A STABILIZED FINITE ELEMENT DYNAMIC OVERSET METHOD

FOR THE NAVIER-STOKES EQUATIONS

By

Chao Liu

James C. Newman III
Professor of Computational Engineering
Committee Chair

W. Kyle Anderson
Professor of Computational Engineering (retired)
Committee Member

Kidambi Sreenivas
Research Professor of
Computational Engineering
Committee Member

John V. Matthews III
Associate Professor of Mathematics
Committee Member

A STABILIZED FINITE ELEMENT DYNAMIC OVERSET METHOD

FOR THE NAVIER-STOKES EQUATIONS

By

Chao Liu

A Dissertation Submitted to the Faculty of The University of
Tennessee at Chattanooga in Partial Fulfillment of
the Requirements of the Degree of
Doctor of Philosophy in Computational Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May, 2016

ABSTRACT

In terms of mesh resolution requirements, higher-order finite element discretization methods offer a more economic means of obtaining accurate simulations and/or to resolve physics at scales not possible with lower-order schemes. For simulations that may have large relative motion between multiple bodies, overset grid methods have demonstrated distinct advantages over mesh movement strategies. Combining these approaches offers the ability to accurately resolve the flow phenomena and interaction that may occur during unsteady moving boundary simulations. Additionally, overset grid techniques when utilized within a finite element setting mitigate many of the difficulties encountered in finite volume implementations. This research presents the development of an overset grid methodology for use within a streamline/upwind Petrov-Galerkin formulation for unsteady, viscous, moving boundary simulations. A novel hole cutting procedure based on solutions to Poisson equation is introduced and compared to existing techniques. A MPI-based parallel three-dimensional overset grid assembly framework is developed. Order of accuracy is examined via the method of manufactured solutions. The potential benefits of using Adaptive Mesh Refinement (AMR) in overset grid simulations are explored by combining the overset method with an AMR approach. The importance of considering linearization due to the overset boundaries within the preconditioning is studied. Numerical experiments are performed comparing an ILU(k) preconditioner with two proposed modifications referred to as “triangular inter-grid ILU(k)” and “Jacobi inter-grid ILU(k)”. The efficiency gains observed from the proposed modifications are also applicable to general parallel simulations on distributed memory machines,

regardless of whether an overset grid approach is used. Overset grid results are presented for several inviscid and viscous, steady-state and time-dependent moving boundary simulations with linear, quadratic, and cubic elements.

DEDICATION

This work is dedicated to my loving wife Maisi and to our lovely daughter Xinyue. This work is also dedicated to my parents, who teach me love and honesty.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. James C. Newman III for his support and encouragement, not only for my work, but also for my life. His dedication to his students and to his work is what impresses me most. I would also like to thank Dr. W. Kyle Anderson, who has given me tremendous guidance. His dedication to his work has always inspired me. I owe a debt of gratitude to Dr. Timothy Swafford and the entire SimCenter faculty and staff for providing an open and friendly environment during my research. Additionally, I would like to thank Bezhad Reza Ahrabi for being a supportive colleague and a great friend.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Nearly all military air vehicles are subject to carry some form of wing-mounted element. Furthermore, these vehicles are typically required to operate with numerous stores (such as munitions, fuel tanks, sensor pods, etc.) as well as with various possible configuration layouts. During the aircraft-store certification process, the carriage loads, stability and control, flutter margin, and store separation trajectories must be ascertained for each layout. In these regards, overset (sometimes referred to as Chimera) grids have found great utility. These methods allow for discretization for each of the stores to be independently generated, to best resolve the geometric complexities and flow physics, and then overlapped onto the grid resolving the aircraft. Hence, the various stores may be interchanged with ease for subsequent simulations. Moreover, for simulations in which large relative motion between bodies is present, mesh movement algorithms typically fail, requiring manual intervention to regenerate the grid and transfer the simulation data before proceeding. Overset grids do not suffer from such a breakdown since the moving body grids move relative to one another. Similar examples illustrating the need for overset grid approaches could have been made for many other application areas in which the interaction between bodies moving relative to one another must be accurately resolved.

Overset grid technology was originally developed for use in finite volume structured-grid computational fluid dynamic [1-5] software to simulate the flow around complex configurations in the early 1980's. At that time, the nemesis in performing a calculation over complicated

geometry with structured-grids was the construction of an adequate grid on which the governing equations could be solved. The development of unstructured-grid technology and simulation capabilities in the late 1980's and early 1990's somewhat mitigated this burden. However, as mentioned above, due to the shortcomings of mesh movement algorithms, overset grid methods were extended for use within unstructured-grid simulation techniques [6-8]. The overset grid assembly software SUGGAR [9] and PEGASUS [10] have been utilized for numerous finite volume solutions to complex, moving boundary aeronautical and hydrodynamic simulations. However, difficulties have been found in processing these overlapped grids for viscous computations where high stretching occurs as well as with orphan points (i.e., points where a proper interpolation stencil cannot be established) in particularly critical regions of the flow. In terms of mitigating orphan points, subsequent manual intervention is required in the aforementioned software. To this end, a method has been proposed based on scattered data interpolation techniques that eliminates the occurrence of orphan points [11], and has been used with finite volume and hybrid approaches that utilize multiple solvers.

Over the past decade, the focal point of considerable research efforts has been the development of higher-order discretization methods. These discretization methods allow for highly accurate solutions while limiting the mesh resolution requirements. They additionally offer the ability to resolve physics at much smaller scales than possible with lower order solution methods. The two most prominent higher-order methods for fluid dynamic applications are the Discontinuous Galerkin (DG) and the Streamline/Upwind Petrov-Galerkin (SUPG) finite element schemes. For the same reasons discussed above, the natural extension of overset grid methods to higher-order methods is currently taking place. To this end, a hybrid scheme whereby a finite volume scheme was utilized in the near-body field and then an overlapped grid approach used to

couple with a DG solver for off-body portions of the domain has been developed [12]. Furthermore, a space-time discontinuous Galerkin scheme formulated for structured-grids was extended with overset grid capabilities [13, 14], and more recently enhanced with implicit boundary conditions to accelerate convergence [15] and a direct hole cutting method for higher-order meshes with curved elements [16].

In this research, an overset grid methodology for use within a SUPG formulation for unsteady, viscous, moving boundary simulations has been developed and verified. In addition, a parallel overset grid assembly framework has been developed, and implemented for 3D flow simulations.

In the remaining chapters, the methodology and numerical results of the overset approach for the SUPG formulation is presented. Chapter II presents the governing equations and discretization methods utilized for both space and time integration for the SUPG method. Chapter III presents the details of the overset grid methodology. Chapter IV demonstrates the 2D and 3D code verification results using the Method of Manufactured Solutions (MMS). Numerical results are presented in Chapter V, which include steady-state and time-dependent moving boundary simulations of inviscid, laminar and turbulent flows. Finally, Chapter VI offers conclusions and summarizes contributions made to the scientific community.

CHAPTER II

GOVERNING EQUATIONS AND DISCRETIZATION

II.1 Governing Equations

The conservative law form of the three-dimensional Reynolds-averaged Navier-Stokes equations may be written as

$$\frac{\partial \boldsymbol{Q}(x,t)}{\partial t} + \nabla \cdot [F_e(\boldsymbol{Q}) - F_v(\boldsymbol{Q}, \nabla \boldsymbol{Q})] = \boldsymbol{S}(\boldsymbol{Q}, \nabla \boldsymbol{Q}) \quad in\ \Omega \tag{II.1}$$

with the conservative flow variables, inviscid and viscous flux vectors defined as

$$\boldsymbol{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho \tilde{\nu} \end{pmatrix} \quad F_e^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \\ \rho u\tilde{\nu} \end{pmatrix} \quad F_e^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \\ \rho v\tilde{\nu} \end{pmatrix} \quad F_e^z = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \\ \rho w\tilde{\nu} \end{pmatrix}$$

$$F_v^x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \kappa \dfrac{\partial T}{\partial x} \\ \dfrac{1}{\rho}\mu(1+\Psi)\dfrac{\partial \tilde{\nu}}{\partial x} \end{pmatrix} \quad F_v^y = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} + \kappa \dfrac{\partial T}{\partial y} \\ \dfrac{1}{\rho}\mu(1+\Psi)\dfrac{\partial \tilde{\nu}}{\partial y} \end{pmatrix}$$

$$F_v^z = \begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + \kappa\frac{\partial T}{\partial z} \\ \frac{1}{\rho}\mu(1+\Psi)\frac{\partial \tilde{\nu}}{\partial z} \end{pmatrix} \quad \boldsymbol{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_T \end{pmatrix} \tag{II.2}$$

where $\rho, p, E, \kappa,$ and T denote the density, pressure, specific total energy per unit mass, thermal conductivity, and temperature, respectively. Furthermore, S_T is the source term and $\tilde{\nu}$ is the turbulence working variable in the modified Spalart-Allmaras model [17, 18] The pressure is related to the state variables via the ideal gas equation of state

$$p = (\gamma - 1)\left[\rho E - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right] \tag{II.3}$$

and assuming air, the ratio of specific heats γ is 1.4. The viscous stress tensor for a Newtonian fluid is given by

$$\tau_{ij} = (\mu + \mu_T)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{II.4}$$

where δ_{ij} is the Kronecker delta, μ andμ_T are the dynamic and turbulent eddy viscosity, and as usual for indicial notation the indices *i*, *j*, and *k* refer to Cartesian components. The variable Ψ has been designed to remove the effects of having a negative turbulent working variable. This variable greatly enhances the robustness of the turbulence model and is given by

$$\Psi = \begin{cases} 0.05ln(1 + e^{20\chi}) & if\ \chi \leq 10 \\ \chi & if\ \chi > 10 \end{cases} \qquad \chi = \frac{\rho\tilde{\nu}}{\mu} \tag{II.5}$$

This variable remains positive or approaches zero as the turbulence working variable becomes negative, thus preventing the instability caused by unbounded turbulent eddy viscosity.

II.2 Discretization

The discretization of the governing equations is accomplished using the Petrov-Galerkin stabilized finite element method as described in [19]. The Petrov-Galerkin method is formulated as a weighted residual method that may be written as

$$\int_{\Omega(t)} \varphi \left\{ \frac{\partial \boldsymbol{Q}(x,t)}{\partial t} + \nabla \cdot [F_e(\boldsymbol{Q}) - F_v(\boldsymbol{Q}, \nabla \boldsymbol{Q})] - \boldsymbol{S}(\boldsymbol{Q}, \nabla \boldsymbol{Q}) \right\} d\Omega(t) = 0 \qquad \text{(II.6)}$$

where φ is a weighting function, and $\Omega(t)$ indicates that for moving domains the spatial integration is a function of time. In the Petrov-Galerkin formulation, field variables are assumed continuous across element boundaries. Therefore, single-valued data is stored at the nodes of the elements and the solution is assumed to vary within each element according to a linear combination of basis functions.

The time-varying spatial integration for higher-order finite element discretization on moving domains is somewhat problematic, and differs significantly from procedures typically utilized within finite volume implementations. This is due to the fact that the conservation equations are satisfied in a weighted integral sense, and the usual constraint that is placed on the mesh velocities (i.e., the geometric conservation law (GCL) that requires the numerical approach to be able to recover a constant or uniform flow state) is no longer uniquely defined over the set of weighting functions. To this end, other approaches have typically been utilized for arbitrary Lagrangian-Eulerian (ALE) finite element formulations, each with advantages and disadvantages. The first approach is a fully conservative space-time formulation [20-22]. In space-time discretization, instead of a semi-discrete approximation, the basis functions are now dependent on both space and time, and the integration is then performed over the space-time slab. This formulation automatically satisfies the geometric conservation law, however, at a greater

computational expense. The integration of the conservation equations is now over the (d+1) dimensional space-time slab, and additionally incurs twice the number of degrees-of-freedom for the same temporal order as compared with a semi-discrete formulation. This is due to the discontinuous in time nature of the space-time approach. The second approach utilizes the kinematic description of the continuum [23, 24], and is commonly found in structural mechanics applications with finite deformations. Similar to the transformation of the conservation equations between material (Lagrangian) and spatial (Eulerian) coordinates, the transformation to another set of coordinates (e.g., ALE) is possible. Hence, the governing equations are related to each other in the various descriptions through a mapping and, therefore, the deformation gradient tensor and its determinant are required. Furthermore, a GCL condition must additionally be satisfied in order to obtain a constant solution in the physical domain. Inherent in this formulation is the requirement that an explicit mapping be provided between the reference and the physical domains in order to obtain the deformation gradient tensor (i.e., the mapping derivatives) and its determinant. Unfortunately, in moving boundary problems, where the volume mesh is being adapted to reflect changes in boundary displacements, explicit or analytical mappings are usually not possible. Thus, only the mesh positions are known at the current time iteration and not the functional form of its time dependence. With this scenario in mind, using a semi-discrete discontinuous Galerkin formulation, Mavriplis and Nastase [25] attempted to derive expressions by which the mesh velocities could be determined such that the resulting scheme was both GCL compliant and retained the design order of temporal accuracy. The approach of [25] was to compare the discrete GCL statement derived for a semi-discrete formulation to the equivalent expression in a space-time formulation which is known to automatically satisfy the GCL. Integrals involving the mesh velocities could be identified and determined, but the actual mesh velocities could not. In the

current work a semi-discrete formulation is implemented and a similar approach to that of [25] could be adopted.

The Reynolds transport theorem for a moving domain (or multidimensional Leibniz integration rule) expresses that the material derivative of the integral may be written as

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \varphi \boldsymbol{Q} d\Omega(t) = \int_{\Omega(t)} \frac{\partial(\varphi \boldsymbol{Q})}{\partial t} d\Omega(t) + \oint_{\Gamma(t)} \varphi \boldsymbol{Q} \boldsymbol{V}_g \cdot \hat{\boldsymbol{n}} d\Gamma(t) \tag{II.7}$$

where $\boldsymbol{V}_g = \boldsymbol{x}_t$ and $\hat{\boldsymbol{n}}(\boldsymbol{x}, t) = \left(\hat{\boldsymbol{n}}_x, \hat{\boldsymbol{n}}_y, \hat{\boldsymbol{n}}_z\right)$ represent the mesh velocity and the outward pointing normal to the domain boundaries, respectively. Utilizing the divergence theorem and the product rule Eq. (II.7) may be recast as

$$\int_{\Omega(t)} \varphi \frac{\partial \boldsymbol{Q}}{\partial t} d\Omega(t) = \frac{\partial}{\partial t}\int_{\Omega(t)} \varphi \boldsymbol{Q} d\Omega(t) - \int_{\Omega(t)} \varphi \nabla \cdot \left(\boldsymbol{Q} \boldsymbol{V}_g\right) d\Omega(t) - \int_{\Omega(t)} \boldsymbol{Q}\left[\frac{\partial \varphi}{\partial t} + \boldsymbol{V}_g \cdot \nabla \varphi\right] d\Omega(t) \tag{II.8}$$

The term in brackets in the last integral represents the substantial derivative of the weighting function. This term is identically zero for the basis functions since they move in time with the mesh. Stated another way, these functions are independent of time in the Lagrangian coordinates. On substitution into Eq. (II.8) yields

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \varphi \boldsymbol{Q} d\Omega(t) + \int_{\Omega(t)} \varphi \nabla \cdot [\bar{F}_e(\boldsymbol{Q}) - F_v(\boldsymbol{Q}, \nabla \boldsymbol{Q})] d\Omega(t) - \int_{\Omega(t)} \varphi \boldsymbol{S}(\boldsymbol{Q}, \nabla \boldsymbol{Q}) d\Omega = 0 \tag{II.9}$$

where the convective fluxes for dynamic grids are now written as

$$\overline{F}_e = F_e - \boldsymbol{Q}\boldsymbol{V}_g$$

$$\overline{F}_e^x = \begin{pmatrix} \rho(u - x_t) \\ \rho u(u - x_t) + p \\ \rho v(u - x_t) \\ \rho w(u - x_t) \\ (\rho E + p)(u - x_t) + p x_t \\ \rho \tilde{v}(u - x_t) \end{pmatrix} \quad \overline{F}_e^y = \begin{pmatrix} \rho(v - y_t) \\ \rho u(v - y_t) \\ \rho v(v - y_t) + p \\ \rho w(v - y_t) \\ (\rho E + p)(v - y_t) + p y_t \\ \rho \tilde{v}(v - y_t) \end{pmatrix}$$

$$\overline{F}_e^z = \begin{pmatrix} \rho(w - z_t) \\ \rho u(w - z_t) \\ \rho v(w - z_t) \\ \rho w(w - z_t) + p \\ (\rho E + p)(w - z_t) + p z_t \\ \rho \tilde{v}(w - z_t) \end{pmatrix} \tag{II.10}$$

For the results shown in the current work, the mesh moves as a rigid body with the geometry and thus the volume of the elements remain constant. This is the typical approach used in numerical studies of store-separation sequences or launch vehicle stage-separation events. Furthermore, since the motion of these bodies is prescribed, the mesh positions and grid velocities are analytic functions of time. In this case, for semi-discrete formulations, the GCL is no longer necessary. Additionally with this analytically prescribed motion, methods based on the kinematic descriptions of the continuum discussed above may be utilized. On the other hand, elastically deforming bodies requires general mesh motion and would indeed need enforcement of the GCL.

The weighting function for Petrov-Galerkin methods consists of two parts. Similar to a standard Galerkin discretization the first part is the same basis functions used to represent the primary variables over the element. The second contribution to the weighting function is a stabilizing term that provides dissipation along preferential directions to eliminate odd-even point

decoupling that often occurs with the standard Galerkin scheme. In the present work, the Streamline/Upwind Petrov-Galerkin (SUPG) method is used in defining the weighting function as

$$\varphi = N_i[I] + [P] = N_i[I] + \left(\nabla N_i \cdot \overline{A}\right)[\tau], \quad i = 1 \cdots n \tag{II.11}$$

where n is the number of Lagrangian basis function for an element, N_i are the i^{th} Lagrangian basis function for the element, $[I]$ is the identity matrix with the dimension of the flow variables, $\overline{A}$ is the linearization of the convective fluxes for dynamic grids given in Eq. (II.10), and $[\tau]$ is obtained using the following definition

$$[\tau]^{-1} = \sum_{i=1}^{n} \left|\nabla N_i \cdot \overline{A}\right| + [V] \qquad \left|\nabla N_i \cdot \overline{A}\right| = [T]|\Lambda|[T]^{-1} \tag{II.12}$$

Here, $[T]$ and $|\Lambda|$ are the right eigenvectors and absolute value of eigenvalues, respectively, of the matrix on the left hand side. For viscous flows, $[V]$ represents the linearized contributions from the viscous terms that are added to the stabilization matrix to maintain the order of accuracy as the Reynolds number is decreased and viscous terms become dominant, and has been found to be critical for robust and accurate simulations. Note that many alternative stabilization matrices can be derived using flux functions often used in finite volume schemes. Specially, flux functions such as flux-vector splitting can be written as a sum of contributions F^+ and F^- whose eigensystems have positive and negative eigenvalues, respectively. Using these definitions, the absolute value matrix $\left|\nabla N_i \cdot \overline{A}\right|$ in Eq. (II.12) can be replaced by the difference of the positive and negative linearized flux matrices. Advantages of this approach are that differentiability, positivity, and total enthalpy conservation can be maintained [26, 27].

The preceding equations may be cast into an isoparametric space using the Jacobian matrix of transformation $J(\boldsymbol{\xi})$for each element. In this formulation, the element geometry $\boldsymbol{x}$ is transformed

using the same basis functions, expressed in natural coordinate $\boldsymbol{\xi}$, as those utilized for the primary variables, namely

$$\boldsymbol{x} = \sum_{i=1}^{n} N_i(\boldsymbol{\xi})\boldsymbol{x}_i \qquad J(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} \tag{II.13}$$

where $\boldsymbol{x}_i$ is the global coordinate corresponding to the i^{th} basis function of the element.

