the way to the wall is equivalent to the assumption of zero skin friction. This is since Stokes' Law states that for Newtonian fluids, skin friction is proportional to the spatial rate of change of velocity in the normal direction at the surface [38], while a power law distribution's derivative will always evaluate to zero when evaluated at the wall. This lack of a shear force means the only remaining nonzero physical mechanism for fluid dynamic forces to act upon the surface is through pressure, in the surface normal direction. Then, simply adapting Eq. (134)'s definition for the net forces acting on a surface element of area:

\[
(-P \mathbf{n} + f \mathbf{t})_{\text{net}} = \left( (-P_1 + P_2) \mathbf{n} \right) + \left( 0 \mathbf{r}_1 + 0 \mathbf{r}_2 \right) = \left( -P_1 + P_2 \right) \mathbf{n} \tag{142}
\]

However, a nonzero shear force would allow the surface integration method to compute separate pressure and viscous force and moment contributions.

These coordinate force coefficients may be then transformed from Cartesian force vector components to components of lift, drag and side force through the use of a partial inverse coordinate system rotation (e.g. about the Y-axis, considering sideslip to be zero) [22] as follows:

\[
C_L = C_Z \cos(\alpha) - C_X \sin(\alpha) \tag{143}
\]

\[
C'_{Y} = -(C_X \cos(\alpha) + C_Z \sin(\alpha)) \sin(\beta) + C_Y \cos(\beta) \tag{144}
\]

\[
C_D = (C_X \cos(\alpha) + C_Z \sin(\alpha)) \cos(\beta) + C_Y \sin(\beta) \tag{145}
\]

Moments about axes follow the same form as the forces directed along those axes:
\[ C_n = C_{Mz} \cos(\alpha) - C_{Mx} \sin(\alpha) \quad \text{Eq. (146)} \]

\[ C_m = -(C_{Mx} \cos(\alpha) + C_{Mz} \sin(\alpha)) \sin(\beta) + C_{My} \cos(\beta) \quad \text{Eq. (147)} \]

\[ C_l = (C_{Mx} \cos(\alpha) + C_{Mz} \sin(\alpha)) \cos(\beta) + C_{My} \sin(\beta) \quad \text{Eq. (148)} \]

For a three-dimensional extrusion (e.g. a two dimensional profile), assuming standard orientation (X-axis aligned to chordwise direction and Z-axis aligned to thickness direction) and zero sideslip simplifies these expressions. Any out-of-plane forces or moments become zero:

\[ C'_l = C_l = C_n = 0 \quad \text{Eq. (149)} \]

Therefore:

\[ C_L = C_x \cos(\alpha) - C_X \sin(\alpha) \quad \text{Eq. (150)} \]

\[ C_D = C_x \cos(\alpha) + C_Z \sin(\alpha) \quad \text{Eq. (151)} \]

\[ C_m = C_{My} \quad \text{Eq. (152)} \]

The Y-direction force and moment coefficients remain unmodified. Trading force coefficient nomenclature with two-dimensional conventions, and assuming arbitrary moment arm and effectivity vector orientations are horizontal and vertical, respectively, simplified surface pressure summations for section lift, drag and pitching moment coefficients are obtained:

\[ C_l = \left[ \sum_{i=1}^{n} (P_{net} \hat{n}_i \cdot \hat{k}) \cos(\alpha) - \left( \sum_{i=1}^{n} (P_{net} \hat{n}_i \cdot \hat{i}) \right) \sin(\alpha) \right] A_i \quad \text{Eq. (153)} \]
\[ C_d = \left[ \sum_{i=1}^{n} \left( (P_{net}^i \hat{n}_i \cdot \hat{t}_i) \right) \cos(\alpha) + \left( \sum_{i=1}^{n} \left( (P_{net}^i \hat{n}_i \cdot \hat{k}_i) \right) \sin(\alpha) \right) \right] A_i \]  
Eq. (154)

\[ C_m = \sum_{i=1}^{n} \left( (P_{net}^i \hat{n}_i \cdot \hat{k}_i) \right) [A_i] \hat{p}_{arm} \]  
Eq. (155)

Above, the arbitrary choice of moment arm for pitching moment is the X-axis, with magnitude equal to the distance between hinge line and panel center.

The only remaining problem to resolve for the surface integration method is that after extrapolating field data back to selected surface nodes it can be clearly seen that only a subset of relevant surface nodes receive data from the flow field.

In two dimensions, a linear interpolation scheme is applied on the basis of arc length to extend this sparse data to the full level set. This is accomplished recursively on any arbitrarily sized intervals where the field data was not detected to have been extrapolated back to the surface. Special treatment is warranted when level set end points or sharp geometric discontinuities do not receive field data directly. For the current work, this treatment amounted to an additional one-sided extrapolation of field data from the nearest level set point populated during the flow solver's execution, prior to filling out the remainder of the level set points via interpolation.