The gradient operators and integrals transform according to $\nabla = J^{-1}\nabla_{\xi}$ and $d\Omega = |J|d\widetilde{\Omega}$, respectively. Here, $\left|J\right|$ denotes the determinant of the Jacobian matrix of transformation. For an isoparametric element, using integration by parts, the resulting weak statement of Eq. (II.9) may be written as

$$\frac{\partial}{\partial t}\int_{\widetilde{\Omega}} N_i \boldsymbol{Q}|J|d\widetilde{\Omega} - \int_{\widetilde{\Omega}} J^{-1}\nabla_{\xi} N_i \cdot \left(\overline{F}_e - F_v\right)|J|d\widetilde{\Omega} + \oint_{\widetilde{\Gamma}} N_i\left(\overline{F}_e - F_v\right)\cdot \hat{\boldsymbol{n}}|J^*|d\widetilde{\Gamma} - \int_{\widetilde{\Omega}} N_i \boldsymbol{S}|J|d\widetilde{\Omega} +$$

$$\frac{\partial}{\partial t}\int_{\widetilde{\Omega}}[P]\boldsymbol{Q}|J|d\widetilde{\Omega} + \int_{\widetilde{\Omega}}[P]\left[J^{-1}\nabla_{\xi}\cdot\left(\overline{F}_e - F_v\right) - \boldsymbol{S}\right]|J|d\widehat{\Omega} = 0 \tag{II.14}$$

where J^* corresponds to the Jacobian of the transformation for element boundaries. Note that because the field variables are assumed to vary continuously in the interior of the domain, the surface integral vanishes on the boundaries of the interior elements and needs only be evaluated on the physical boundaries of the domain. On these domain boundaries, appropriate boundary conditions are strongly enforced by incorporating them into the surface integral. Under the assumption that the mesh moves as a rigid body with the geometry, the determinant of the Jacobian matrix of transformation is constant. Additionally, the grid velocities and mesh positions are analytically prescribed. Therefore, standard temporal discretization may be utilized for Eq. (II.14). In the current work, the BDF2 time integration scheme is used.

CHAPTER III

OVERSET METHDOLOGY

III.1 Overset Boundary Condition and Discretization

When multiple grids overlap, each will have faces that require data from another grid in order to complete the discretization. In finite volume implementations, nodes at which this data is interpolated are referred to as fringe points and a stencil from a donor grid must be found such that bilinear in two-dimensions (trilinear in three-dimensions) or higher interpolation may be accomplished. These fringe points are then used to supply the needed data in order to evaluate the fluxes through these overlapped faces using a discretization scheme consistent with interior faces. In a finite element implementation, this data may be found directly from the donor cell basis functions. Furthermore, the consistent discretization of the fluxes across the faces in the overset region naturally appears in the element equations. That is, the boundary integral in Eq. (II.14) that typically vanishes on the faces of interior elements for a Petrov-Galerkin method must now be evaluated, and constitutes a flux boundary condition. Additionally, unlike finite volume implementation, as long as all faces on overset boundaries lie within the cells of donor grids, no orphan points would exist. For planar interface, this would simplify to a patched-grid boundary.

As an example shown in Figure III.1, the convective flux on the overlapped faces may be viewed as a Riemann problem, and determined from any approximate Riemann solver. In the current work, the convective flux is split into contributions

$$\overline{F_e} = \overline{F_e}^{+}(\boldsymbol{Q}_L) + \overline{F_e}^{-}(\boldsymbol{Q}_R) \tag{III.1}$$

where $\boldsymbol{Q}_L$ and $\boldsymbol{Q}_R$ represent the left and right states, and the extension of the van Leer flux-vector splitting [28, 29] scheme for dynamic meshes [30] is utilized. For van Leer flux vector splitting, when the Mach number normal to the face is greater than unity (i.e. $M_n > 1$), the convective fluxes as given in Eq. (II.10) are utilized based on the appropriate upwind state. For flow where the magnitude of the normal Mach number is less than unity, the van Leer fluxes are given by

$$\overline{F_e}^{\pm} \cdot \hat{\boldsymbol{n}} = \begin{pmatrix} f_{mass}^{\pm} \\ f_{mass}^{\pm}\left(\dfrac{-\overline{U} \pm 2a}{\gamma}\hat{\boldsymbol{n}}_x + u\right) \\ f_{mass}^{\pm}\left(\dfrac{-\overline{U} \pm 2a}{\gamma}\hat{\boldsymbol{n}}_y + v\right) \\ f_{mass}^{\pm}\left(\dfrac{-\overline{U} \pm 2a}{\gamma}\hat{\boldsymbol{n}}_z + w\right) \\ f_{mass}^{\pm}\left(\dfrac{-(\gamma-1)\overline{U}^2 \pm 2(\gamma-1)\overline{U}a + 2a^2}{\gamma^2-1} + \dfrac{u^2+v^2+w^2}{2} + \dfrac{-\overline{U} \pm 2a}{\gamma}\boldsymbol{V}_g \cdot \hat{\boldsymbol{n}}\right) \\ f_{mass}^{\pm}\tilde{\nu} \end{pmatrix}$$

$$f_{mass}^{\pm} = \pm\rho a\,(M_n \pm 1)^2/4 \tag{III.2}$$

with a denoting the speed of sound, and the following definitions

$$M_n = \frac{\overline{U}}{a} \qquad \overline{U} = \left[(u, v, w)^T - \boldsymbol{V}_g\right] \cdot \hat{\boldsymbol{n}} \tag{III.3}$$

For stationary meshes the mesh velocity is zero, and the above reduces to the standard van Leer fluxes. The diffusive fluxes appearing in the boundary integral of Eq. (II.14) is found by averaging the contributions from the left and right states as

$$F_v = \frac{1}{2}\left(F_v(\boldsymbol{Q}_L, \nabla\boldsymbol{Q}_L) + F_v(\boldsymbol{Q}_R, \nabla\boldsymbol{Q}_R)\right) \tag{III.4}$$

where $\nabla\boldsymbol{Q}_L$ and $\nabla\boldsymbol{Q}_R$ represent the gradients of the left and right states, respectively.

When the grid moves a new set of donor cells must be determined for each host face, and the residual calculation proceeds as described above. As discussed in the following section, for implicitly treating the overset boundaries, the linearization stencil continuously changes.

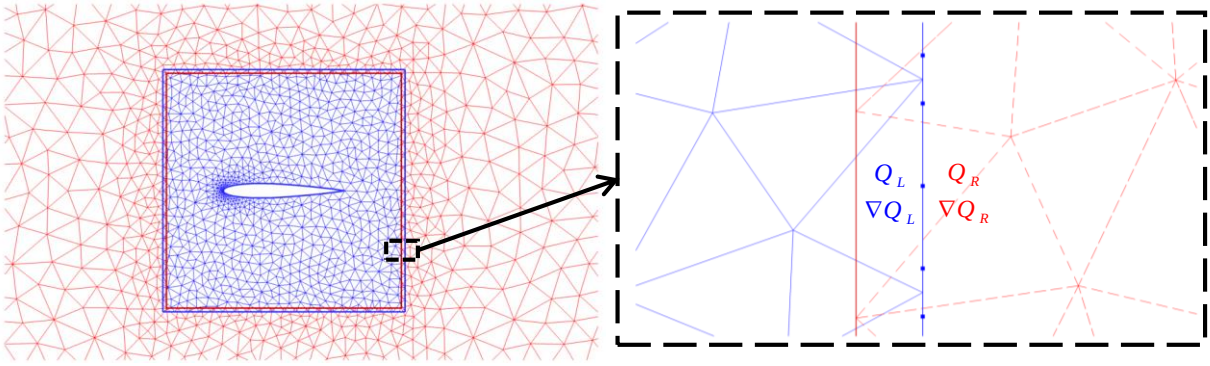


Figure III.1 Illustration of flux boundary condition for overset boundaries

III.2 Linearization

For stationary meshes the dependencies between nodal degrees of freedom of the meshes may be established during pre-processing when the donor cells are found for each quadrature point on the overset boundaries. Unfortunately, for moving meshes, this dependency changes each iteration. The compressed row storage scheme used in the current work would continuously require updating. To avoid this costly updating, the exact Jacobian matrix is split into two contributions and stored separately as $A = \tilde{A} + O$. The first contribution $\tilde{A}$ appearing on the diagonal sub-matrices of A represents the linearization without regards to the dependencies due to overset

boundaries (i.e. intra-grid dependency). Thus, the compressed row storage for this contribution does not change for moving meshes. The second contribution O appearing on the off-diagonal sub-matrices of A represents the linearization with respect to the donors from other grids at the overset boundaries (i.e. inter-grid dependency). The sparse pattern of this contribution does change from iteration to iteration for dynamic meshes. The cost of re-computing the compressed row storage of this contribution is trivial owing to its limited size. The sum of these contributions results in an exact linearization over the overset grids. As an example, the structure of the Jacobian matrix of a 16-airfoil overset grid case, whose mesh is subsequently shown in Figure III.22(d), is illustrated in Figure III.2.

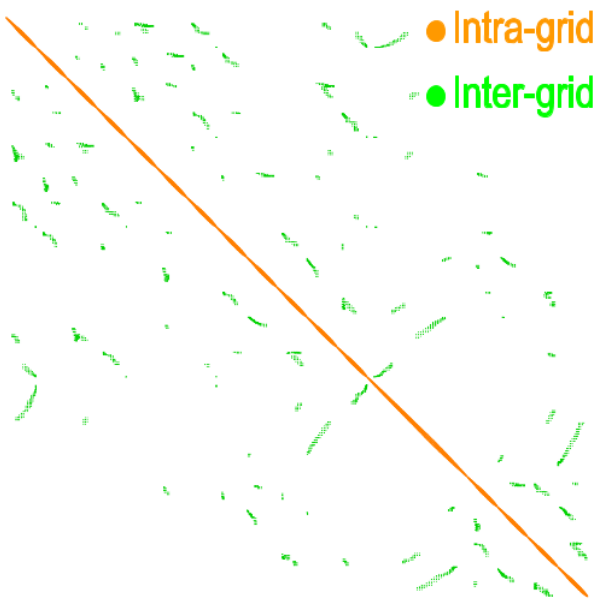


Figure III.2 Illustration of Jacobian matrix for the 16-airfoil overset grid case

At each time iteration, in order to eliminate temporal discretization errors, the time-dependent residuals are converged to a specified tolerance using discrete-Newton relaxation. In

the sub-iterations, the systems of equations are solved using a modified generalized minimal residual (GMRES) method [31] with preconditioning. Both intra-grid and inter-grid dependencies, $\tilde{A}$ and O, are utilized in matrix-vector products within GMRES, which means, overset boundaries are treated as implicit boundary conditions. For the preconditioner, there are various choices that have significant effects on the rate of convergence of the linear system. ILU(k) and two modifications to this preconditioner are evaluated in following section.

III.3 Preconditioner

With the aid of Figure III.3, three preconditioners are considered here. The first preconditioner is the incomplete LU decomposition of the intra-grid linearization $\tilde{A}$, and is referred to as "Intra-grid ILU(k)" in this section. Inter-grid linearization O is ignored in the intra-grid ILU(k) because its large bandwidth makes incomplete LU decomposition prohibitively expensive. As the number of overset grids used in the simulation increases, the portion of inter-grid linearization O with respect to the complete linearization A also increases; therefore, ignoring inter-grid linearization O in the preconditioner may ultimately slow down the convergence of GMRES. However, this preconditioner has been used successfully by Galbraith [15]. In order to investigate the importance for the preconditioner to include inter-grid linearization, two modifications to the intra-grid ILU(k) are constructed.

The first modification includes three variants: $L(U + L^{-1}O_U)$, $(L + O_L U^{-1})U$ and $(L + O_L U^{-1})(U + L^{-1}O_U)$, where O_L and O_U are the lower and upper triangular part of the inter-grid dependency O. This modification is referred to as "triangular inter-grid ILU(k)" in this research.

The second modification is utilizing the intra-grid ILU(k) inside Jacobi iterations, which results in a preconditioner that approximates $LU + O \approx A$. It is referred to as "Jacobi inter-grid ILU(k)" in this research. The implementation of the "Jacobi inter-grid ILU(k)" is described in Figure III.4.

$$A = \tilde{A} + O$$

$$\tilde{A} = \begin{bmatrix} \tilde{A}_1 & & \\ & \tilde{A}_2 & \\ & & \tilde{A}_3 \end{bmatrix} \quad \text{Intra-grid linearization}$$

$$O = \begin{bmatrix} & O_{12} & O_{13} \\ O_{21} & & O_{23} \\ O_{31} & O_{32} & \end{bmatrix} \quad \text{Inter-grid linearization}$$

$$O_L = \begin{bmatrix} & & \\ O_{21} & & \\ O_{31} & O_{32} & \end{bmatrix} \quad O_U = \begin{bmatrix} & O_{12} & O_{13} \\ & & O_{23} \\ & & \end{bmatrix}$$

$LU \approx \tilde{A}$ Incomplete LU decomposition of intra-grid linearization

$$L = \begin{bmatrix} L_1 & & \\ & L_2 & \\ & & L_3 \end{bmatrix} \quad U = \begin{bmatrix} U_1 & & \\ & U_2 & \\ & & U_3 \end{bmatrix}$$

$$L + O_L U^{-1} = \begin{bmatrix} L_1 & & \\ O_{21}{U_1}^{-1} & L_2 & \\ O_{31}{U_1}^{-1} & O_{32}{U_2}^{-1} & L_3 \end{bmatrix} \quad U + L^{-1} O_U = \begin{bmatrix} U_1 & {L_1}^{-1} O_{12} & {L_1}^{-1} O_{13} \\ & U_2 & {L_2}^{-1} O_{23} \\ & & U_3 \end{bmatrix}$$

Figure III.3 Structure of the Jacobian matrix for an arbitrary 3-overlapping-grid case

Applying preconditioner $\boldsymbol{x} = (LU + O)^{-1}\boldsymbol{b}$

by solving $(LU + O)\boldsymbol{x} = \boldsymbol{b}$ for $\boldsymbol{x}$ using Jacobi iterations

$\boldsymbol{x}^0 = 0$

do $i = 1, n$

solve $LU\boldsymbol{x}^i = b - O\boldsymbol{x}^{i-1}$ for $\boldsymbol{x}^i$

end do

$\boldsymbol{x} = \boldsymbol{x}^n$

Figure III.4 Pseudo code of the "Jacobi inter-grid ILU(k)" preconditioner

To evaluate the effectiveness of the three preconditioners, simulations are carried out for the 16-airfoil overset grid case, whose mesh is subsequently shown in Figure III.22(d), and Jacobian matrix illustrated in Figure III.3. The freestream has a Mach number of 0.2, angle-of-attack of 2°, and the flow is assumed inviscid. The CFL number ramps up from 1 to 2000 in 100 iterations. At each pseudo time iteration, GMRES is used to reduce the residuals of linear system by 10 orders of magnitude. The fill-level for the incomplete LU decomposition for the intra-grid Jacobian matrix is set to 1 in all three preconditioners. Note that for triangular inter-grid ILU(k), the variant $L(U + L^{-1}O_U)$ is used; for Jacobi inter-grid ILU(k), the number of Jacobi iterations of 3 and 5 are tested.

All the simulations have the same convergence history, as shown in Figure III.5. As shown in Figure III.6 and Figure III.7, the two modified preconditioners considerably reduce the number of GMRES search directions required. However, extra computation per GMRES search direction are needed. Compared with intra-grid ILK(k), the triangular inter-grid ILU(k) uses the least CPU time. The Jacobi inter-grid ILU(k) with 3 iteration also uses less CPU time at later stage of the

simulation. This is due to the fact that as the GMRES search directions increases, the saving in computing the Arnoldi iterations for GMRES finally overcomes the extra cost of performing an "intra-grid ILU(k)" in each Jacobi iteration. This study clearly shows the inter-grid linearization plays an important part in the efficiency of the preconditioner. For illustrative purpose, the Mach contours for this 16-airfoil case are shown in Figure III.8.

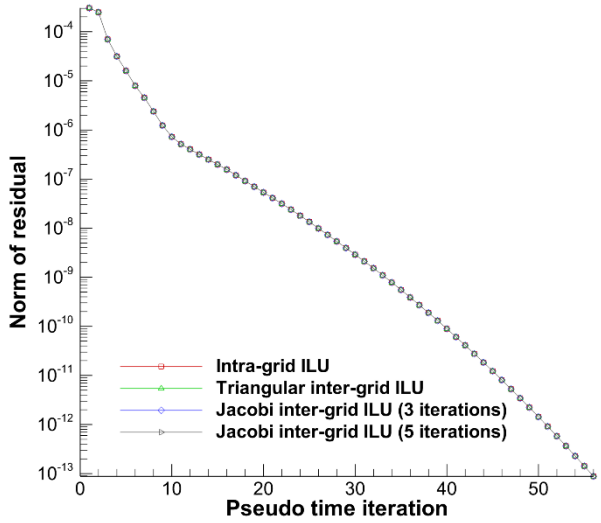


Figure III.5 Convergence histories of 16-airfoil overset grid case

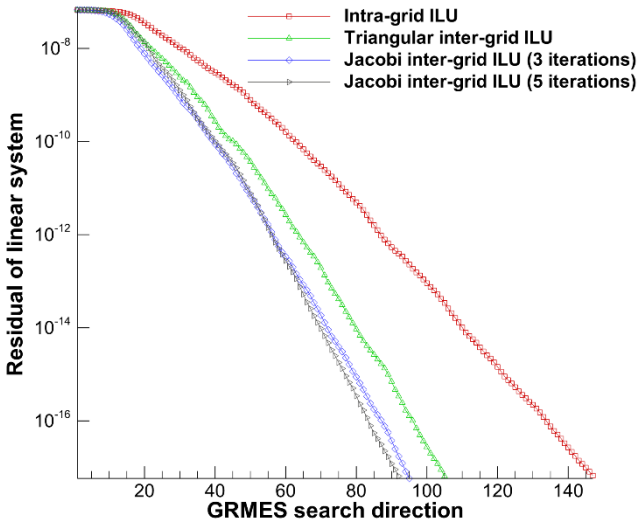
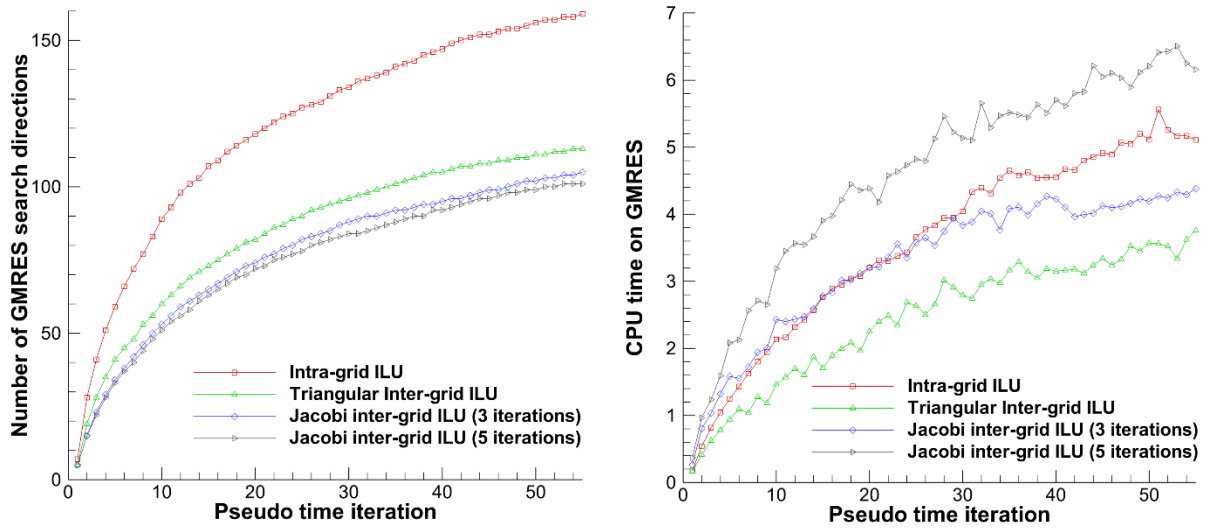


Figure III.6 Convergence histories of the linear system at pseudo time iteration 40



(a) Number of GMRES search directions (b) CPU time

Figure III.7 Number of GMRES search directions and CPU time in each pseudo time iteration

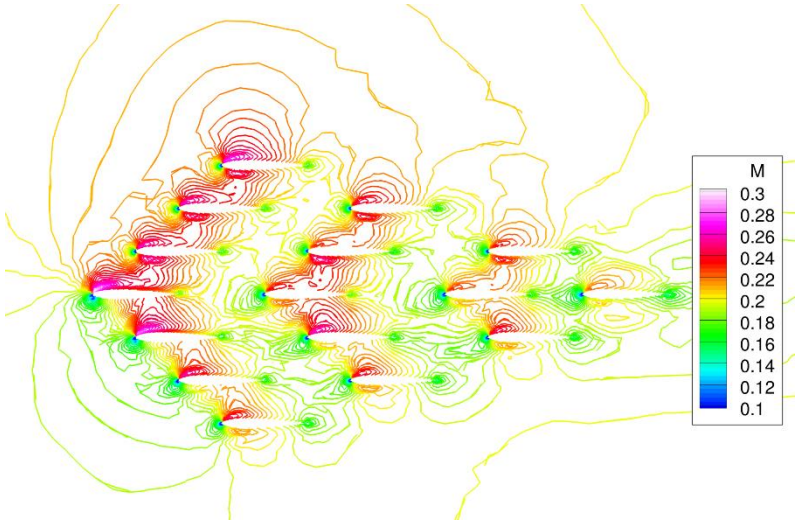


Figure III.8 Mach contours of the 16-airofoil overset grid case

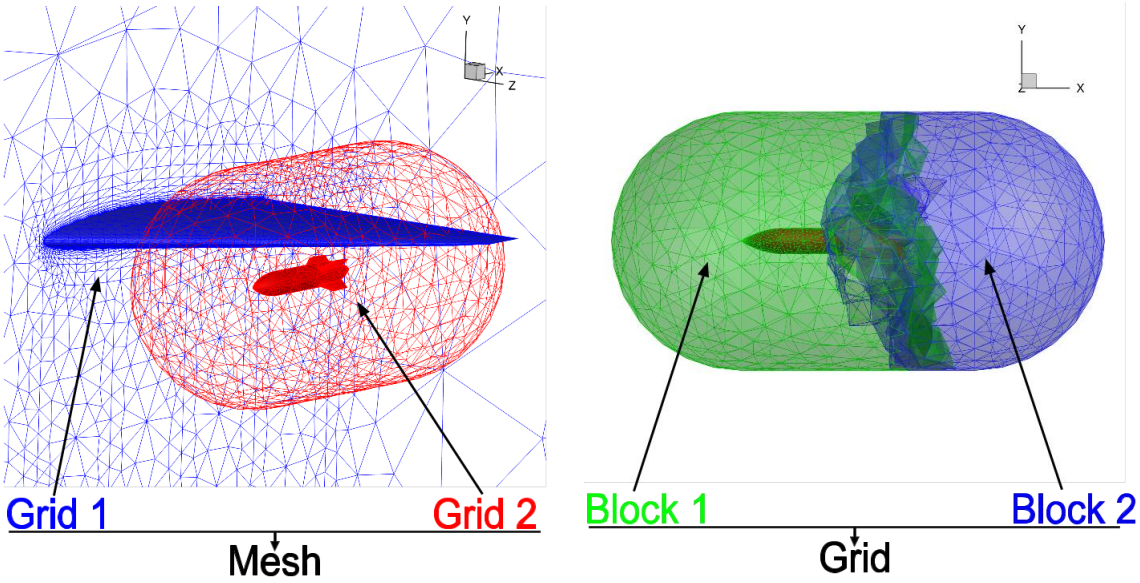
It is worth noting that, due to the similarity in the structures of the Jacobian matrices, by making the analogy between intra/inter-grid linearization and intra/inter-computational-node linearization, these two modified preconditioners can also be extended as a parallel preconditioner

for distributed memory machines, regardless whether an overset grid approach is used. While the "triangular inter-computational-node ILU(k)" preconditioner is inherently serial in nature, because of its triangular structure, the "Jacobi inter-computational-node ILU(k)" preconditioner is readily parallel.

III.4 Overset Grid Assembly

Various methods have been utilized for hole cutting within overset grid schemes [9, 10, 16, 32-35]. An overview of the history and current state of the overset gird methods has been given by Cameron [36]. The dynamic hole cutting method used in this research is conducted in two-stages. The first stage identifies invalid cells (and corresponding nodes), which are cells that intrude or are interior to a geometry. The second phase represents the cell selection process among the valid cells. A MPI-based parallel 3D overset grid assembly module following this procedure is developed.

To avoid any confusion in terminology, in this section, a "mesh" is the spatial discretization used for a simulation. For an overset simulation, a "mesh" consists of multiple "grids. As the Wing/Finned-Store (WFS) configuration shown in Figure III.9(a), the mesh contains a body fitted grid for the store, and another grid for the wing. When carrying out a simulation on distributed memory machines, after domain decomposition, a "grid" is deposed into one or multiple "blocks", as shown in Figure III.9(b); block 1 is assigned to a MPI rank, and block 2 is assigned to another MPI rank. All the "blocks" on the same MPI rank form a "partition". In this research, the overset grid assembly and the flow solver operate on the same partition.



(a) A mesh and its grids (b) A grid and its blocks

Figure III.9 Illustration of mesh, grid and block

III.4.1 Point Search Algorithm

A basic functionality needed by the overset grid assembly is a robust and efficient point search algorithm, needed to identify the cell of a specific block, which encloses a point with given coordinates. The adopted method is a "stencil walking" type of algorithm, which begins by taking an initial guess of the enclosing cell denoted as *current_cell*. Then the relative position of the point with respect to *current_cell* is calculated. If the point is outside of *current_cell*, then the neighboring cell in the correct direction with respect to *current_cell* is set to be the new *current_cell*. The procedure continues until the enclosing cell is found. The algorithm is illustrated in Figure III.10, where $\boldsymbol{\xi}$ represents element-wise natural coordinate, $\boldsymbol{x}_o$ and $\boldsymbol{\xi}_o$ are the global and natural coordinates of the point to be located, respectively. Furthermore, n is the number of basis functions of the element, and $\boldsymbol{x}_i$ is the global coordinate corresponding to the i^{th} basis function of the element.

Pick initial *current_cell*

do

Calculate $\boldsymbol{\xi}_o$ by solving $\sum_{i=1}^{n} \boldsymbol{x}_i \phi_i(\boldsymbol{\xi}) = \boldsymbol{x}_o$ for $\boldsymbol{\xi}$ using Newton's method

if $\boldsymbol{\xi}_o$ is inside of element, then

enclosing_cell = *current_cell*

else

current_cell = neighbor of *current_cell* on the direction of $\boldsymbol{\xi}_o$

end if

while (*current_cell* ≠ *enclosing_cell*)

Figure III.10 Pseudo code of stencil walking algorithm

The stencil walking algorithm starts from the initial cell, and traverses the block until it contacts the enclosing cell. However, as a scenario illustrated in Figure III.11, in the search for the target point O, if choosing location B as the initial guess, the algorithm would fail because of the gap along the traversing path. The solution to this problem is based on an intuitive fact: if a point is enclosed by a block, then there would be no gap between the point and the nearest surface of the block. Therefore, the solution is to find the surface of the block that is nearest to the target point O, which is illustrated as location A in Figure III.11, and use it as the initial guess. Note that surfaces of a block include all its boundary surfaces, as well as surfaces exposed by domain decomposition.

An efficient nearest neighbor search algorithm is needed to accelerate locating the nearest surface to a point. To this end, an octree is built for the surfaces of each block, which is based on the approach used by Taylor [37], where it was originally used for calculating point-to-wall

distance needed in turbulence modeling. Note that the octree for the surface of a block is generated and stored only on the MPI rank that owns the block.

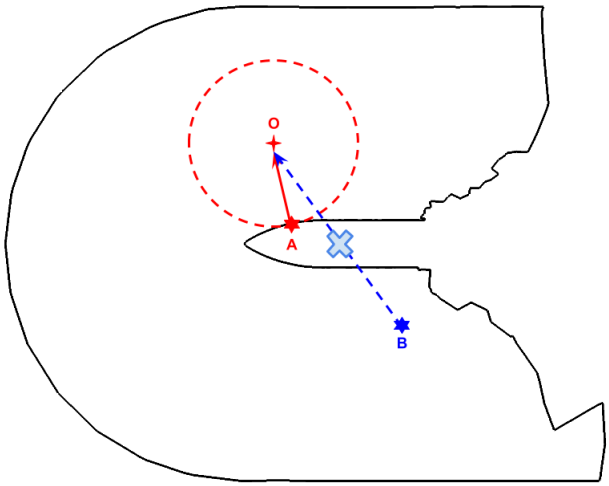


Figure III.11 Influence of different initial guesses on stencil walking algorithm

III.4.2 Block Profile

Many procedures in the overset grid assembly require identifying the location of a point. Some of them (e.g. finding the donor cell for a Gauss quadrature point on an overset boundary) need the exact point location, including the enclosing cell and the natural coordinate of the point in that cell, while others only need to know if the point is inside or outside of a specific grid or block (e.g. identifying points that are inside a geometry, determining the overlap region of two blocks, etc.). While the first group of procedures requires a point search algorithm, the latter requires a simpler algorithm.

For procedures that only require determining whether a point is inside or outside a specific grid, block or geometric entity, an axis-aligned auxiliary Cartesian mesh for each block is built, as a way to provide information of the block profile. Depending on its location (with respect to the

block and the grid that owns the block), each cell of a Cartesian mesh has one of several statuses, which include but are not limited to: “inside block”, “outside block but inside grid”, “outside grid and geometry”, “inside geometry”, “on geometric boundary”, “on non-geometric boundary”, “on decomposition surface”, etc., as illustrated in Figure III.12. In this research, the Cartesian meshes are 3D meshes in implementation. The status of every Cartesian cell is stored. However, a more memory efficient approach exists. PEAGSUS [10] uses a pseudo 3D Cartesian mesh to profile the geometry for cutting minimal hole. The Cartesian mesh is a 2D mesh in implementation, but store the 3rd dimension of the geometry as a function of the other two dimensions, in this way, memory usage is greatly reduced.

Noting that the blocks are distributed over multiple MPI ranks, as the example previously discussed for the store grid of the WFS configuration, and now shown in Figure III.13(a, b), the procedure of generating Cartesian meshes for blocks belonging to a specific grid used in this dissertation is described in the following paragraphs.

1) Lower and upper bounds of the coordinates of the grid are determined. A global Cartesian mesh enclosing this grid is generated virtually. That is, the lower and upper bounds of the coordinates, the spacing, and the index system is determined, however, the actual mesh is not generated.

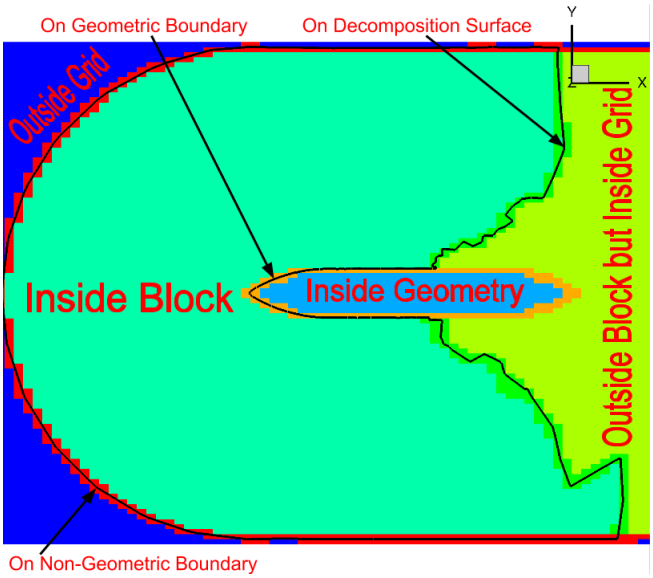
2) For each block belonging to the grid, a local Cartesian mesh, which encloses the block and is a portion of the global Cartesian mesh for the grid, is generated on each corresponding MPI rank. Note that a local Cartesian mesh may contains some of the same cells found in remote Cartesian meshes (e.g. Cartesian meshes built on remote MPI ranks for other blocks that belong to the same grid).

3) The surfaces of the block (e.g. geometric and non-geometric boundaries, decomposition surfaces) are located in the local Cartesian mesh, and the enclosing Cartesian cells are marked with corresponding status.

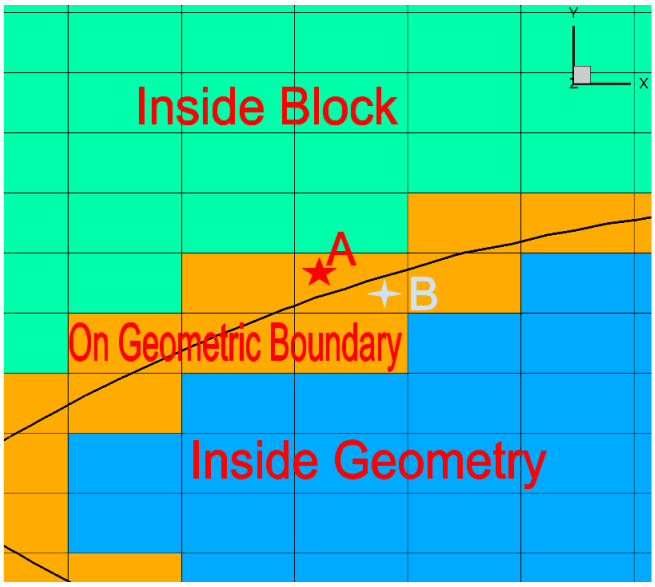
4) If a marked cell in the local Cartesian mesh is also contained by a remote Cartesian mesh, then the cell is marked in the remote Cartesian mesh accordingly. At this stage, only the cells with "on geometric/non-geometric/decomposition surface" status have been marked. All the marked Cartesian cells at this stage are depicted in Figure III.13(c, d) for the store grid example.

5) The Cartesian mesh contains two type of cells at this stage: the marked cells, and the unmarked cells, which are isolated into multiple regions by the marked cells. For each region, a seed cell is chosen, and a point search for its central point is performed in the local and remote blocks. If the central point is located, depending on whether it is located on the local or a remote block, the seed cell is marked as "inside block" or "outside block but inside grid" respectively. If the point is not located, depending on whether the region is bordered by marked cells with the status of "on geometric boundary" or "on non-geometric boundary", the seed cell is marked with the status of "inside geometry" or "outside grid and geometry" respectively. Finally, each region is marked with the same status as its seed cell. Intersecting planes of the resulting Cartesian meshes for the store grid are shown in Figure III.13(e, f).

6) After a local Cartesian mesh is built for each block on the corresponding MPI rank, portions of certain Cartesian meshes can be sent to other MPI ranks, to provide block profile on those MPI ranks if needed.

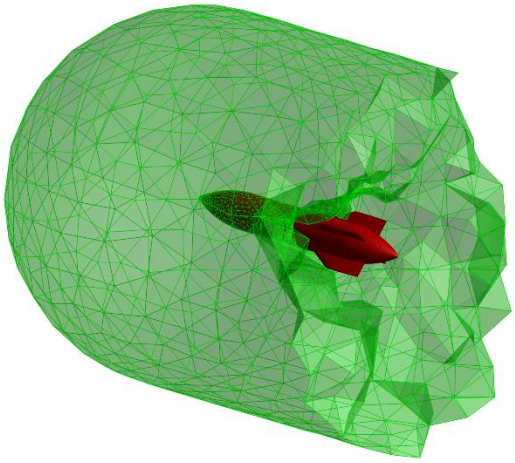


(a) Full view

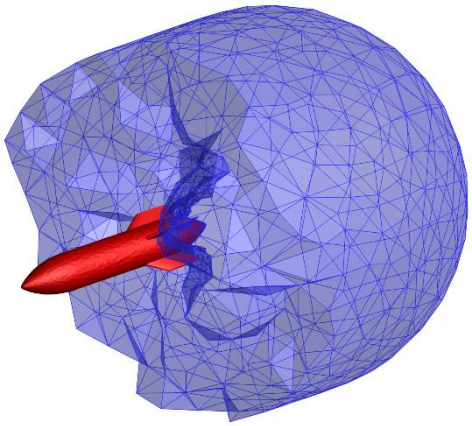


(b) Focused view

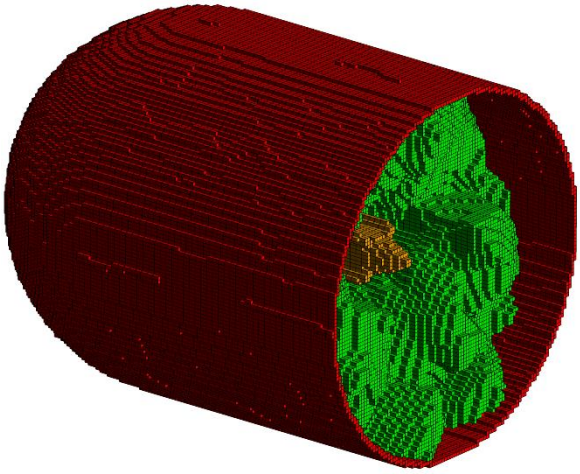
Figure III.12 Cartesian auxiliary mesh for a block for the store grid



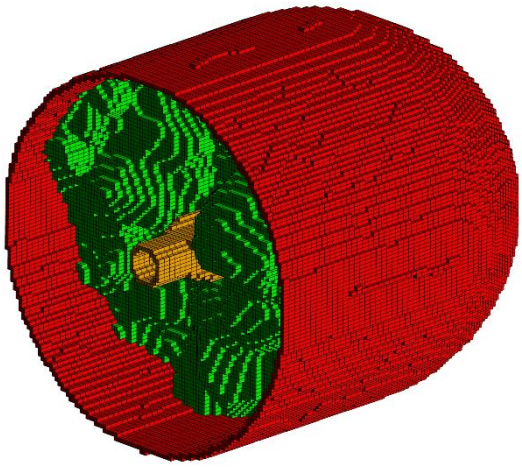
(a) Block 1 for the store grid



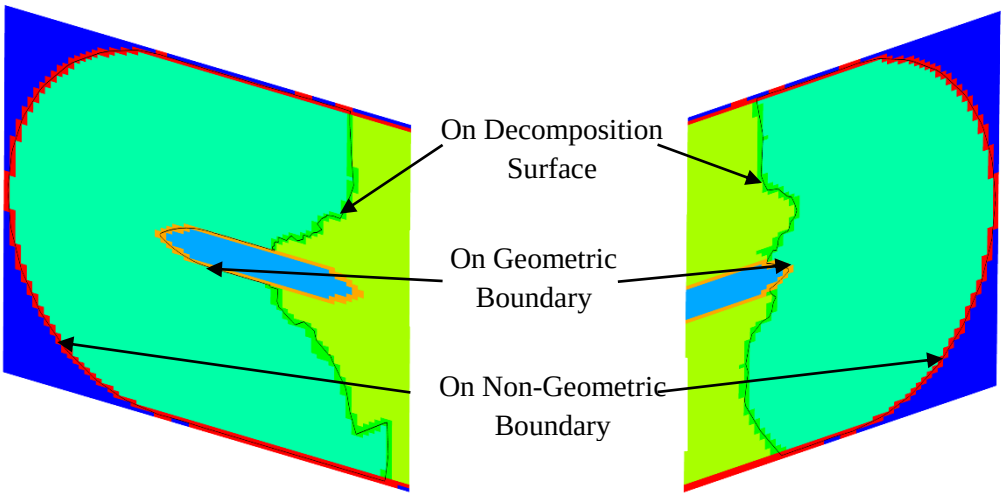
(b) Block 2 for the store grid



(c) Marked Cartesian cells for block 1



(d) Marked Cartesian cells for block 2



(e) Slice of Cartesian mesh for block 1

(f) Slice of Cartesian mesh for block 2

Figure III.13 Block profile for the store

As Beatrice and Jayanarayanan [35] note, if a point is enclosed by a Cartesian cell with a status of “on geometric surface” or “on non-geometric surface”, then whether the point is inside or outside of the corresponding grid cannot be determined by using the Cartesian mesh alone and, thus, a point search for the point is needed. If the point search is able to locate the point in the grid, then the point would be “inside the grid”; otherwise, depending on the status of the enclosing Cartesian cell is “on geometric surface” or “on non-geometric surface”, the point would be “inside the geometry” or “outside the grid and geometry”, respectively. As the example illustrated in Figure III.12(b), point A and B are enclosed by the same Cartesian cell with the status of “on geometric boundary”. A point search would be able to locate point A inside the grid, but would fail to locate point B inside the grid, which means point A is “inside the grid”, and point B is “inside the geometry”. Beatrice and Jayanarayanan [35] also note that, in order to uniquely determine the status of a point enclosed by a Cartesian cell with a status of “on geometric surface” or “on non-geometric surface”, the Cartesian cell should not have the statuses of “on geometric boundary” and “on non-geometric boundary” at the same time. The requirement can be translated into two further requirements imposed on the grid and Cartesian mesh respectively: 1) the geometric boundaries and the non-geometric boundaries of the same grid should not contact; 2) for a grid that satisfies the first requirement, the corresponding Cartesian mesh should have a resolution that is high enough, such that none of the Cartesian cells would contact a geometric surface and a non-geometric surface of the grid at the same time.

III.4.3 Identify Invalid Cells

Invalid cells refer to the cells that are completely or partially inside of geometric components, and nodes contained by the invalid cells referred to as invalid nodes. Invalid cells and nodes need to be identified and excluded from computational domain.

Invalid cells and nodes can be identified by using the geometric boundaries to perform a direct cut on the grid. The approach is explained by the 2D example depicted in Figure III.14. In this example, geometric boundary node 2 would be identified to be enclosed by cell A by a point search, and therefore cell A would be considered invalid and added to the list of invalid cells, and the edges of cell A that intersect the geometric boundaries are added to the list of intersected edges. Once cell A has been identified, its neighboring cells (cell B, C, and D) are visited to determine if any edge of these cells intersects the geometric boundaries as well and, if so, the corresponding cells and intersected edges would also be added to the list of invalid cells and the list of intersected edges, respectively. This process is repeated until no additional invalid cell or intersected edge is identified. Then for each intersected edge (shown in blue dashes), each of its nodes is identified as inside or outside of the geometric component, by using the information of the surface normal of the corresponding intersecting geometric boundary and the location of intersecting point on the edge, and those that are inside of the geometric component are marked as invalid nodes (shown in blue dots) and added to the list of invalid nodes. Subsequently, nodes that are connected to the invalid nodes by non-intersected edges are identified (shown in red dots) and added to the list of invalid nodes, and this process is repeated until no additional invalid node is identified.