On a paneled three dimensional surface, an analogous, more complex yet less precise treatment may be achieved by expanding properties outward from populated panels to unpopulated ones, in a constant manner until the surface is fully populated.

Then, the surface solution is output directly, and the aforementioned surface
pressure integration procedure may be applied.

Momentum deficit methods may require a larger domain for accuracy and longer convergence times for stabilization, and cannot easily determine moments. Momentum deficit methods also leave some question as to proper control volume placement and size. Although specialized formulations of momentum deficit methods offer a potential for isolating certain subsets of forces such as wave drag, this is not relevant for an incompressible formulation.

The force aligned with a coordinate direction can be found by summing the net momentum fluxes calculated on the cell interfaces coinciding with a control volume. Mathematically, from the steady-state portion of the momentum equations in a Cartesian coordinate system and the Gauss theorem:

\[
F_x \approx - \iint_A \left( (\rho uu + P - \tau_{xx})|\vec{n} \cdot \vec{i}| + (\rho vv + P - \tau_{yx})|\vec{n} \cdot \vec{j}| + (\rho wv + P - \tau_{xz})|\vec{n} \cdot \vec{k}| \right) dA \quad \text{Eq. (156)}
\]

\[
F_y \approx - \iint_A \left( (\rho uu + P - \tau_{yx})|\vec{n} \cdot \vec{i}| + (\rho vv + P - \tau_{yy})|\vec{n} \cdot \vec{j}| + (\rho wv + P - \tau_{yz})|\vec{n} \cdot \vec{k}| \right) dA \quad \text{Eq. (157)}
\]

\[
F_z \approx - \iint_A \left( (\rho uu + P - \tau_{xz})|\vec{n} \cdot \vec{i}| + (\rho vv + P - \tau_{zy})|\vec{n} \cdot \vec{j}| + (\rho ww + P - \tau_{zz})|\vec{n} \cdot \vec{k}| \right) dA \quad \text{Eq. (158)}
\]

Other works have included accumulation-like time terms [16, 13], which likely improves or eliminates the lag and temporal decoupling from surface methods noted below with the momentum summation technique in this paper. These expressions translate directly to interface flux summations on the boundaries of an arbitrary control surface surrounding the computational domain; they may also be transformed to a volume integral (or summation over a volume of computational cells, including control volume interior)
using the Green or Gauss theorems. Using the volume integral and including the
neglected time term may be particularly helpful if large accumulations or abrupt
transients occur, or if the formulation does not exhibit the telescoping property (the
ability for the flux reconstruction to perfectly cancel when applied to either side of a
steady-state cell interface). LDFSS exhibits this property.

The coordinate force totals for the momentum deficit methods may be
transformed between coordinate systems similarly to the surface pressure integration,
using Eq. (143) to (145) for three dimensional results or Eq. (150) to (151) for two
dimensional results. Regardless, either momentum deficit method may be applied on a
near-body or farfield basis.

Figure 2.9.1 below illustrates a calculation history of lift predictions using the
detailed momentum method in the far field control volumes and control surfaces, versus
near-body control volumes and control surfaces.

![Figure 2.9.1 – Lift History on NACA0012 (α=16° & Reₜ = 2e6)](image)

Figure 2.9.1 – Lift History on NACA0012 (α=16° & Reₜ = 2e6)
The far field regions stretched within 10 cells of the calculation domain boundaries; the near-body regions were no more than 10 cells away from the surfaces. Relative lag is observed on the far field methods, as the proper flow field information has yet to propagate through the domain and replace the initial solution. The inner region calculations are coupled closely enough with the assumed instantaneous surface pressure integration such that they may also be considered instantaneous in this work, for purposes of monitoring convergence or output. Control surface integration also trails slightly behind control volume integration. Even in the steady state, the far field calculation histories exhibit a phase shift from the near-body and surface pressure integration histories, and even different amplitudes and ranges.

2.10 - Grid Generation of Immersing Grid and Surface Nodes

In general, the immersing Cartesian grids for this work are generated using industry guidelines for spacing and stretching in overset structured grids using wall functions [8], while adjusting for the size scales and location of surface features of interest. The immersed surface grids are then sized relative to the surrounding immersing volume grid.

Uniform, isotropic volume grid spacing on the order of $Y^+ = 100$ is provided in regions with high spatial or temporal flow gradients or curvature; these regions include the leading and trailing edges of airfoils, and the regions over which bodies move or
phenomenon of interest lie. On the basis of the ratio of adjacent cell lengths, grid stretching was in general less than ten to fifteen percent, but never more than twenty-five percent. Cell aspect ratio was disregarded, as is common for near-body grid regions in viscous solvers. Furthermore, refinement of the immersing grid near small surface features much be fine enough to resolve those features.