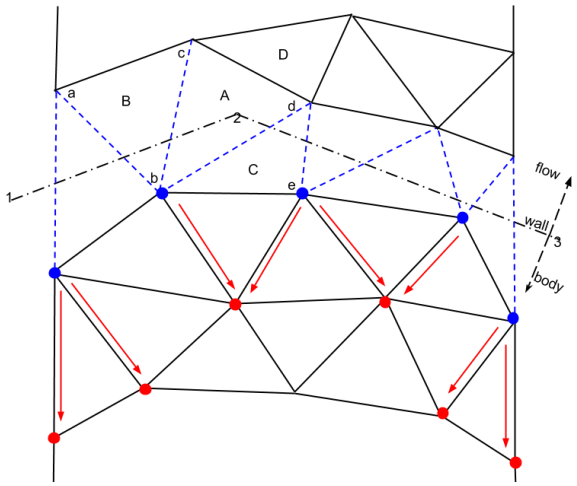


Figure III.14 Identification of invalid nodes/cells using direct cut

Direct cut is able to identify all the invalid cells and nodes precisely [38], but 3D direct cut on arbitrary meshes requires a more complicated implementation than the one illustrated in the 2D example, and therefore is not used in the 3D overset grid assembly in this research. In this research, nodes inside of geometric components are identified by using auxiliary Cartesian meshes combined with point searches as described in section III.4.2, and considered to be invalid nodes. Cells that contain any of these invalid nodes are considered to be invalid cells. Note for a cell that is partially inside of a geometric component, it is possible that none of its containing grid nodes is inside the geometric component. Therefore, some invalid cells or nodes might not be identified by this approach in this situation. However, although possible, this situation has not appeared in any test case in this research. If this situation was to occur, a direct cut would be needed to precisely identify all the invalid cells and nodes.

III.4.4 Cell Selection Methods

After the invalid cells have been identified and removed from the computational domain, there may still exist some regions with excessive overlap between grids. A cell selection procedure must then be performed in order to minimize the extent of overlap. This reduction in the overlap region removes unnecessary computation from the flow solver.

Three types of cell selections are discussed in this section: an existing method that uses a distance function, an existing Implicit Hole Cutting (IHC) method, a newly proposed modified IHC, and a proposed method referred to herein as Elliptic Hole Cutting (EHC).

III.4.4.1 Cell Selection Using Distance Function

Regions in the vicinity of geometries are places where many detailed flow phenomena take place, and body fitted grids usually have high quality in these regions. Therefore, a continuous selection of cells from these body fitted grids is preferred. Cell selection using a distance function is based on the notion that, in the region close to a particular geometry, the body fitted grid for the geometry will have a lower value of distance to the geometry than to any other geometries. For a node in the overlapping region with multiple body fitted grids, the nearest geometry is determined, and the node is selected only if it belongs to the body fitted grid for the geometry. The effectiveness of this cell selection method will be demonstrated by an example presented in section V.2.1.

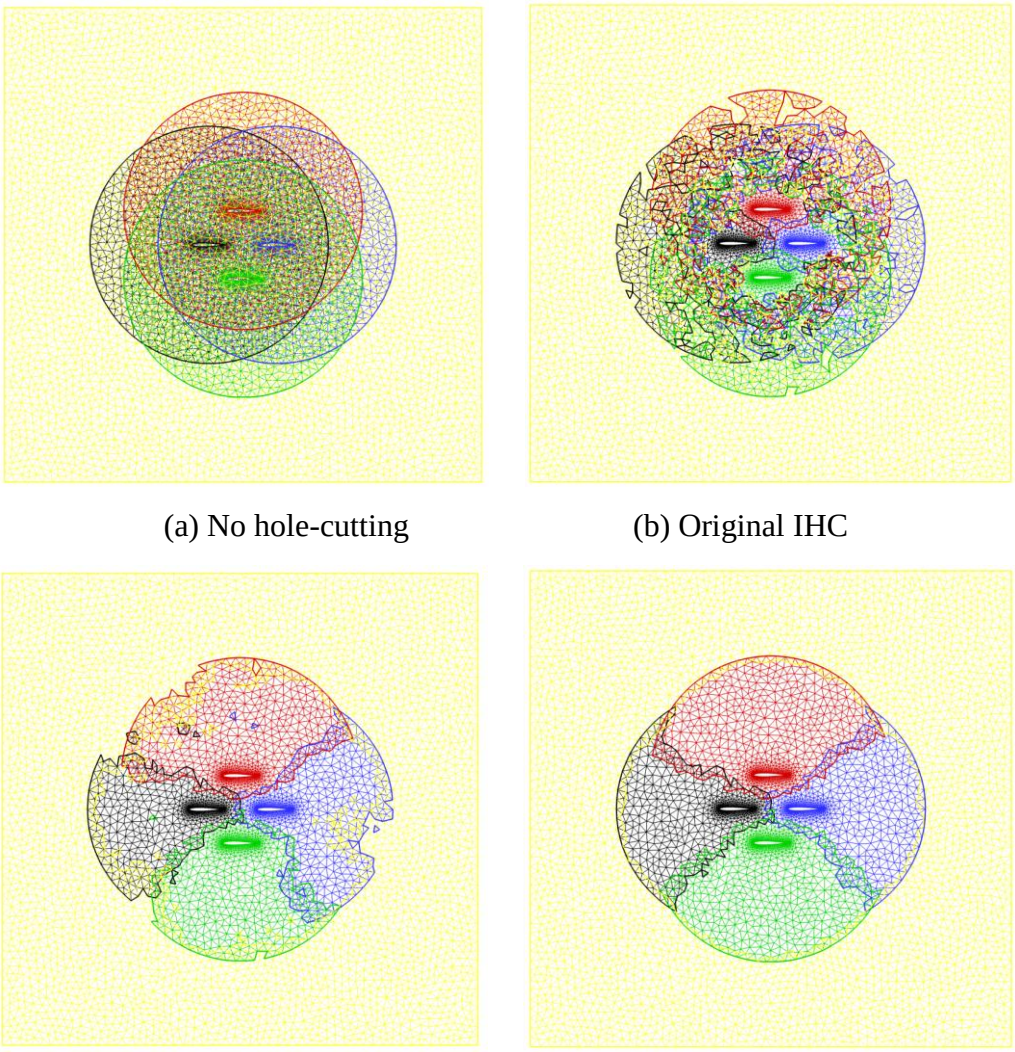
III.4.4.2 Cell Selection Using Implicit Hole Cutting

Denoting a sampling point as any valid mesh node or a quadrature point on an overset boundary, the original Implicit Hole Cutting (IHC) method [34] determines all potential donor

cells from the various overset grids that enclose this sampling point. Among the potential donor cells and the cell from which the sampling point originates, only the cell with the highest *cell_quality* is kept, and the remaining cells are blanked-out. *cell_quality* is a user-defined metric such as the inverse of cell volume, aspect ratio, etc. The most promising feature of this cell selection technique is its simplicity. Many hole cutting algorithms need the user to specify the priority of each overset grid to be selected. This priority can become difficult to specify when numerous grids reside in the same region. IHC does not need specification of such priority.

Unfortunately, the simplicity of this method may produce unwanted results. Specifically, the cells selected may become randomly distributed between the various overset grids and (i.e., the final grids are not "continuous"), therefore, reduces the efficiency of solving the system of equations. An example of this problem with original IHC is depicted in Figure III.15. The body fitted grids for four airfoils overlapping a background grid is shown in Figure III.15(a). The four airfoil grids are identical to each other, with cell volume from 10^{-5} on the wall boundaries to 10^{-2} on the outer boundaries. The background grid has a uniform cell volume of 10^{-2}. Defining the *cell_quality* as the inverse of the cell volume, the cell selection approach successfully chooses the cells from the nearest airfoil grid as illustrated in Figure III.15(b). However, in the regions overlapped by multiple grids away from the airfoil bodies, the cells are selected more randomly, because the *cell_quality* from different grids are very close. Considering the impact on solution efficiency with such grids, the cell selection process has been modified to favor selecting cells from the same grid within a local scope. To this end, an iterative process is developed to give a higher *grid_priority_factor* at each sampling point to the grid that has been selected by more nearby sampling points, and the potential donor cell with the highest *grid_priority_factor*cell_quality* is kept for this sampling point. Utilizing this improved cell

selection method, Figure III.15(c, d) show the overset grids after the first iteration and the final mesh, respectively.



(a) No hole-cutting (b) Original IHC

(c) Modified IHC, after first iteration (d) Modified IHC, final mesh

Figure III.15 Hole-cutting using original and modified IHC

III.4.4.3 Elliptic Hole Cutting

The implicit hole cutting method, using the modification discussed above to favor mesh continuity, may be utilized to construct overset grids that are more amenable to efficient solution

of the systems of governing equations. As an alternative to this technique, a new method based on the solutions to Poisson equation over each overset grid, referred to as elliptic hole cutting (EHC), is introduced. In this method, a Poisson equation is solved on each grid and may be written as

$$\nabla^2 T + s = 0 \tag{III.5}$$

where, due to similarity to the equation governing heat conduction, T is referred to as the pseudo-temperature and s is the source term. The pseudo-temperature is used as the criterion for cell selection. That is, at each sampling point, only the potential donor cell with the highest pseudo-temperature is kept. For computational efficiency, the Poisson equation is solved using a node-centered, first-order finite volume scheme. To solve this elliptic equation, the boundary conditions and source term must be specified on each grid.

To illustrate this method, the body fitted grids for two airfoils are utilized and are shown in Figure III.16 after the invalid cells have been removed. Dirichlet boundary conditions are specified for invalid nodes, and for nodes in non-overlapping regions. Although arbitrary, invalid nodes are set to a value of -1, whereas nodes in non-overlapping regions are specified as +1. The overset boundaries are treated as Neumann boundary conditions. Using the heat equation analogy, these are specified as adiabatic wall boundary conditions $\nabla T \cdot \hat{\boldsymbol{n}} = T_n = 0$. Figure III.17 illustrates the specified boundary conditions for the two airfoil example. The green areas are computational domains for the Poisson equations; the red and blue areas are imposed with Dirichlet boundary conditions $T = 1$, and $T = -1$ respectively, and the overset boundary of grid 2 is specified as adiabatic wall. Many possibilities exist for the choices of the source terms. In the current work two choices are explored. The first uses the source term to favor *cell_quality* (in the current work, it is the inverse of the cell volume), and is given by

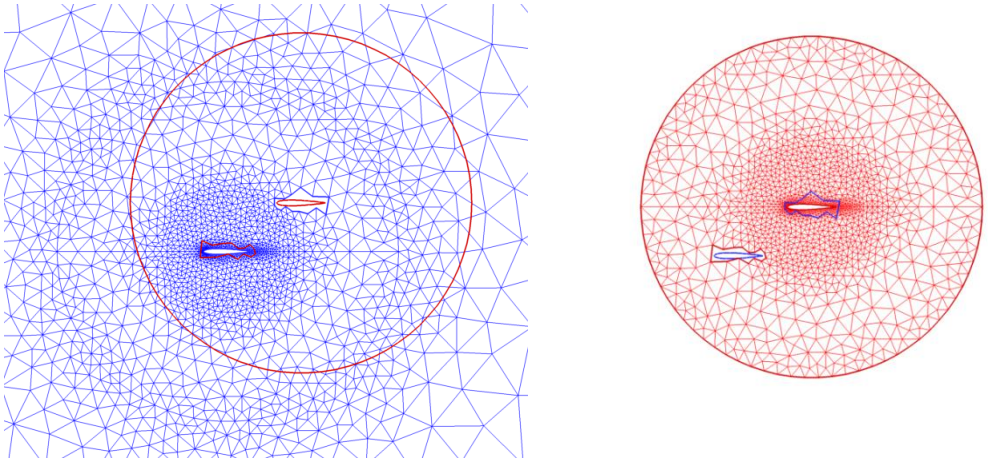
$$s = s_{\text{global\_min}} + \frac{c - c_{\text{local\_min}}}{c_{\text{local\_max}} - c_{\text{local\_min}}}\left(s_{\text{global\_max}} - s_{\text{global\_min}}\right)$$

$$s_{\text{global\_max}} = \frac{\alpha}{A} \qquad s_{\text{global\_min}} = -\frac{\alpha}{A} \tag{III.6}$$

where c represents the *cell_quality* of the sampling point (i.e., average of the *cell_quality* of the cells from which this sampling point originates), $c_{\text{local\_min}}$ and $c_{\text{local\_max}}$ are the minimal and maximal *cell_quality* of the sampling point as well as all the potential donor cells. A is a characteristic area defined as the average of the areas of all grids. α is a constant chosen by the user and assumed as 4 in current work. Another possibility is to specify the source term to favor specific grids, and is given by

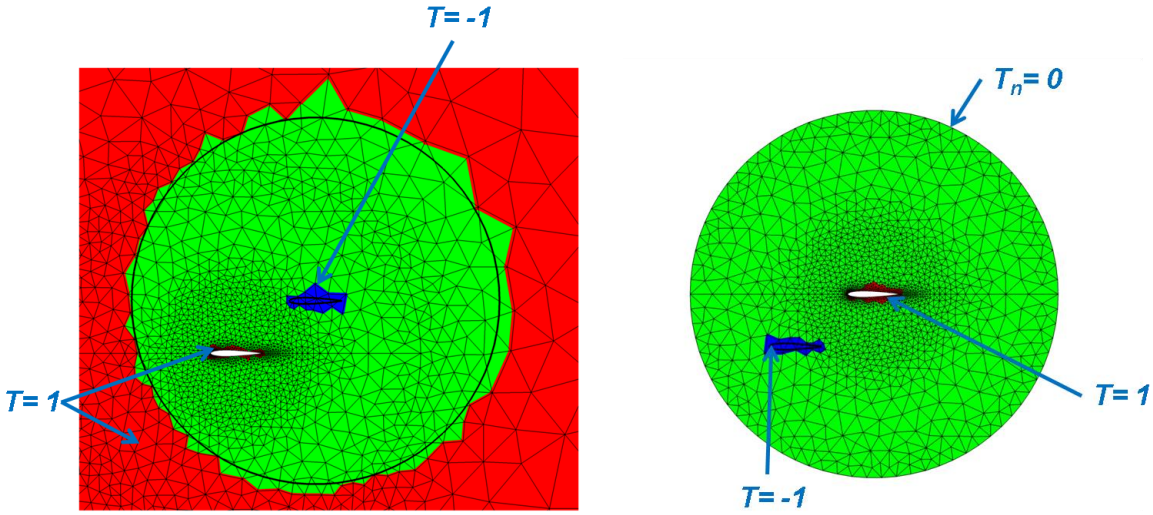
$$s = \begin{cases} s_{\max} & \text{for prefered grids} \\ s_{\min} & \text{for other grids} \end{cases} \tag{III.7}$$

In the current work, the values of $s_{\min}$ and $s_{\max}$ are left for the user to specify. The source terms, in favor of *cell_quality*, are shown in Figure III.18 for each grid used in the current demonstration. Using the elliptic hole cutting method, on each grid, the solution to Poisson equation is shown in Figure III.19. Figure III.19(a, b) depict a three-dimensional view, where the pseudo-temperature is the third dimension for this two-dimensional example, illustrating the solution using the aforementioned boundary conditions and source terms. To illustrate the cell selection process from the solutions to the Poisson equation, Figure III.19(c) presents an overlay of the solutions on the final overset mesh. As a comparison, Figure III.20(a-c) shows the overset grids produced for the current example with the original implicit hole cutting method, the modified implicit hole cutting method, and the elliptic hole cutting method, respectively. As seen, the elliptic hole cutting method generates overset grids with a higher degree of continuity within each grid.



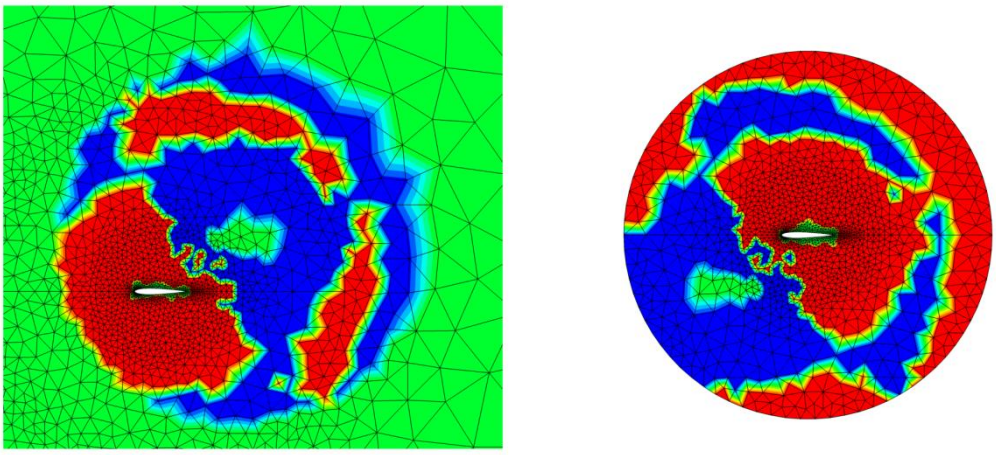
(a) Grid 1 (b) Grid 2

Figure III.16 2-airfoil overset grids after removing invalid cells



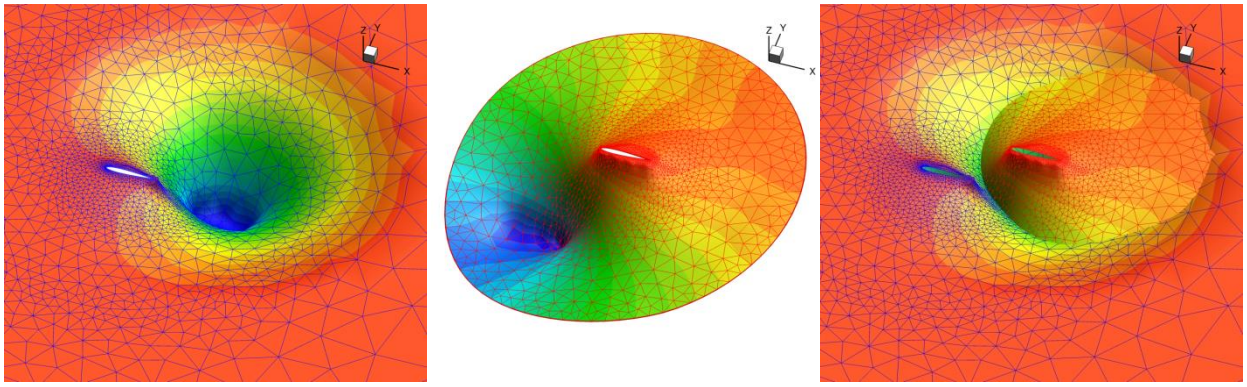
(a) Boundary conditions for grid 1 (b) Boundary conditions for grid 2

Figure III.17 Boundary conditions for EHC



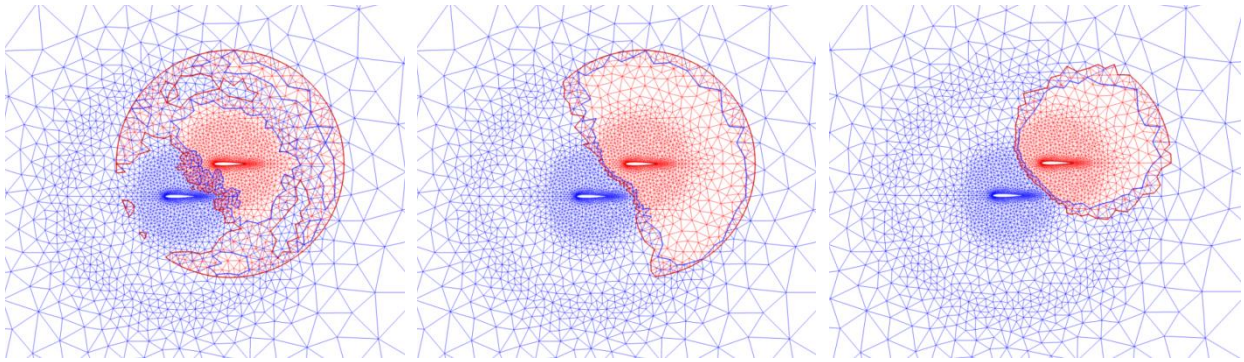
(a) Source terms for grid 1 (b) Source terms for grid 2

Figure III.18 Source terms for EHC



(a) Grid 1 (b) Grid 2. (c) Grid 1 and 2

Figure III.19 Solutions to Poisson equation in EHC



(a) Original IHC (b) Modified IHC (c) EHC

Figure III.20 Comparison of different hole cutting methods in the 2-airfoil overset grid case

The previous example selected for discussion and demonstration only involved two grids and is relatively benign. A more challenging demonstration is shown in Figure III.21, and consists of 16 identical airfoil grids overlapping on a background grid. The overset grids using the implicit hole cutting, modified implicit hole cutting and the elliptic hole cutting methods are shown in Figure III.22. As seen in Figure III.22(a), the original implicit hole cutting selects cells based only on cell quality and, therefore, in regions away from the airfoils becomes randomly distributed between the various grids. The modified implicit hole cutting method that favors cells from the grid within a local scope constructs an improved overset grid as shown in Figure III.22(b). Figure III.22(c, d) illustrate the overset grids using the elliptic hole cutting method based on source terms favoring cell quality and airfoil grids, respectively. As can be observed, the constructed overset grids using this method produces continuous grids with smooth transitions.