Beyond the requirement that nodal spacing of the surfaces defining the bodies are densely refined relative to the immersing grid, surface representation is relatively unimportant to the accuracy of the calculation. This is due to the fact that the surfaces are represented entirely by the immersing grid during flow simulation. Surface nodes were refined to roughly a factor of four of the surrounding grid spacing at a minimum, and located exactly in cell-center plane for two-dimensional calculations. Thick surfaces should have sufficient thickness to allow resolution on the immersing grid.

3 - Results

3.1 - Ducts With Thin-Baffle Turning Vane

The three-dimensional proof-of-concept case chosen was a duct turning vane design. For a given internal flow system, achievable mass flow, internal drag, pressure recovery and other measures of performance diminish as a strong function of the total pressure losses incurred at ducting turns [28]. Experiment-based analytical methods exist [29] which show that a properly designed turning vane configuration can minimize these
losses, and eliminate or reduce separated regions downstream of sharp turns. Starting with only knowledge of the duct geometry a priori presents a typical turning vane design problem. The ability to complete a design iteration quickly is demonstrated by producing a parametric thin-baffle turning vane and performing an analysis upon it in an independently gridded duct.

Two different ducts were used as test cases to troubleshoot the initial capabilities of the method, including a thin-baffle surface treatment.

The first duct was a three-dimensional, rectangular profile geometry physically measuring 12.3m long, and around 2.5m high average by 7.3m wide average, represented by 193 (i) x 49 (j) x 49 (k) stretched, skewed cells which were broken into 5 multigrid-capable blocks in the streamwise direction. The main intent was to test the thin-baffle distance function, and the method itself, as both were still undergoing development. Figure 3.1.1 below illustrates the grid geometry, nodal surface mesh and normal vectors.

![Figure 3.1.1 – 3D Complex Rectangular Duct Grid Block, Surface Nodes & Vectors](image)
This early case's results serve as an important example of poor practices for the immersed boundary method. As is visible below in Figure 3.1.2, the immersed surface nodes were widely spaced and decoupled from grid spacing in the Z-direction, and spacing in the Y-direction was sparse and stretched near the thin baffle surface. Not surprisingly, the discrete grid representation of the surface is disjointed and ill-defined. This resulted in the flow solution not achieving tangency and overall poor solution quality and convergence, which is visible in the stream traces shown in Figure 3.1.3.

Figure 3.1.2 – Complex Duct Sparse Grid, Distance Function Contour and Poor Blanked Heaviside Representation of Smooth Surface
These items were fixed in the straight duct simulation, where the immersed surface acts more as a mixing vane and restrictor. Uniform, isotropic grid and a relatively refined surface node collection are matched on a single true two-dimensional plane to ensure good discrete grid representation of the surface. A 36 meters length straight duct, 4.35 meters high and 7.5 centimeters wide are resolved by 8 evenly divided grid blocks arranged in the length direction, measuring 481 (i) by 59 (j) by 2 (k) computationally.

These grid and surface attributes produced reasonable fluid behavior and good convergence. Figure 3.1.4 below illustrates stream traces and the heaviside function, while Figures 3.1.5 and 3.1.6 show U-velocity and pressure coefficient contours with a blanked grid representation of the surface, respectively.
The asymmetric areas allotted for the flow above and below the vane force the flow to expand around the leading edge and rarify below the vane to the point of separating at the curvature reflex mid-chord; above the vane, near stagnation occurs upstream, which accelerates as it moves into the channel between the vane and wall. Variation of this behavior over time produces a cyclic shedding downstream. These prominent flow
3.2 - Single-Element Symmetric NACA0012 Airfoil and Trailing Edge Device

The first validation case chosen was a study on a trailing edge device. In the early 1970's, Dan Gurney pioneered the use of a simple short orthogonal tab on the pressure side of race car rear wing trailing edges, which had the effect of increasing both zero-angle and attainable lift with a slight penalty in drag. Since then, Gurney flaps (and similar dual-sided T-strips) have become common additions to various aircraft surfaces, to fix an assortment of different unwanted handling characteristics including roll trim, rudder centering, stick and rudder pedal force issues. However, in the author's own experience, properly sizing these devices can require several expensive flight test iterations. Quickly and easily determining proper sizing computationally is simulated by determining force and moment changes due to a single tab size. A single-element symmetric NACA0012 airfoil is modeled with and without a 1% chord Gurney flap on the lower surface trailing edge.

The airfoil chord length measures one meter, with chord line aligned to the X-axis and leading edge located at the origin. A total of 2,164 nodes make up the surface definition, ranging smoothly from a maximum spacing of around 0.1% chord in low-curvature regions to under 0.05% chord around the leading and trailing edges, on the Y=0.5mm plane. 420 nodes comprise the added 1% chord Gurney flap. Figures 3.2.1 and
3.2.2 below illustrate the arrangement of the surface nodes and orientation of the surface node normal vectors for the geometry with the Gurney flap.

Figure 3.2.1 – Surface Nodes comprising NACA0012 with 1% Gurney Flap on Lower Surface Trailing Edge

Figure 3.2.2 – Normal Vector Orientation of Surface Nodes comprising NACA0012 with 1% Gurney Flap on Lower Surface Trailing Edge

The surrounding Cartesian grid measures 409 x 2 x 280 in computational space and 100m x 0.001m x 100m in physical space, with 8m farfield spacing and uniform, isotropic 0.1% chord extending a distance of 3% chord from the leading and trailing edges. The geometry is extruded one millimeter in the Y-direction. Around the maximum thickness location, the chordwise spacing stretches to 1% of chord length. Grid stretching does not exceed 20% in any direction, and in any location.

Figures 3.2.3, 3.2.4, 3.2.5 and 3.2.6 below show far field and near-body views of
the blanked immersing grid surrounding the Gurney flap case, with zoomed views of the leading and trailing edge regions. A chord-based Reynolds number of 2,100,000 was simulated for close comparison to available experimental data [11, 30, 39, 42].

Figure 3.2.3 – Far field view of Stretched Cartesian Grid Containing NACA0012

Figure 3.2.4 – Near-body view of Stretched Cartesian Grid containing NACA0012

Surface Nodes
Figure 3.2.5 – Leading Edge Zoom of Stretched, Blanked Cartesian Grid for Clean NACA0012 Airfoil

Figure 3.2.6 – Trailing Edge Zoom of Stretched, Blanked Cartesian Grid for Clean NACA0012 Airfoil
Figure 3.2.7 shows nondimensional velocity contours predicted by the immersed boundary code around the 1% chord Gurney flap configuration at zero angle of attack.

The expected flow field features are seen, including leading edge stagnation, higher acceleration around the suction side of the surface than the pressure side, pressurization ahead of the Gurney flap, and a circulation zone behind the Gurney flap. A viscous response is seen near the immersed boundary surface, where the width of the velocity defect grows similarly to a boundary layer.

Figures 3.2.8 and 3.2.9 below provide the surface node pressure distributions filled from extrapolated immersed boundary code flow field data, for both the clean and 1% chord Gurney flap configurations at zero and four degrees angle of attack.
Figure 3.2.8 – Calculated Pressure Coefficient Distributions for Clean NACA0012 and 1% Chord Gurney Flap Configurations at Zero Angle of Attack

Figure 3.2.9 – Calculated Pressure Coefficient Distributions for Clean NACA0012 and 1% Chord Gurney Flap Configurations at Four Degrees Angle of Attack
The immersed boundary code successfully predicts that the Gurney flap widens the pressure difference across the suction and pressure sides of the airfoil, creating a net lift increment.

Figures 3.2.10, 3.2.11 and 3.2.12 below illustrate the section force and moment properties of the NACA0012 with and without the Gurney flap, compared to published experimental data [30, 39] at chord-based Reynolds numbers of 2.1 million and 2.19 (3.5 corrected) million, respectively.

Figure 3.2.10 – Calculated and Published Experimental Section Lift Coefficient versus Angle of Attack for both NACA0012 Configurations
For higher angles of attack, the immersed boundary code results provided force and moment values which oscillated significantly over time. For brevity, a visual average for each configuration at the highest angle of attack was used instead of a more accurate technique such as time-averaging the results of a fast-Fourier transform.
Sectional lift coefficient is matched very well for both NACA0012 configurations at zero degrees angle of attack. The immersed boundary code under-predicts lift curve slope by around 10% (which probably indicates compressibility in the experimental results), but roughly matches the stall angle shown in the experimental data. The section lift coefficient increment between configurations is well-predicted by the immersed boundary code for the entire range of angles tested.

Section drag coefficient agrees surprisingly well, given that the experimental data includes laminar and turbulent portions of skin friction drag while the computational data is essentially a pressure drag value. Regardless, trend with angle of attack and delta between the two configurations appears reasonable if not precisely accurate. Using a book-method from a well-known source for calculating incremental drag increases [28] shows a frontal-area based independent drag coefficient of 1.25 (for h/δ = 7.5%). Correcting from tab height to chord length (e.g. multiplying by 0.01m / 1m) and adjusting to an approximate height-to-boundary layer thickness (e.g. multiplying by the cubic root of 1%/7.5%) yields roughly 65 counts of sectional drag result from adding a two-dimensional protuberance of height 1% chord on the trailing edge of a 1m airfoil. The drag delta calculated by the immersed boundary code between configurations looks reasonable relative to both experimental data and a book method.