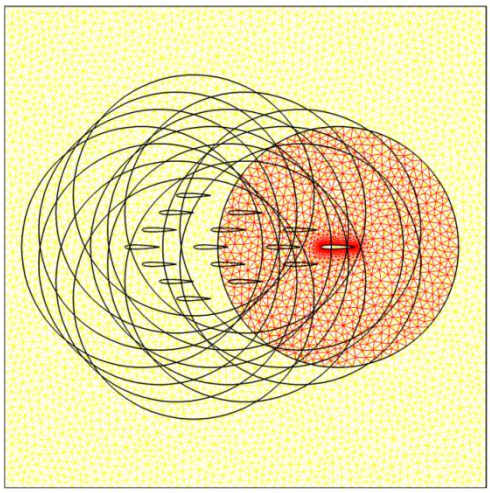
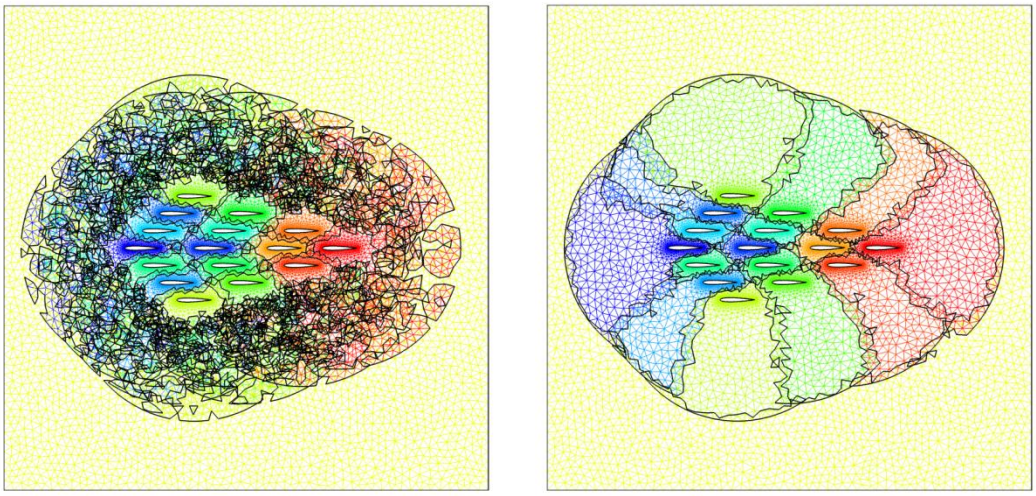
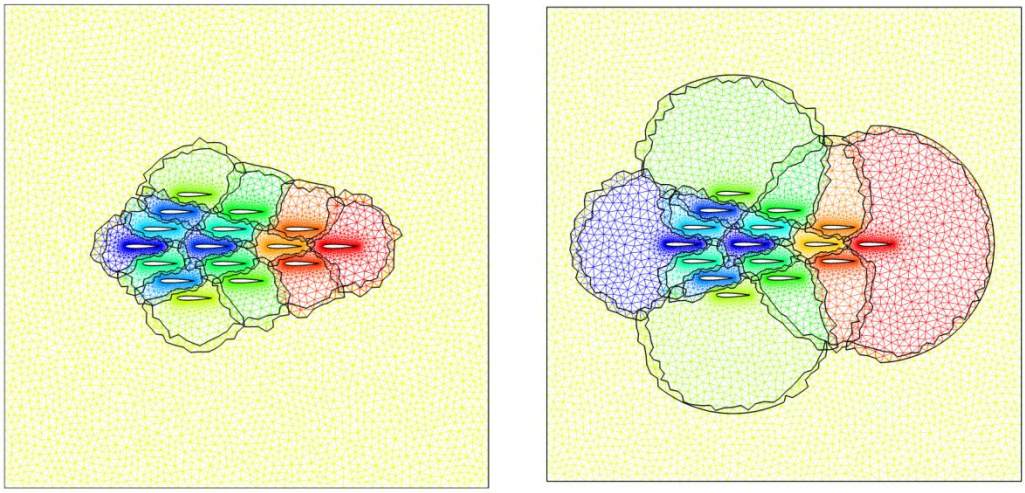


Figure III.21 16 airfoil grids overlapping on a background grid



(a) Original IHC (b) Modified IHC



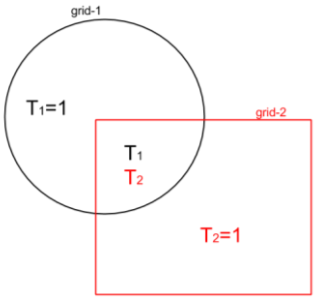
(c) EHC, in favor of cell quality (d) EHC, in favor of airfoil grids

Figure III.22 Comparison of different hole cutting methods for the 16 airfoil overset grid case

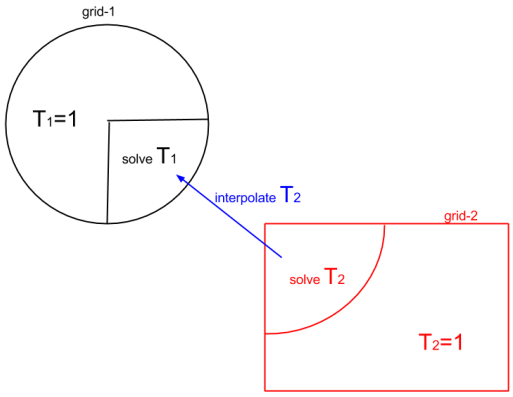
A potential drawback of the EHC procedure, like in IHC, is the large amount of point search operations required. This problem is illustrated in Figure III.23. The pseudo-temperatures T_1 and T_2 are first solved on grid-1 and grid-2, respectively. Subsequently, T_1 and T_2 must be compared for every grid node within the overlapped region. As depicted in Figure III.23(b), for each node of grid-1 in the overlapped region, the corresponding T_2 at the node must be interpolated

based on its location in grid-2. A similar process is required for each node of grid-2 in the overlapped region. In a parallel implementation on distributed machines, if the two grids are mapped to different MPI ranks, extra communication between the MPI ranks would be needed in order to perform the point search. Considering the large amount of grid nodes in the overlapped region, the added communication cost would not be trivial. In order to minimize the amount of point search operations in EHC an alternative procedure has been devised. In this modified process, once T_1 and T_2 have been solved within their respective grids, the corresponding pseudo-temperatures for the other grid in the overlapped region are additionally solved via the solution of a Poisson equation. For example, as shown in Figure III.23(c), this can be accomplished by interpolating T_2 only at the boundary of the overlapped region from grid-2 to grid-1. Thereafter, an additional Poisson problem may be used to determine T_2 on grid-1. The number of grid nodes that need to be searched is significantly reduced (number of boundary nodes versus interior nodes of the overlapped region). As should be evident, although this modification greatly reduces the amount of point search operations, it incurs the added cost of solving extra Poisson equations. In the current implementation, the cost associated with solving the extra Poisson equations in the modified EHC is commensurate with that of the added point search operations in the original EHC. However, the modified EHC is much more scalable by trading communication cost for computation cost and, thus, more suitable for large scale problems. However, the modified EHC cannot use the source term that favors cell quality as given in Eq. (III.6) for the Poisson equation, because that requires point searches for the nodes in the interior of the overlapped region, followed by interpolating the source term at these nodes, which is intentionally avoided in modified EHC to achieve better parallel performance.

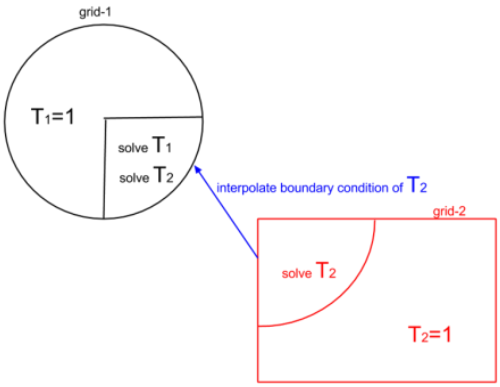
A qualitative demonstration of the modified EHC is depicted in Figure III.24. Figure III.24(a) show the 9 separate grids generated about each letter of the word “SimCenter” and the background grid. The modified EHC is utilized to construct the overset grid assembly as shown in Figure III.24(b). As seen, despite having numerous grids overlapped within the same regions, the final overset grid assembly possesses smooth transitions between the various grids.



(a) Overset grids

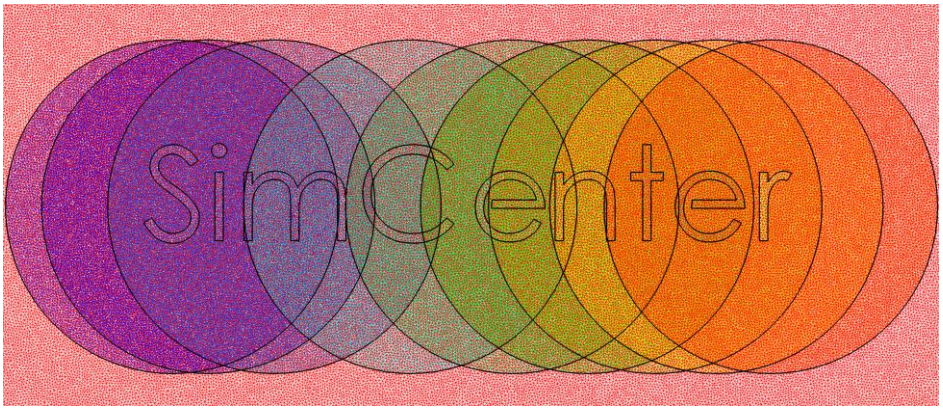


(b) Original EHC

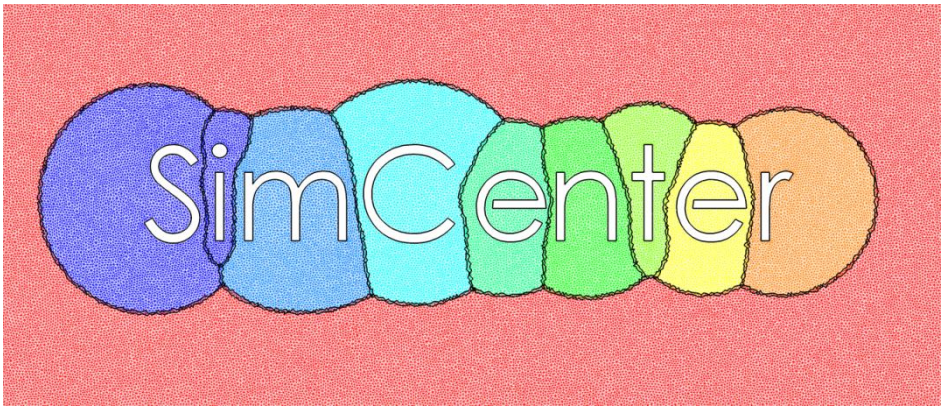


(c) Modified EHC

Figure III.23 Illustration of original and modified EHC



(a) Grids prior to hole-cutting



(b) Final assembled grids

Figure III.24 Hole-cutting of "SimCenter" using modified EHC

By using cell selection approaches previously described, the overlap minimization can become an automated procedure, without the need for user input. However, there are some distinct advantages for the EHC approach. Like IHC, EHC can yield a mesh in favor of a user-chosen mesh quality metrics. This may be accomplished by devising corresponding source terms for the Poisson equation. Additionally, EHC yields a continuous cell selection in the vicinity of geometries, like the cell selection approach based on the distance function. Moreover, if parallelization is preferred, the modified EHC, which limits communication by eliminating point searches in the interior of the overlapped regions, is a viable approach. Finally, although the method requires solutions to

Poisson equations, which may be computationally expensive particularly on anisotropic grids, these solutions do not need to be fully converged to construct the overset grids. Typically, only a few orders of magnitude of reduction in the residual is required.

Note that, in this research the modified IHC and EHC are only implemented for 2D simulations, while the 3D overset grid assembly only utilizes the distance function approach. However, the 3D overset grid assembly is a general framework, and other implementations of cell selection approaches are possible.

III.5 Dynamic Solution Strategy

In the case of multiple bodies moving relative to one another, the regions of overset change at each instant of time. Hence, the hole cutting is performed dynamically and the nodes that are active may be different as the simulation proceeds. Therefore, if a node that has been previously absent from the solution (i.e. blanked-out) is introduced into the computational domain, no data is available to construct the time history of the solution required for temporal discretization. In order to provide sufficient information for temporal integration, the solutions at previous time instances (e.g. previous two time instances for BDF2 scheme) for this node must be re-initialized by interpolating from the donor cells at previous time instances.

On the other hand, a node that has been an invalid node in previous time instances cannot be re-initialized and thus, during the overset grid assembly for current time instance, the node should be blanked-out.

III.6 Adaptive Overset

The benefits of Adaptive Mesh Refinement (AMR) have been well established and these methods have been utilized for both steady and unsteady simulations. Moreover, AMR may have an even greater impact when used within an overset grid methodology. The primary motivations behind the original development of overset grid approaches were to ease the burden of grid generation and to allow for applications where large relative motion between bodies was present. In terms of grid generation, these methods allow for discretization of each component to be independently generated, as to best resolve the geometric complexities and flow physics, and then overlapped onto the grid resolving the far field. However, for moving boundary simulations with large relative motion, as the bodies separate, the disparity between the cell volumes in the overlapped grids typically become pronounced. Hence, important flow features may become distorted or lost when crossing the overset boundaries. Combining adaptive mesh refinement with an overset grid methodology mitigates these difficulties.

In the current work, the adaptive grid methodology developed by Behzad [39] has been incorporated into the overset grid approach for 2D applications. This adaptive grid methodology utilizes constrained approximation and is capable of performing multi-level h-, p-, and hp-refinement and de-refinement. This methodology allows for hanging-nodes and may be utilized with any continuous Galerkin formulation. Furthermore, adjoint-based adaptation is available for steady flows, while feature-based adaptation can be used for both steady and unsteady cases. Further details concerning the adaptive grid methodology may be found in the cited reference. For the preliminary results shown herein, the combined capabilities can perform dynamic hole cutting using the modified EHC on adaptively h-refined grids. The ability to perform de-refinement on overset grids has not been added.

The results presented for the adaptive overset capabilities have been selected for demonstration purposes only. Within the current work only adaptive refinement has been implemented.

III.6.1 Steady Multiple Airfoils

The first case is concerned with the steady, inviscid flow over three airfoils in close proximity. The freestream has a Mach number of 0.2 and 0° angle-of-attack. The boundaries of the three airfoil grids in relation to the far field grid is shown in Figure III.25. Second-order (P_1) elements are used for the simulation. The initial assembled overset grids and Mach number contours are shown in Figure III.25(a). As seen, the resolution in the region between the airfoils is not sufficient. As previously discussed, the relevance of such a case stems from the fact that overset grid technology was originally developed to allow separately generated grids to be overset prior to simulation. Although each grid may be adequate to resolve the flow about the isolated body, once overlapped the discretizations between bodies may no longer be capable of resolving the interaction. Additionally, in order to obtain a valid grid after hole cutting, the cells sizes in the various grids must be such that at least one layer of overlap exists. If the airfoils are moving relative to one another, like in the case of retracting high-lift devices, the initial resolution would constrain the extent of possible motion. The multi-level h-refined overset grids, and Mach number contours, are shown in Figure III.25(b). Unlike in the original overset grids, the interaction between the airfoils and the overset boundaries are well resolved. Utilizing the adaptive overset capabilities, the airfoils may be essentially moved to the point of contact.

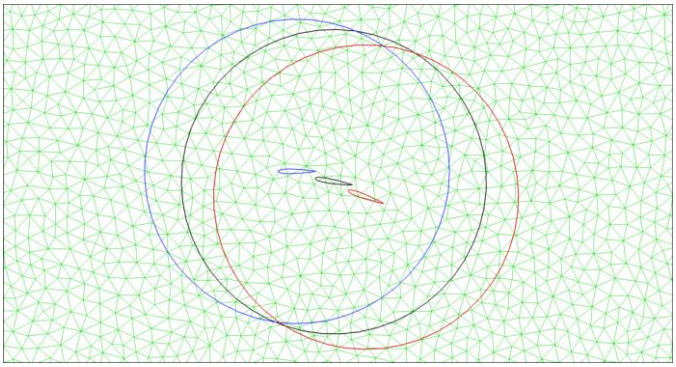
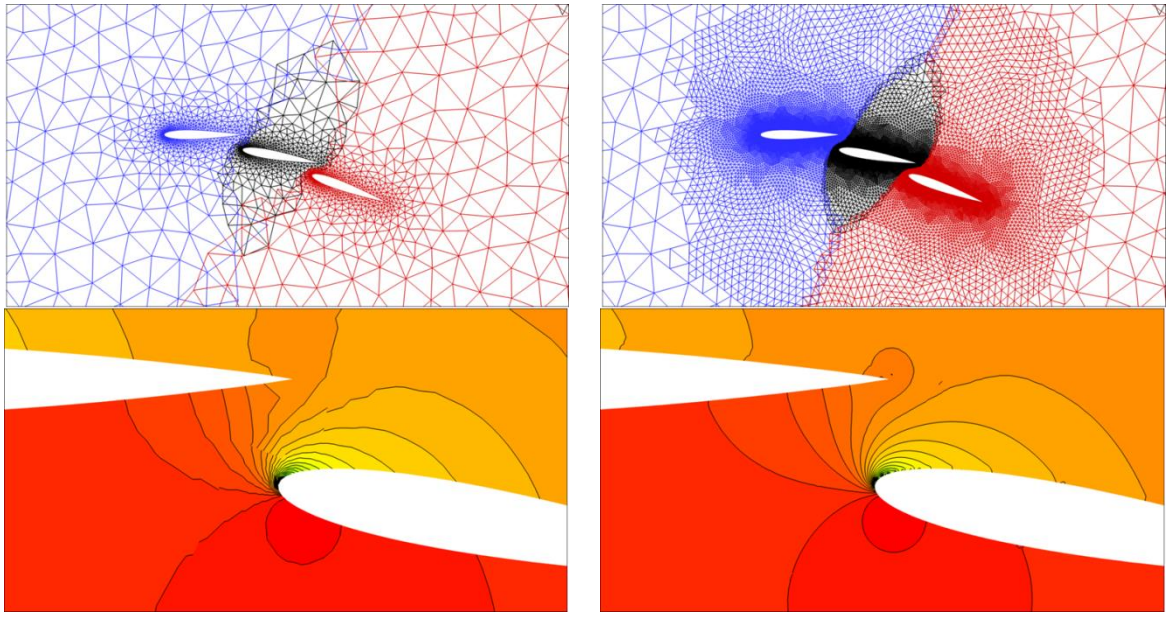


Figure III.25 Multiple airfoil overset grids before hole cutting



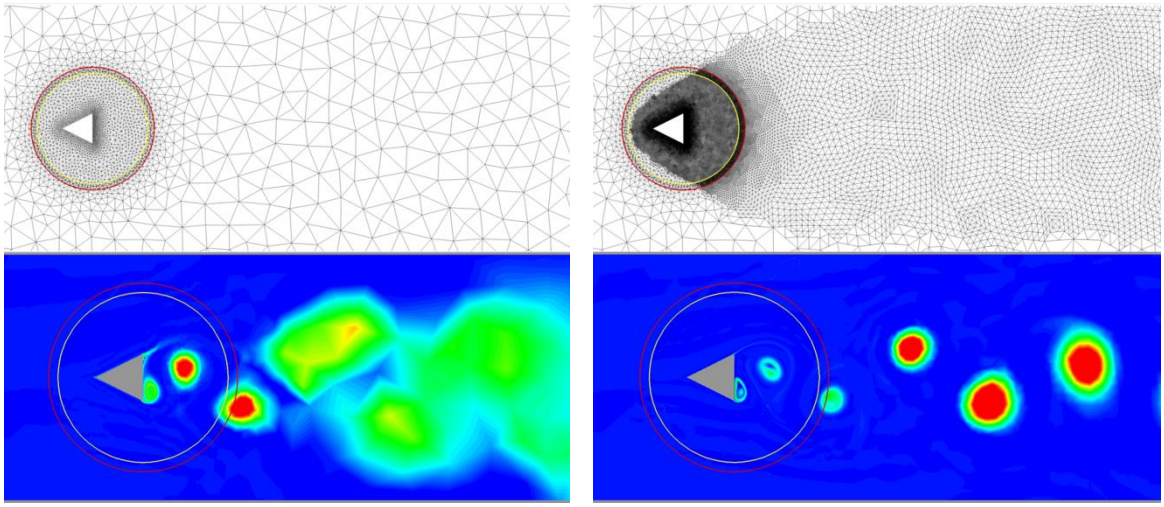
(a) Initial overset grids and Mach contours (b) Adapted overset grids and Mach contours

Figure III.26 Mesh and Mach contours in the multiple airfoil overset grid case

III.6.2 Unsteady Inviscid Triangular Wedge

To illustrate the use of adaptive overset grids to resolve features that propagate across overset boundaries, the unsteady inviscid flow over a triangular wedge is simulated. Once again, second-order (P_1) elements are used and the freestream Mach number is 0.2 with 0° angle-of-

attack. Figure III.27 depicts entropy contours on the original and multi-level h-adapted overset grids, respectively. It should be Note that the flow is assumed to be inviscid and, hence, the entropy is used to highlight the vortical flow. For the non-adaptive overset grids the vortices dissipate quickly, whereas in the adaptive overset grids they are adequately resolved far downstream.