Section quarter-chord moment coefficient values agreed well with experimental for the entire range of tested angles, for both conditions, including the change in section quarter-chord moment coefficient between configurations.
Similar low-speed experimental data for an untwisted, cambered NACA23012 wing with aspect ratio 6, Gurney flap height of 1% chord and at chord-based Reynolds numbers of between 0.51 and 1.95 million is also available [11]. It shows a lift coefficient increment of 0.19 at zero angle of attack, a maximum lift coefficient increase of 0.31, a quarter-chord moment shift of -0.05 at zero angle of attack, and a drag coefficient increment of around 70 counts at zero angle of attack. These values closely match the increments in section force and moment coefficients seen on the two-dimensional airfoil results using the immersed boundary code.

From the results of an available computational study on a similar symmetric NACA0011 airfoil at a chord-based Reynolds number of 2.2 million [42], a section lift coefficient increase of 0.30 at zero angle of attack, a maximum section lift coefficient increase of 0.40, a visually imperceptible section drag coefficient increase at zero angle of attack, and a section quarter-chord moment decrease of around -0.07 at zero angle of attack are all seen. The lift and moment values are 30% or so larger than predicted by the immersed boundary code.

Figures 3.2.13 and 3.2.14 provide flow visualization of the flow structure captured by the immersed boundary code for the NACA0012 configuration with the Gurney flap height of 1% chord, at zero and positive angles of attack.
A qualitative comparison of these flow structures to those in published works [11, 30, 42] show good agreement at zero angle of attack, and reasonable agreement at 10 degrees angle of attack- despite the size of the geometry and flow features being on the same order of magnitude of the surrounding grid cells (which is visible by noting the blanked geometry contour is pixelated coarsely to the surrounding grid by the plotting package).
3.3 - Single-Element Cambered NACA23012 Airfoil and Leading Edge Ice Contamination

The second validation case chosen was an investigation on the performance and handling effects of ice accretions. At altitudes extending from the ground upward to over twenty thousand feet above mean sea level (MSL), and at mild to moderate sub-freezing temperatures, aircraft periodically encounter clouds containing supercooled droplets of water which will impinge upon surfaces and freeze. Despite the availability of several proven icing protection systems (IPS), regulations for Flight Into Known Icing (FIKI) require that an aircraft be able to handle a certain level of ice accretions before and between IPS cycle operation, on unprotected surfaces, and during IPS failure. Depending upon exposure time, exposure intensity, flight condition and IPS operation, these ice accretions can dramatically lower achievable lift and increase drag at all conditions. Mild to severe impacts to aircraft handling also may occur. Icing conditions frequently play a role in accidents [47].

Currently, impingement and shape prediction codes exist ([52], for example) which can accurately model these effects, given an accurate flow field solution at each time step. However, this requires either applying low-order fluid simulations (e.g. panel methods) to the evolving surface shapes, or pairing more sophisticated fluid simulation techniques with grid regeneration between each accretion time step. These codes typically require flow field input, and they output ice growth evaluated at a given surface grid node. Therefore, the grid independence of the surfaces and node-normal vector
surface data structure of the Immersed Boundary Method currently implemented offers a uniquely simple approach to integrating a more sophisticated flow solver into existing ice impingement and shape prediction codes.

As a first step in that direction, this work simulates an ice shape in the flow solver using only surface nodes. A single-element cambered NACA23012 airfoil is simulated with and without leading edge contamination from an atmospheric icing encounter.

The airfoil chord length again measures one meter, with chord line aligned to the X-axis and leading edge located at the origin. 2,735 nodes make up the surface definition at Y=0.5mm, with curvature-dependent surface spacing stretching slowly from 0.05% to 0.1% chord. 714 additional nodes are added to define the shape of the leading edge contamination. The surface nodes and normal vectors for the configuration with leading edge ice contamination are plotted in Figures 3.3.1 and 3.3.2 below.

Figure 3.3.1 – Surface Nodes for NACA23012 With Leading Edge Contamination

Figure 3.3.2 – NACA23012 Vector Orientation with Leading Edge Contamination
As a demonstration of the versatility of the immersed boundary method, the same grid generated for the single-element symmetric airfoil cases was utilized for this test case. Figure 3.3.3 below shows the clean and contaminated surface nodes, along with the blanked grid geometry for the leading edge of the NACA23012 airfoil.