(a) Initial overset grids and entropy contours (b) Adapted overset grids and entropy contours

Figure III.27 Unsteady vortex shedding over a triangular wedge

CHAPTER IV

CODE VERIFICATION

Previous attempts to study the order of accuracy of overset grid schemes have been performed by Galbraith [40] using a discontinuous Galerkin finite element formulation for steady channel flow, and by Meakin [41] in finite volume implementations. In the current work, the order of accuracy of the Petrov-Galerkin overset scheme is assessed using the method of manufactured solutions based on a comprehensive set of guidelines [42]. The method of manufactured solutions is a general procedure for generating nontrivial exact solutions to a PDE or system of PDEs.

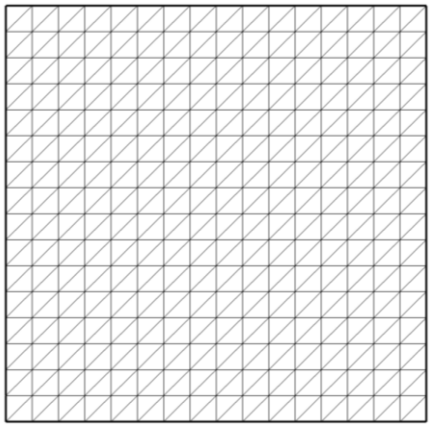
IV.1 2D Code Verification

For the 2D code verification, the following trigonometric functions are used to derive the forcing functions and boundary conditions

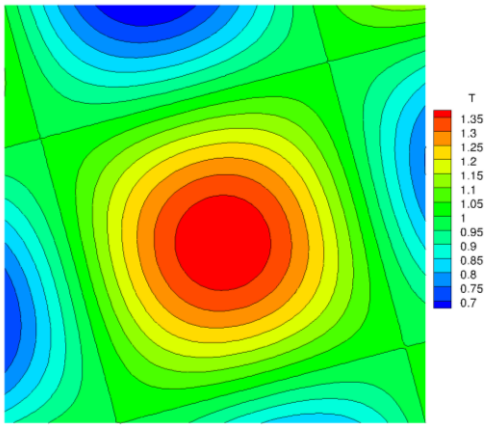
$$\begin{aligned} \rho &= \rho_0\{1 + 0.2\cos[\pi(c_1x - s_1y)] + 0.2\cos[\pi(s_1x + c_1y)]\} \\ u &= u_0\{1 + 0.2\cos[\pi(c_2x - s_2y + 0.1)] + 0.2\cos[\pi(s_2x + c_2y + 0.1)]\} \\ v &= v_0\{1 + 0.2\cos[\pi(c_3x - s_3y - 0.1)] + 0.2\cos[\pi(s_3x + c_3y + 0.1)]\} \\ T &= T_0\{1 + 0.2\cos[\pi(c_4x - s_4y - 0.1)] + 0.2\cos[\pi(s_4x + c_4y - 0.1)]\} \end{aligned} \tag{IV.1}$$

where ρ_0, u_0, v_0, and T_0 corresponded to the freestream conditions for a flow with angle of attack, Mach number, and Reynolds number of 15°, 0.2, and 10^2, respectively. The terms c_i and s_i correspond to cosine and sine function of 0°, 40°, 80°, and 120°, respectively.

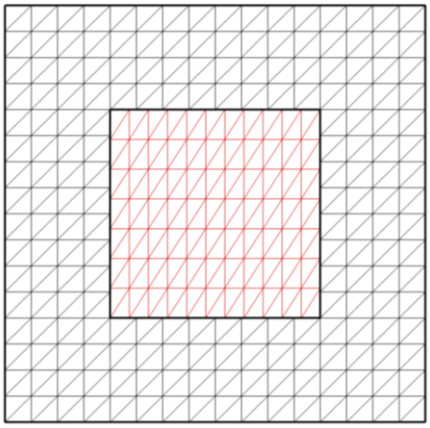
The order of accuracy is evaluated over a sequence of refined meshes for linear (P_1), quadratic (P_2), and cubic (P_3) polynomial bases. The coarsest meshes used in this refinement are shown in Figure IV.1(a-c) for the single grid, the zero-layer non-matched (i.e., zonally patched) grids, and the overlapping grids, respectively. The contours of the manufactured solutions using P_3 elements for the temperature field in laminar cases are shown for the corresponding grids in Figure IV.1(d-f). As seen, even for the coarsest meshes, smooth transitions occur between the overlapped boundaries for both zero-layer non-matched and overset grids. Only considering density and temperature for brevity, Figure IV.2 illustrates that the slope of the error during refinement on each of the grids is maintained for both inviscid and laminar flows, and matches the order of accuracy for the utilized elements. That is, reference slopes of approximately 2, 3, and 4 for linear, quadratic, and cubic elements, respectively.



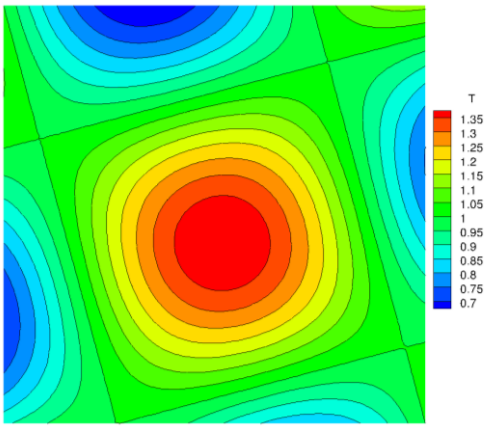
(a) Single grid



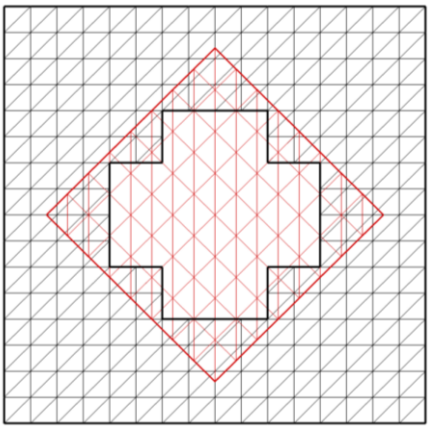
(d) Single grid, laminar, P_3 elements



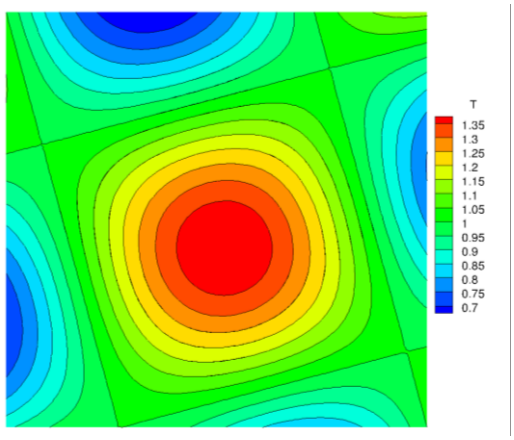
(b) Zero-layer non-matched grids



(e) Zero-layer, laminar, P_3 elements

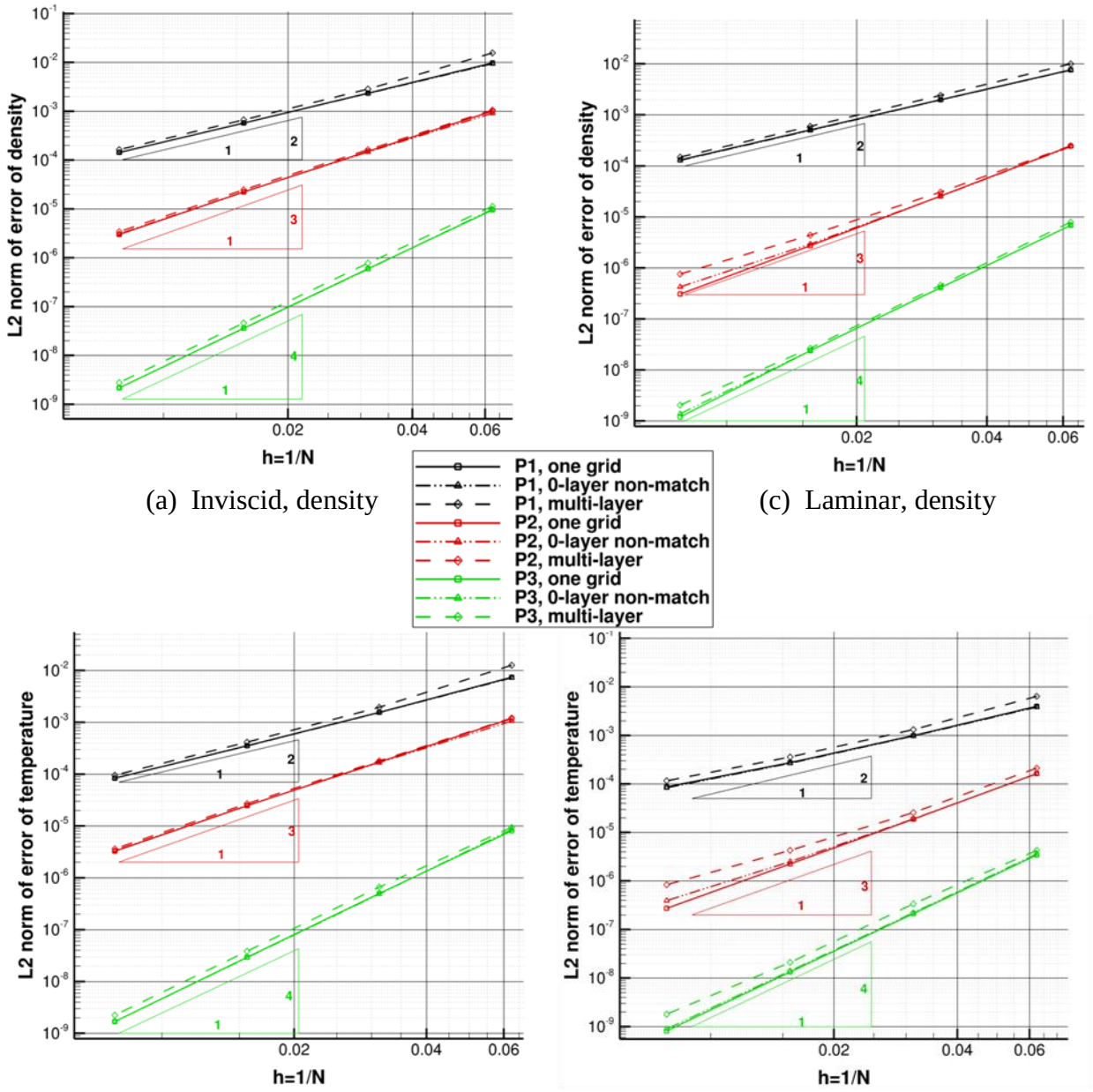


(c) Overlapping grids



(f) Overlapping, laminar, P_3 elements

Figure IV.1 Coarsest grids for assessing order of accuracy using MMS and temperature contours



(a) Inviscid, density (c) Laminar, density

(b) Inviscid, temperature (d) Laminar, temperature

Figure IV.2 Order of accuracy for inviscid and laminar flow using P_1, P_2, and P_3 elements

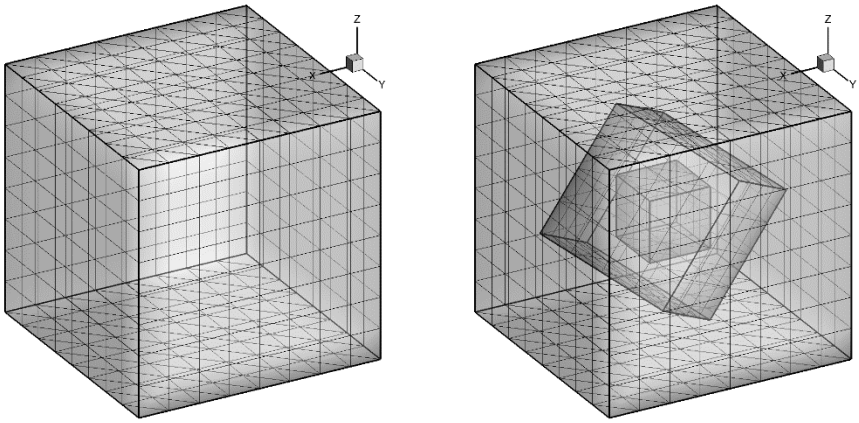
IV.2 3D Code Verification

For the 3D code verification, the following trigonometric functions are used to derive the forcing functions and boundary conditions

$$\rho = 1.0 * [1.0 + \sin(\pi x)\cos(\pi x)\sin(\pi y)\cos(\pi y)\sin(\pi z)\cos(\pi z)]$$
$$\rho u = 0.5 * [1.0 + \sin(1.5\pi x)\cos(1.5\pi x)\sin(1.5\pi y)\cos(1.5\pi y)\sin(1.5\pi z)\cos(1.5\pi z)]$$
$$\rho v = 0.5 * [1.0 + \sin(1.5\pi x)\cos(1.5\pi x)\sin(1.5\pi y)\cos(1.5\pi y)\sin(1.5\pi z)\cos(1.5\pi z)]$$
$$\rho w = 0.1 * [1.0 + \sin(1.5\pi x)\cos(1.5\pi x)\sin(1.5\pi y)\cos(1.5\pi y)\sin(1.5\pi z)\cos(1.5\pi z)]$$
$$\rho E = 3.0 * [1.0 + \sin^2(\pi x)\sin^2(\pi y)\sin^2(\pi z)]$$

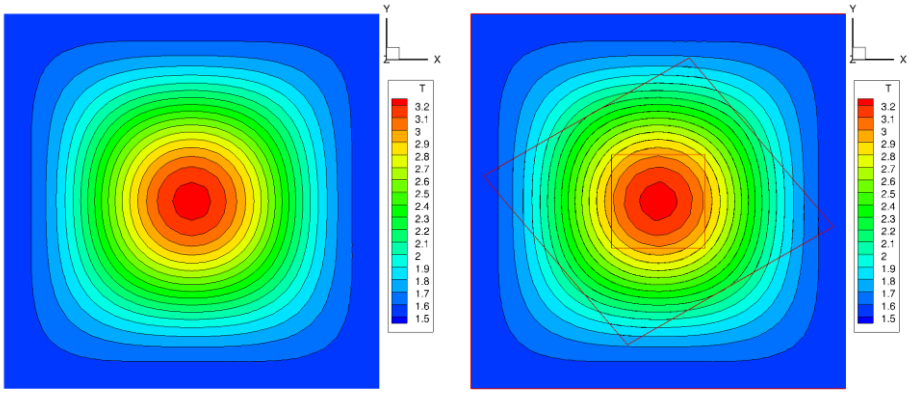
(IV.2)

The order of accuracy is evaluated over a sequence of refined meshes for a unit cube with prismatic elements for linear (P_1) and quadratic (P_2) polynomials. Both single grids and overset grids are considered. Note that the order of accuracy for linear and quadratic tetrahedral elements in non-overset approach have been previously verified with the current software [37]. The sequence of meshes for the single grid results range from 512 to 262144 nodes, with the coarsest mesh shown in Figure IV.3(a). The overset approach, with the coarsest mesh shown in Figure IV.3(b), has a corresponding sequence of meshes ranging from 853 to 307187 nodes. Contours of temperature and z-component of velocity on the second finest grids are shown in Figure IV.4 for the center plane in the z-direction. In Figure IV.4(b, d), the overset boundaries are indicated. As seen, smooth transitions occur across the overset boundaries. Figure IV.5 illustrates that the slope of the error during refinement for both the single grid and overset grid computations are maintained, and matches the design order of accuracy for the linear and quadratic elements. That is, a slope of approximately 2 for the P_1 computations, and a slope of approximately 3 for the P_2 computations.

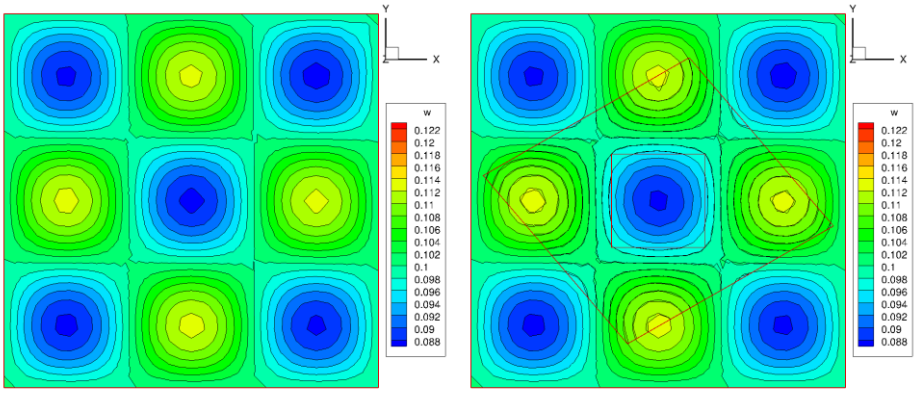


(a) Single grid (b) Overset grids

Figure IV.3 Coarsest grids used for assessing order of accuracy using MMS

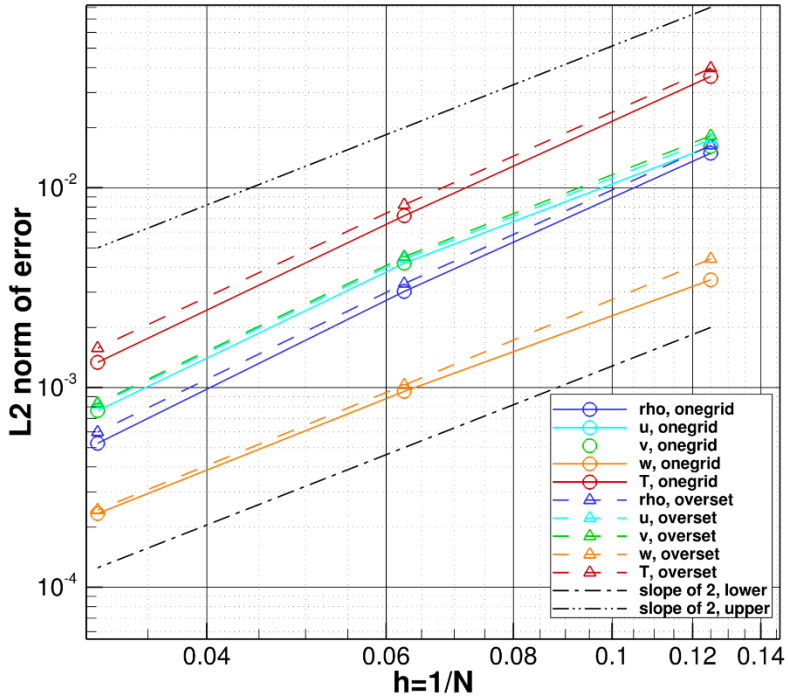


(a) Temperature on single grid (b) Temperature on overset grids

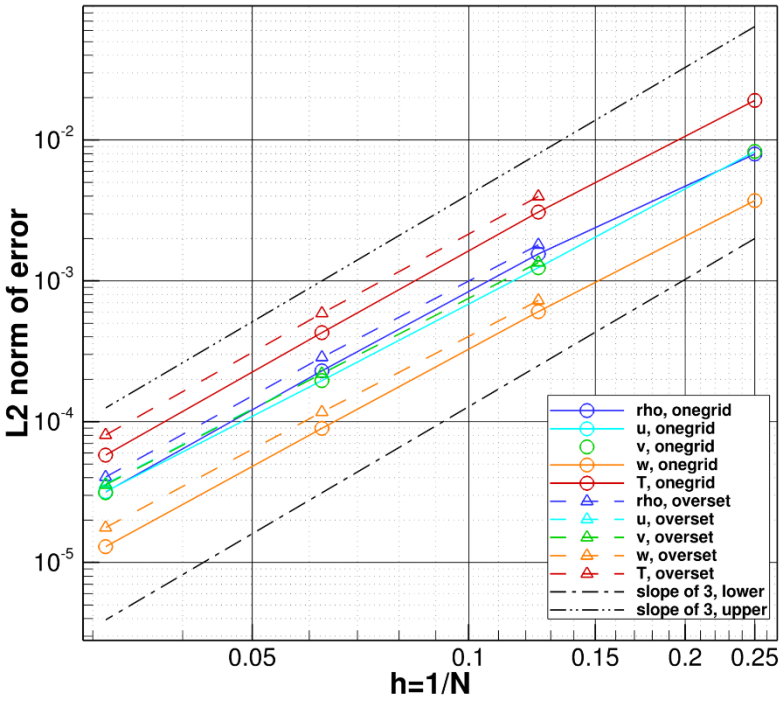


(c) z-Velocity on single grid (d) z-Velocity on overset grids

Figure IV.4 Computed manufactured solutions for 3-D Euler equations using P_1 elements on central z plane



(a) P_1 elements



(b) P_2 elements

Figure IV.5 Observed order of accuracy

CHAPTER V

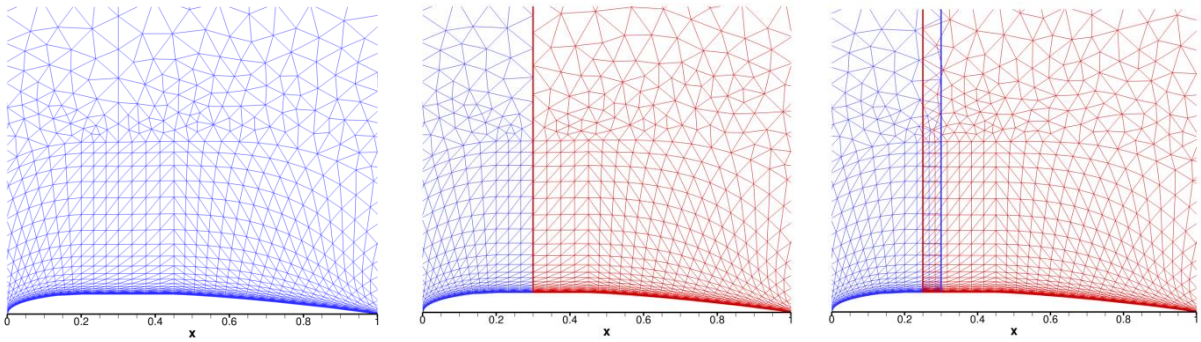
RESULTS

V.1 2D Results

V.1.1 Steady-state Turbulent NACA0012 Airfoil Simulations

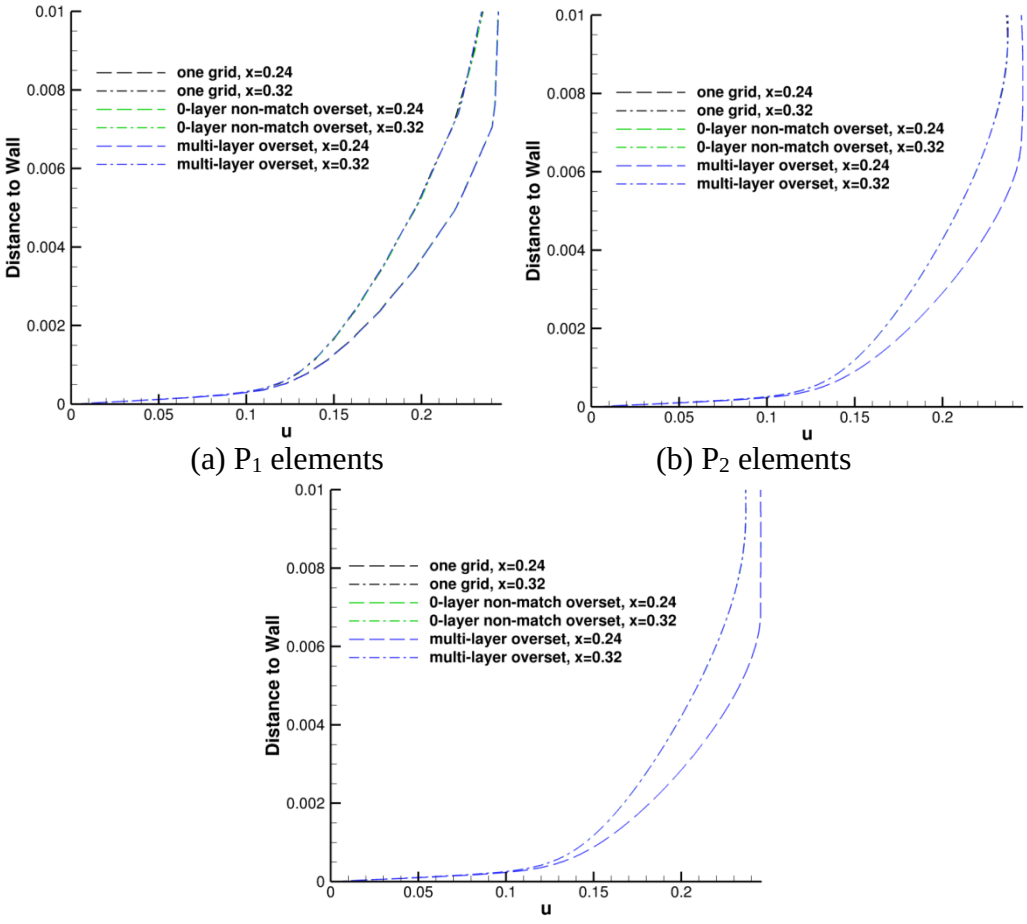
To assess the performance of the overset method for a more realistic flow, the steady turbulent flow over a NACA 0012 airfoil was examined. The modified one-equation Spalart-Allmaras turbulence model is used [19]. The freestream angle-of-attack, Mach number, and Reynolds numbers are 3°, 0.2, and 10^6, respectively. The single, zero-layer non-matched, and multi-layer overset grids for the upper surface of the airfoil are shown in Figure V.1. The overlapping regions in the two overset cases extend from the airfoil surface to the far field. Note that, for the zero-layer non-matched overset grids, the upstream and downstream grids do not match each other on the overset interface, neither inside the boundary layer nor in the far field.

Illustrated in Figure V.2 is the x-component of velocity versus wall distance at two chord-wise locations. The chord-wise locations are slightly upstream and downstream of the overlapped region ($x = 0.24$ and $x = 0.32$). Figure V.2(a, b, c) show the profiles obtained from single and overset grid simulations with P_1, P_2 and P_3 elements. As seen, the profiles are nearly indistinguishable for the single grid and the two overset grid simulations for all orders. With close inspection only minute differences can be observed for the P_1 profiles, but all essentially coalesce for P_2 and P_3 results.



(a) Single grid (b) Zero-layer non-matched grids (c) Multi-layer overset grids

Figure V.1 Meshes used in the steady turbulent NACA0012 airfoil simulations



(a) P_1 elements (b) P_2 elements

(c) P_3 elements

Figure V.2 x-velocity profiles using P_1, P_2, and P_3 elements for turbulent flow over NACA0012 airfoil

V.1.2 Unsteady Sinusoidally Oscillating Airfoil

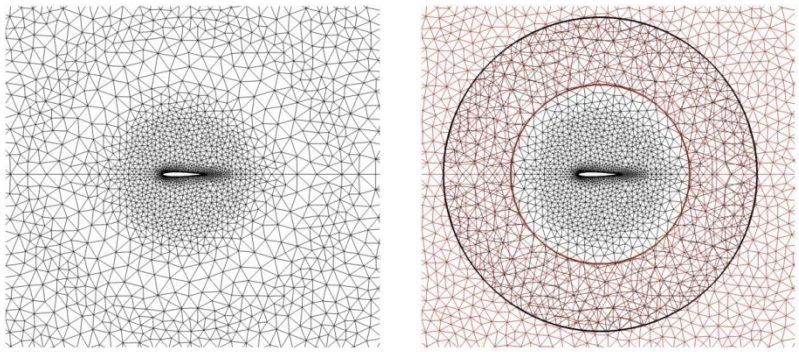
Two cases have been selected to demonstrate the overset method for moving boundary problems in which a hole in the overset mesh is generated prior to simulation. In both cases, the flow is assumed to be inviscid, the motion of the airfoil is prescribed, and the grids move as rigid bodies. Furthermore, analytic grid velocities are utilized based on the prescribed motion.

V.1.2.1 Sinusoidally Pitching Airfoil

The forced pitching oscillation of a NACA 0012 airfoil [43] has been used as a benchmark case for many unsteady dynamic mesh code validation studies. The pitching of the airfoil about its quarter chord is analytically prescribed as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t) \tag{V.1}$$

where the mean angle-of-attack α_m and amplitude-of-oscillation α_0 are 2.89° and 2.41°, respectively. The non-dimensional circular frequency is defined in terms of the reduced frequency k and freestream Mach number M_∞ as $\omega = 2kM_\infty$. In the present results, k and M_∞ are assumed to be 0.0808 and 0.6, respectively. The single grid and multi-layer overset grids used for comparisons are shown in Figure V.3. Although multiple layers of overlap can be seen in Figure V.3(b), only minimal overlapping is required for this pitching motion. Shown in Figure V.4 is the time histories of the coefficient of lift for both the single grid and overset grid simulations using P_1 elements. Good agreement is observed between the different discretizations, indicating acceptable accuracy for the overset method on moving meshes.



(a) Single grid (b) Overset grids

Figure V.3 Single and overset grids used for the sinusoidally pitching airfoil simulations

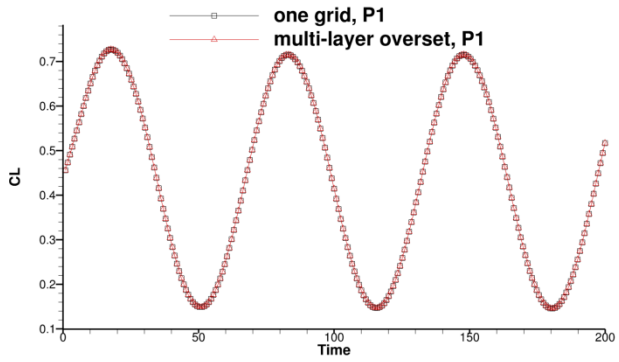


Figure V.4 Time histories of the C_L for the sinusoidally pitching airfoil simulations with P_1 elements

V.1.2.2 Sinusoidally Pitching and Plunging Airfoil

The NACA 0012 airfoil meshes shown in Figure V.3 are additionally used to examine a forced pitching and plunging response. In this case, multiple layers of overlap are required to account for the airfoil movement due to plunging. Plunging and pitching of the airfoil about its quarter chord is analytically prescribed as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t)$$
$$h(t) = h_0 \sin(0.5\omega t) \tag{V.2}$$

where the mean angle-of-attack α_m and amplitude-of-oscillation α_0 are 0° and 5°, respectively. The reduced frequency is the same as previously defined, and the freestream Mach number is 0.4 in this case. The plunge amplitude h_0 is $0.4 \cdot c$, where c is the chord length of the airfoil. To illustrate the extent of the motion, Figure V.5 depicts the overset grid boundaries at the maximum and minimum pitching and plunging positions. Comparison of the time histories of the coefficient of lift, for the single grid and the multi-layer overset grid simulations, are shown in Figure V.6 using P_3 elements. Once again, excellent agreement is observed in the time histories.

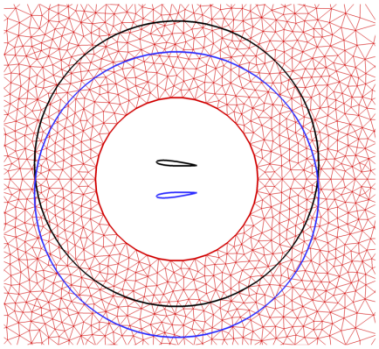


Figure V.5 Extent of motion of airfoil grid in the sinusoidally pitching and plunging airfoil simulation

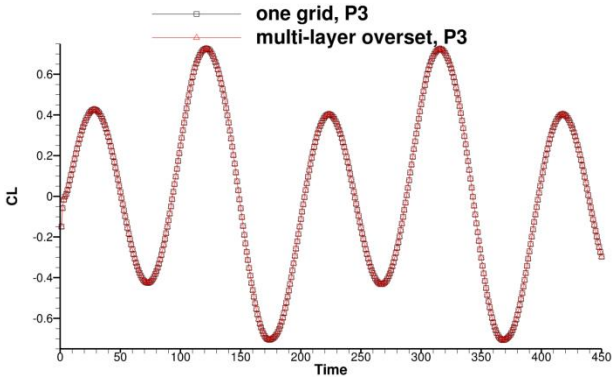
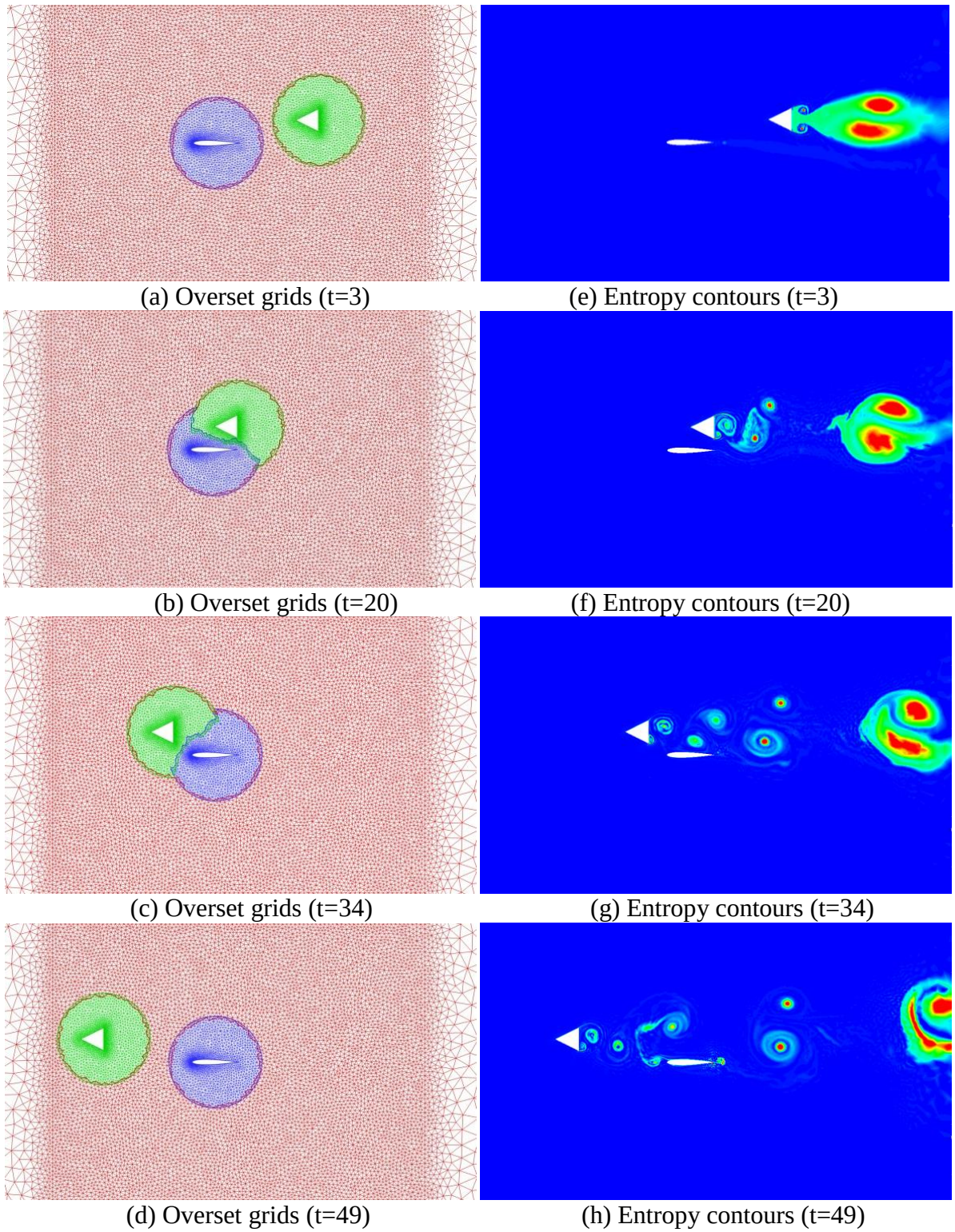


Figure V.6 Time histories of the C_L for the sinusoidally pitching and plunging airfoil simulations using P_3 elements

V.1.3 Dynamic Hole Cutting for Relative Body Motion

To demonstrate the previously discussed dynamic hole cutting procedure, the simulation of a triangular wedge passing a NACA0012 airfoil using P_2 elements has been carried out. The purpose of the simulation is to demonstrate the ability of the current method in dealing with large relative motion between multiple bodies. Note that modified IHC is used in the simulation. As discussed, when a previously blanked-out node is introduced into the computational domain, its donor cell at each previous time instance is used to re-initialize the solution at that time instance. The Mach number and angle-of-attack of the freestream are 0.1 and 0°, respectively. The airfoil is stationary, while the triangle is moving upstream at a constant Mach number of 0.1. Separate grids are used to discretize the far field, the triangular wedge, and the airfoil geometry. As seen in Figure V.7, four non-dimensional time instances are selected to display the relative body motion. Figure V.7(a-d) depicts the three overlapping grids after the dynamic hole cutting has been performed, whereas Figure V.7(e-h) show entropy contours at the corresponding time instances. It should be noted that the flow is assumed to be inviscid and, hence, the entropy is used to highlight the vortical flow. The interaction between the triangular wedge and airfoil can be clearly seen in this sequence of figures. Although this particular simulation is academic in nature, it exemplifies the type of simulations that requires higher order spatial accuracy with overset grid capability. Higher spatial accuracy is required to preserve and propagate the vortices over relatively large distance to capture the interaction.



(a) Overset grids (t=3)
(e) Entropy contours (t=3)
(b) Overset grids (t=20)
(f) Entropy contours (t=20)
(c) Overset grids (t=34)
(g) Entropy contours (t=34)
(d) Overset grids (t=49)
(h) Entropy contours (t=49)

Figure V.7 Overset grids after hole cutting and entropy contours from P_2 simulation at various time instances

V.2 3D Results

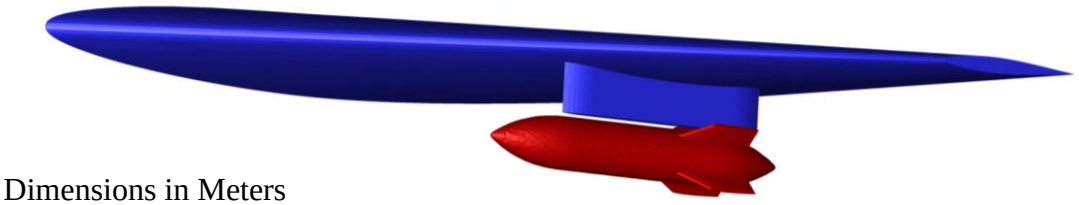
V.2.1 Steady-State Turbulent Wing/Pylon/Finned-Store Simulations

Steady-state turbulent simulations of a wing/pylon/finned-store (WPFS) configuration on single grid and overset grids are carried out using third order (P_2) tetrahedral elements. The WPFS configuration consists of a delta wing with 45° of leading edge sweep and a NACA-64A010 airfoil section. Connected to this wing is an ogive-flat-ogive pylon. Under the pylon is an ogive-cylinder-ogive store with four fins, which have a NACA-0008 airfoil section and a swept angle of 60°, located at 45°, 135°, 225°, and 315° with respect to the centerline of the pylon. There is a gap of 0.07 meters between the pylon and store. Dimensions and orientations of this geometry are depicted in Figure V.8. The simulations assume a freestream angle-of-attack, Mach number and Reynolds number of 0°, 0.6 and 10^6, respectively.

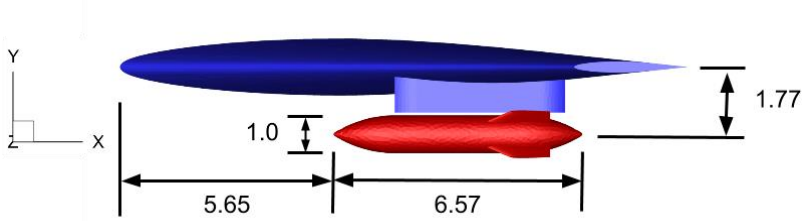
Quadratic tetrahedral elements are used in the simulations. The single grid contains 1,536,985 quadratic P_2 tetrahedral elements with 2,073,761 nodes. The overset grids contain 1,557,030 quadratic P_2 tetrahedral elements with 2,102,028 nodes. Note that all element boundaries are linear, and no mesh curving is performed. Both the single grid and the overset grid simulations are carried out using 500 CPU cores.