![Figure 3.3.3 – Clean & Ice-Contaminated NACA23012 with Grid Representation](image)

Although not shown, a similar ice shape geometry can be approximated in a common icing impingement and ice accretion shape prediction codes [52] using 15 micron droplets, 0.74 g/m$^3$ liquid water content, low angles of attack and a 17 nm cloud horizontal extent. This shape is typical of a large double-horn glaze ice formation in small droplet, relatively warm, high liquid-water-content meteorological conditions.

Both the clean and contaminated leading edge geometries exhibit the flow field features expected. Figure 3.3.4 below illustrates the near-surface velocity magnitude contours around the iced configuration.
Leading edge stagnation is present, with high acceleration around the sharp edges of the ice shape followed by recirculating separation bubbles and a wide velocity defect.

Figure 3.3.5 shows similar immersed boundary code results for the iced configuration after a well-developed stall has occurred at 8 degrees angle of attack. Sharp, large-scale flow gradients have moved beyond the grid clustering near the body.
Figures 3.3.6, 3.3.7 and 3.3.8 provide a comparison of calculated clean and iced configuration surface pressure coefficient distributions versus clean configuration experimental data for three different angles of attack.

Figure 3.3.6 – Clean & Iced Pressure Distributions at Zero Degrees Angle of Attack

Figure 3.3.7 – Clean & Iced Pressure Distributions at Four Degrees Angle of Attack
Although experimental pressure distributions were only found for the clean configuration, generally good agreement is seen with the results calculated with the immersed boundary code. Immersed boundary code results for the iced configuration at four degrees angle of attack closely resemble experiments on a casting and other CFD analysis results, both at chord-based Reynolds numbers of 1,800,000 and five degrees angle of attack [7, 9]. The leading edge suction peak is slightly under-predicted at higher lift conditions, most likely due to neglecting compressibility. Large jumps in surface pressure are predicted around the sharp, small leading edge ice accretion, alternating between stagnation at blunt forward-facing features and intense localized expansions.

Figures 3.3.9, 3.3.10 and 3.3.11 show section lift, drag and moment coefficient curves calculated for both the clean NACA23012 configuration and the configuration
with leading edge ice contamination. Clean configuration experimental results from two other sources are provided at nearly incompressible Mach numbers and chord-based Reynolds numbers of around three million [50] and two million [7, 9]; the latter of these sources also provides subscale iced airfoil results.

Figure 3.3.9 – Section Lift Coefficient Curves for Clean & Iced LE NACA23012

Figure 3.3.10 – Section Drag Coefficient Curves for Clean and Iced LE NACA23012
Figure 3.3.11 – Section Quarter-Chord Pitching Moment Coefficient Curves for Clean and Iced Leading Edge NACA23012

The immersed boundary code results match experimental section lift coefficient very well at low angle of attack for both configurations. Trend over the entire range of angles of attack is also matched well. The lift curve slope is underestimated by around 13% for the clean configuration and slightly more for the iced geometry. This is again most likely attributable to the incompressible solver. The immersed boundary results for the configuration with leading edge ice contamination experience a premature stall and around 33% more loss in maximum lift than the experimental results, although the plotted results were averaged visually due to highly unsteady oscillations. These values tend on the conservative side, for interpretive purposes.

The calculated section drag coefficients also match well, with the exception that the clean configuration drag calculated by the immersed boundary code at zero angle of attack is much lower that the experimental value. Changes in calculated section drag
coefficients between the clean and iced airfoils match the experimental deltas very well. The section drag of the iced airfoil is nearly an order magnitude higher than the clean configuration for both calculated and experimental results.

An available reference for the minimum drag of two cambered airfoils with various leading edge ice accretions [28] shows a much lower minimum drag value (160 counts) than the computational and experimental NACA23012 results. This is likely due to the ice horn angles and heights in the reference resulting in a far more streamlined shape; by comparison, the accretion simulated by the immersed boundary code and tested by the experimental data protrudes further from the clean geometry in the local flow-normal direction.

The section quarter-chord pitching moment coefficients calculated by the immersed boundary code for the NACA23012 airfoil configurations appear mildly exaggerated in trend and slightly more leading-edge-up in general than experimental results.

3.4 - Multi-element Airfoil With High-Lift Devices Extending Over Time

The third test case chosen was a moving-body simulation of high lift elements on a airfoil moving between cruise and landing configuration.

Almost all civilian aircraft utilize long-duration flap extension travel times to allow pilots time to react and reorient to aircraft handling changes between the two
configurations. However, this gradual adjustment phase is unnecessary for unmanned aerial vehicles and aircraft with fly-by-wire type control systems between the aircraft and pilot. Regardless, decreasing deflection times will at some point result in an unsteady fluid response. The presence of such a response at an arbitrarily shorter deflection time is investigated.

A three-element cambered McDonnell Douglas 30P-30N airfoil is simulated with slat and flap movement modeled over time, at eight degrees angle of attack and a chord Reynolds number of nine million.

With all elements retracted, the airfoil's one-meter long chord line coincides with the X-axis and the leading edge point is located at the origin. The main element, flap and slat are defined by 3,015 nodes, 1,601 nodes and 2,689 nodes at Y=0.5mm, respectively. Spacing varies gradually from a maximum of 0.1% chord down to below 0.05%. Sharp corners actually defined by very finely spaced points on a very small radius, to avoid altering high-lift element gap widths and overlap. A chord-based Reynolds number of 9 million was used for both the computation in this work, and the computation and experiment in reference data [17, 43].

Movement of the high-lift elements is modeled using a combined prescription of surface and hinge line translational position, and surface rotational position, at endpoints along a line segment curve. Linear interpolation is carried out between these endpoints in time. The effect resembles dual-roller motion along a track. With a prescription of position, derivatives such as surface velocities and accelerations are easily found by
differencing between positions at nearby movement time levels. Although central differencing was utilized, backwards differencing is likely more consistent with the fluid simulation. Total slat and flap deflections are 30 degrees each (in opposite directions).

Movement begins with both high lift elements stationary and retracted, and this pause persists for long enough for the flow field to become initialized. The elements pull directly away from the body before deflecting towards their final positions. The flap drops down before rotating trailing-edge down and translating aft, and the slat lifts up off of the main element before rotating nose-down and translating forward. The entire motion of the extension of both the flap and slat are completed over the course of 0.09 seconds.

A surface node and normal vector time history is plotted in Figure 3.4.1 below.

Near the surfaces, immersing grid spacing ranges from isotropic, uniform spacing of 0.1% chord in leading edge and trailing edge regions of motion, to around 1% chord length at the mid-chord location. Farfield spacing is 8 chord lengths, at a distance of 50 chords from the body; extruded two-dimensional width is 0.1% chord length. Stretching is smooth and does not exceed a stretching ratio of 15%. The domain computationally measures 897 (i) x 2 (j) x 327 (k), and is partitioned into three blocks of roughly the same size.
Figure 3.4.1 – Surfaces & Normal Vectors (0, 20, 40, 65 & 100% of Extension)
Figure 3.4.2 below shows the resulting blanked distance function representation of the bodies on the immersing grid, with steady-state-type streamline flow visualization.

Figure 3.4.2 – Blanked Distance Function & Streamlines for Moving-Body Multi-Element Airfoil Case at 0, 10, 40, 70 and 100% of Extension Motion
The surfaces are well-resolved, but an unoccupied isolated region is seen in the cove above the flap (early in the deflection, and the slat cove opens up after movement begins). These regions experience severely low pressures as a result of regions of initially sealed freestream density are stretched, and external flow rushes to equalize them.

Figure 3.4.3 below shows these velocity magnitude contours. The high-speed channel and suction peak regions, and lower-speed coves, stagnation points and slat velocity defect are all clearly shown.

A stagnant flow artifact originating from the slat cove is visible being diffused and convected downstream along the main element's upper surface in the first few illustrated time steps.
Figure 3.4.3 – Velocity Magnitude Contours for Moving-Body Multi-Element Airfoil at 0%, 10%, 20%, 30%, 60% and 100% of Extension Motion
Figure 3.4.4 below illustrates pressure coefficient contours.

Figure 3.4.4 – Pressure Coefficient Contours for Moving-Body Multi-Element Airfoil at 0%, 10%, 20%, 30%, 60% and 100% of Extension Motion
The extension of the high-lift elements is easily observed to increase pressure and suction on the upper and lower surfaces, respectively; however, the slat and main elements each also appear to successfully limit the peak suction values on their neighboring downstream elements.

The results predicted for the moving-body extension appear reasonable. In Figure 3.4.5 below, stabilized steady-state surface pressure coefficient distributions are given for the extended configuration. These are compared to experimental results [43] and computational results [17].

![Figure 3.4.5 – Experimental and Computational Surface Pressure Coefficient Contours on Multi-Element Airfoil at In Steady State after Extension](image)

The steady-state pressure coefficient distributions appear to slightly underpredict...
the suction on the upper surfaces by an amount resulting in roughly 10% lower lift than experimental data. This is most likely due to not accounting for compressibility. However, trends match very well.

Lastly, the lift, drag and moment coefficient histories are presented below in Figures 3.4.6 through 3.4.8. In these plots, time zero is a steady-state retracted solution, and the time at 0.09 seconds later is the stationary configuration with completed extension. It is seen that some unsteady response to the extension is created. Initially, the solution lags behind the extension, but then changes continue after it is completed.

![Figure 3.4.6 – Lift Coeff. Time History During Multi-Element Extension Motion](image)

Figure 3.4.6 – Lift Coeff. Time History During Multi-Element Extension Motion
Figure 3.4.7 – Drag Coeff. Time History During Multi-Element Extension Motion

Figure 3.4.8 – Pitching Moment Coeff. History During Extension Motion
As expected, instantaneous lift begins growing steadily as the slat and flap begin deploying; however, it lags behind the steady-state value, and continues developing after the movement completes, with a slight but rapid increment as movement ceases. Calculated drag increases almost linearly over the extension time, but tapers off after the movement ceases. Quarter-chord pitching moment acts in an increasingly nose-down manner, as a larger and larger percentage of total developed lift moves aft of the aerodynamic center. All three parameters show noticeable noise near 10% of the way through the extension motion, where the high lift elements separate from the main element.

Although no references were located for force and moment comparison, configurations with slats and flaps at similar deflections from [1] exhibit changes in lift, drag and pitching moment of the same order of magnitude.

4 - Conclusions

The incompressible Navier-Stokes equations with Spalart-Allmaras turbulence are solved with the immersed boundary method at high Reynolds number. The primary applications are chosen from the Aerospace industry, including an unsteady two-dimensional moving body case simulating high lift element deployment over time, several steady two-dimensional results for a Gurney flap on a symmetric airfoil and leading edge ice contamination on a cambered airfoil, and limited three-dimensional and
Comparing calculated values from the immersed boundary code to experimental data, very good agreement in lift is seen at low angles of attack, while higher angles of attack tend to underpredict lift. Moderate to good agreement is seen with both drag coefficient and quarter-chord pitching moment. Trends for all forces and moments match well throughout the envelope of flight conditions in the validation cases. Force and moment increments match very well for almost all cases at almost all conditions. Pressure results also agree very well at low angles of attack, and upper surface suction is slightly underpredicted in the same situations as lift. Suction peak under-prediction is most likely attributable to local compressible flow physics which are not captured by the incompressible solver. Separation angle matched well for most cases.

The versatility of the immersed boundary method for first-cut design, quick approximate analysis and moving body simulation at high Reynolds numbers in the Aerospace industry is demonstrated due to its ability to allow independent changes to the immersing grid, surface definition and a movement prescription for multiple bodies while requiring minimal revision to the remaining inputs.

Related low Reynolds number immersed boundary code results from classic experiments which have been previously published agree very well with experimental data [16]. Particle resuspension and transport simulations also agree well with available experimental data [34].
5 - Recommendations for Future Research

Many directions lie open to pursue further research beyond the current work. A few of these directions include increases in accuracy and computational efficiency, application to a compressible flow solver, further closure of the method to reduce the required inputs, improvement of the decomposition of the domain into fluid and surface regions, improvement of the thin baffle extension, extension of surface force and moment integration to components in three dimensions with arbitrary coordinate systems, the use of an interpolation function or correlation which offers realistic nonzero skin friction values, and the inclusion of generalized six-degree-of-freedom movement modules.

The primary weakness of the method at high Reynolds numbers appears to be under prediction of the magnitude of suction on bodies at high lift conditions. This should be investigated further. A compressible flow solver would most likely eliminate this issue, and compressibility paired with three-dimensional surface force and moment integrations would greatly widen the relevance of the method to high Reynolds number flows and Aerospace applications, since a majority of these flows are highly three-dimensional and exist beyond the incompressible regime. Further testing of the capability to handle infinitely thin surfaces is also warranted.

Efficiency is another area where improvement might be made. A sensitivity study upon grid refinement may show an allowable reduction in the required grid size with little to no change in accuracy. Specializations such as localized time stepping for steady-state cases and the ability to completely toggle off unwanted output, and three-dimensional
contributions for two-dimensional flows, would allow much faster analysis.

To become increasingly useful as a first-cut analysis and design tool, and to simulate a transition-like behavior in the boundary layer surrounding the body, closure of some of the method's inputs (such as power law value and reference velocity) could be sought. An ability to estimate skin friction and wave drag contributions to forces and moments, and a more complete, general movement capability would also ease the usage of the method.

Ambiguity in the domain partitioning between interior, band and exterior points may also be addressed with further research. In particular, a closest-approach formulation between a paneled body (instead of just panel corner points) and the immersing grid, paired with surface normal vector interpolation between panel corner points, is proposed by the author as a potential improvement to the existing point placement methods.

6 - References


Vol. 34, No. 4, pp. 756-763, 1996.


[34] Oberoi, Roshan C. “Thesis: Large-Eddy Simulation of Particulate Resuspension and Transport Under Influences of Human-Body Motion in an Indoor Setting.”