Both the single grid and overset grids have viscous layers which have a spacing on wall of approximately 2×10^{-5} meters normal to the surface and extend approximately 0.6 meters into the freestream. The overset grids include a body fitted grid for the wing/pylon geometry, which extends to far field, and another body fitted grid for the finned-store geometry. Note that the single and overset grids use the same surface discretization on the geometries. The single grid and the overset grids are shown in Figure V.9 to Figure V.11. Note that for visualization purpose, each

quadratic tetrahedral element is broken into four linear tetrahedral and two linear pyramid elements such that all degrees of freedom can be visualized.



Dimensions in Meters



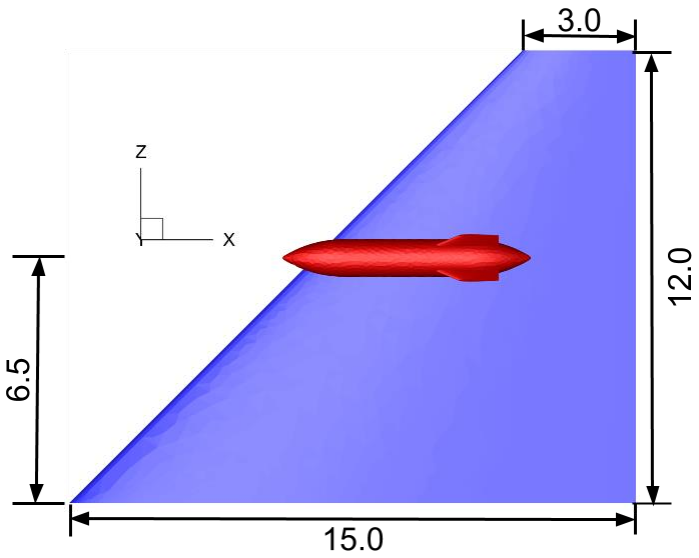
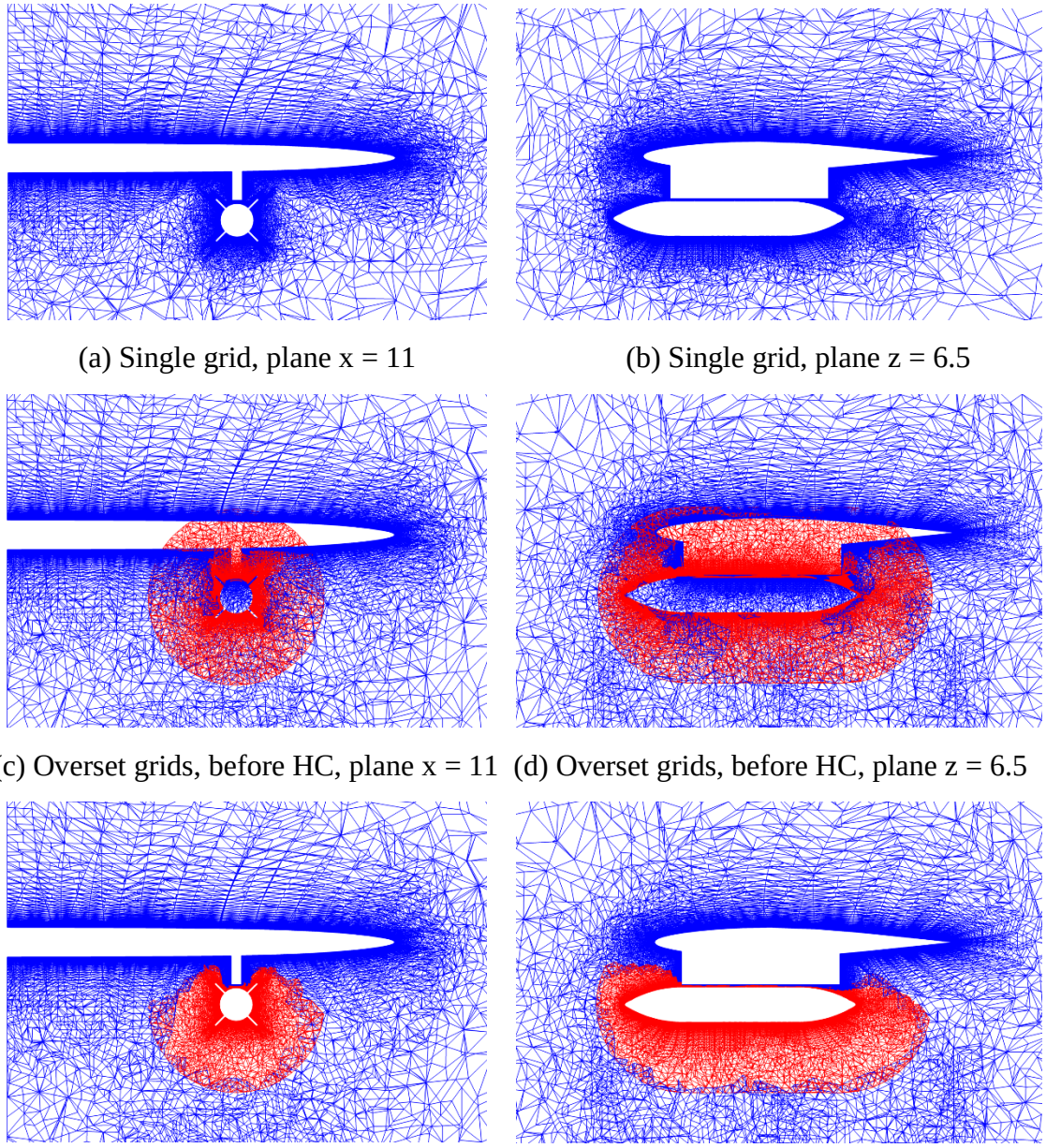


Figure V.8 The WPFS geometries



(a) Single grid, plane x = 11

(b) Single grid, plane z = 6.5

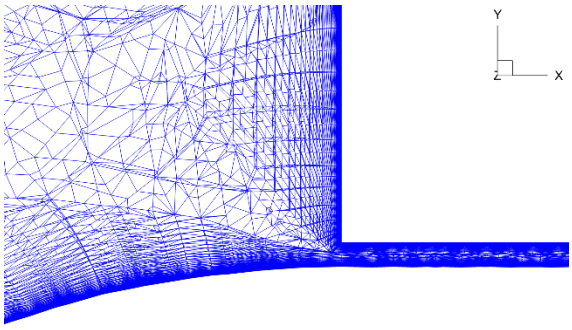
(c) Overset grids, before HC, plane x = 11

(d) Overset grids, before HC, plane z = 6.5

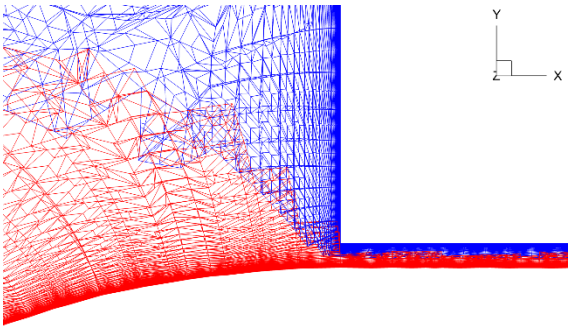
(e) Overset grids, after HC, plane x = 11

(f) Overset grids, after HC, plane z = 6.5

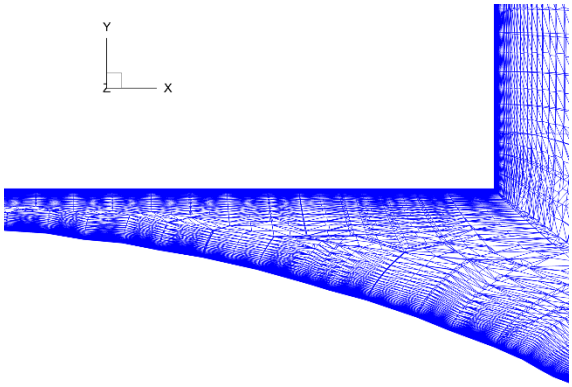
Figure V.9 Single and overset grids for steady-state turbulent WPFS simulations



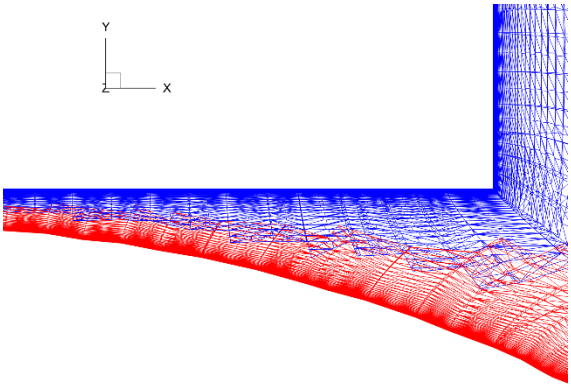
(a) Frontal part, single grid



(b) Frontal part, overset grids

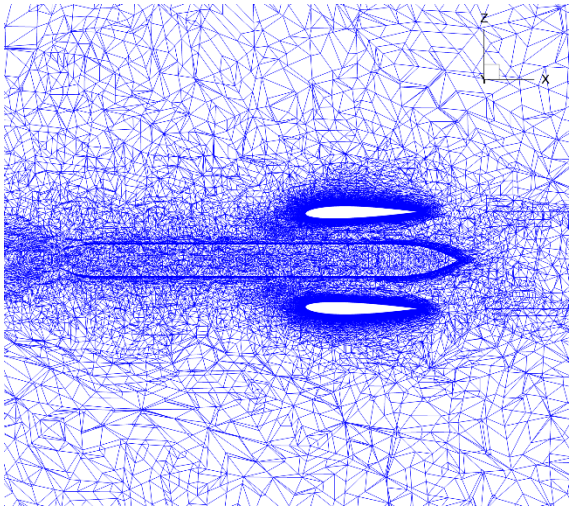


(c) Rear part, single grid

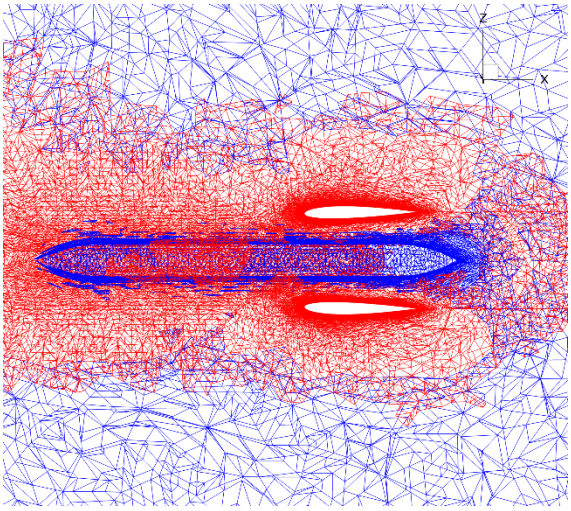


(d) Rear part, overset grids

Figure V.10 Grids in the gap region between the pylon and store (plane z = 6.5)



(a) Single grid

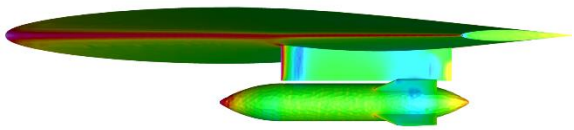


(b) Overset grids

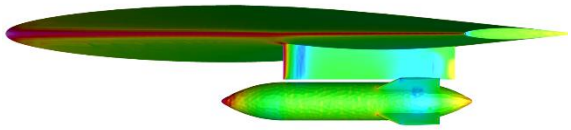
Figure V.11 Grids in the gap region between the pylon and store (plane y = -1.24)

The flow simulations utilize the modified one-equation Spalart-Allmaras turbulence model [19], an ILU(k) (with a fill-level of 2) preconditioned GMRES to solve the linear systems, and third order elements in the SUPG discretization. Steady state solutions are achieved, with the L_2 norm of the density and turbulent working variable being reduced to 10^{-15} and 10^{-14}, respectively.

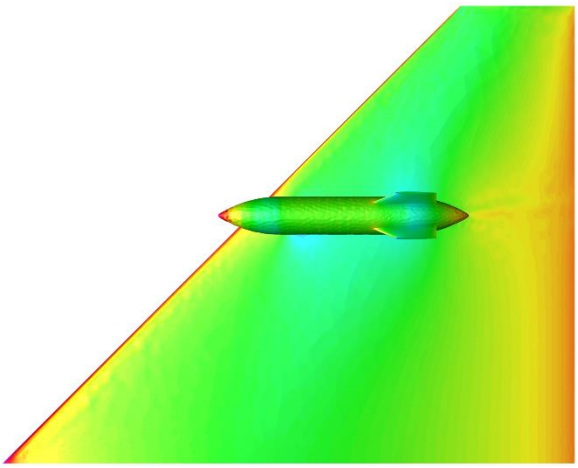
Excellent agreement between the single grid and overset grid simulations is observed in general. This agreement is evident in comparison of the C_p distributions on the WPFS geometry shown in Figure III.12, and at span-wise locations in Figure V.13. The Mach number contours are illustrated in Figure V.14 at the same span-wise locations as Figure V.13. In the gap region between the pylon and store, where the two grids overlap inside a very tight space as seen in Figure V.10 and Figure V.11, the C_p and Mach number agree well as indicated in Figure V.15. The only substantial difference is that the overset grid simulation predicts a larger separation area towards the rear part of the store. The cause of this discrepancy is speculated to be the large size of the overset boundary discretization, as shown in Figure V.10(d), along with the intense convection across the overset boundaries in this region.



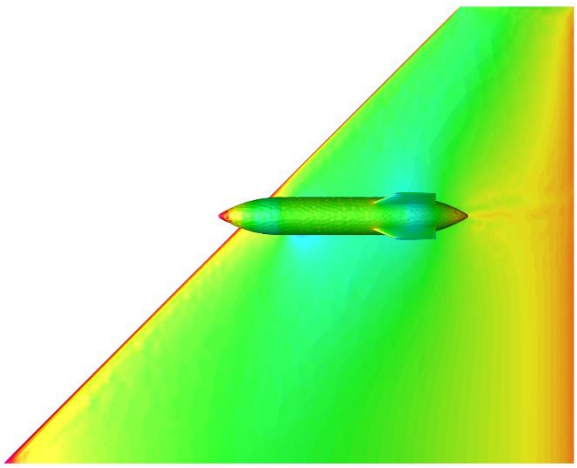
(a) Single grid, -z direction



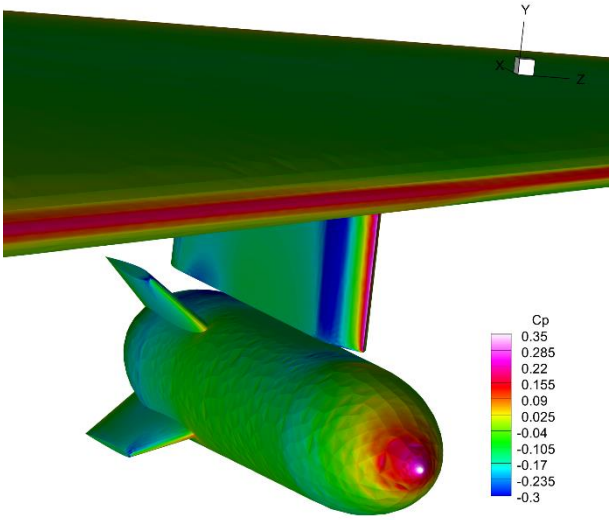
(b) Overset grid, -z direction



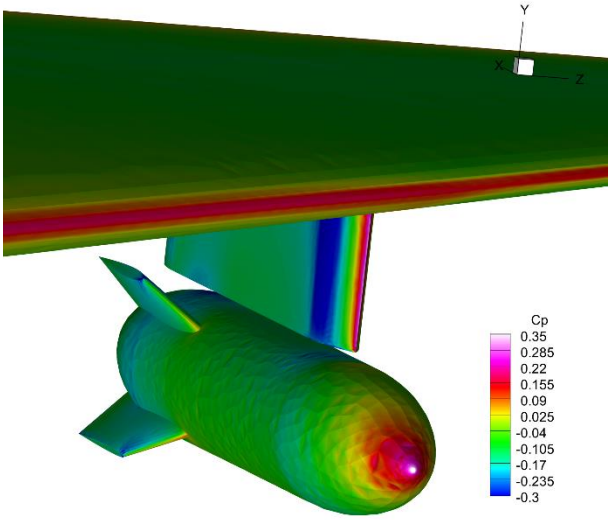
(c) Single grid, +y direction



(d) Overset grid, +y direction

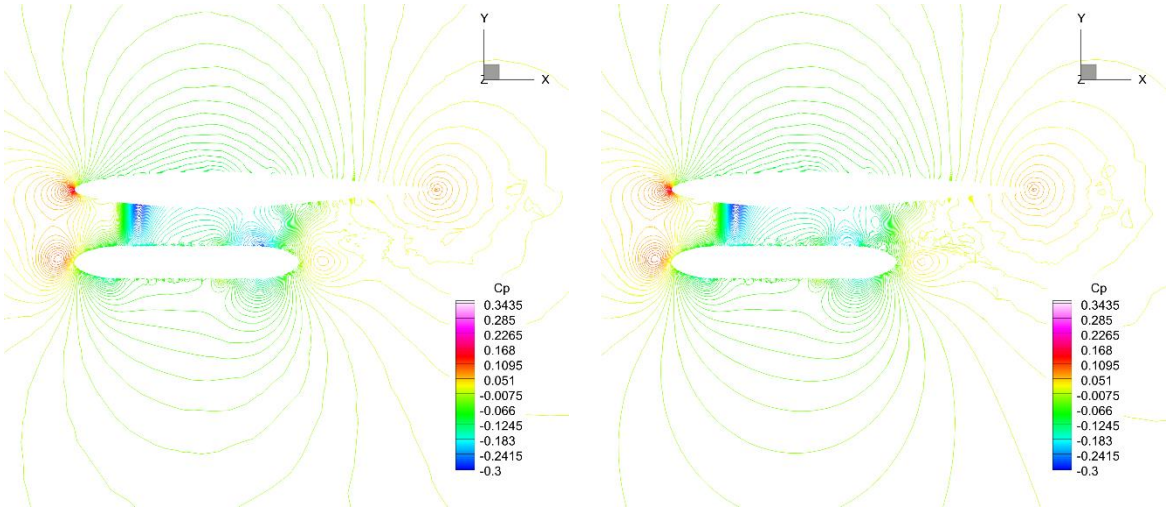


(e) Single grid

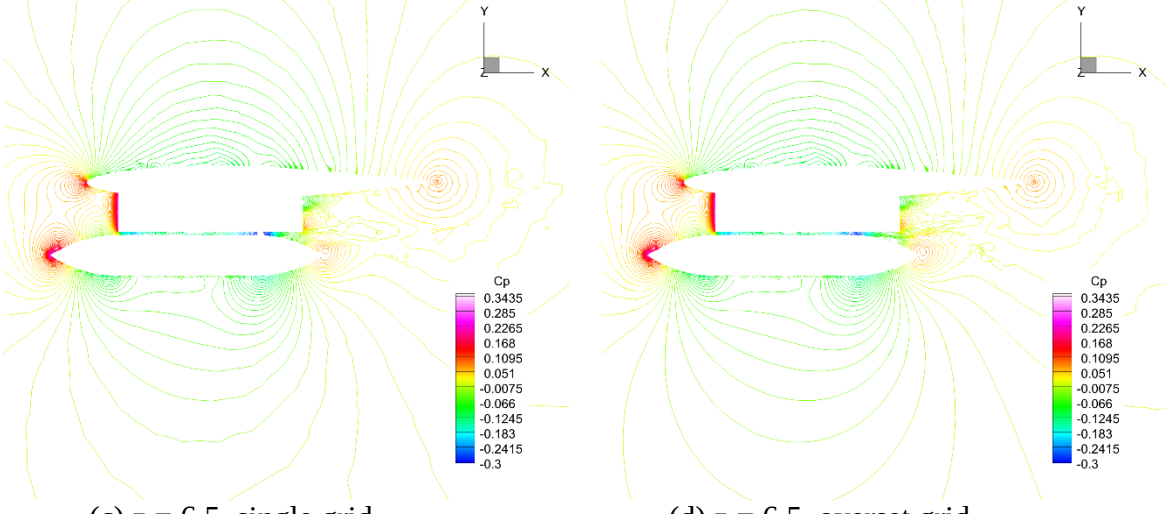


(f) Overset grid

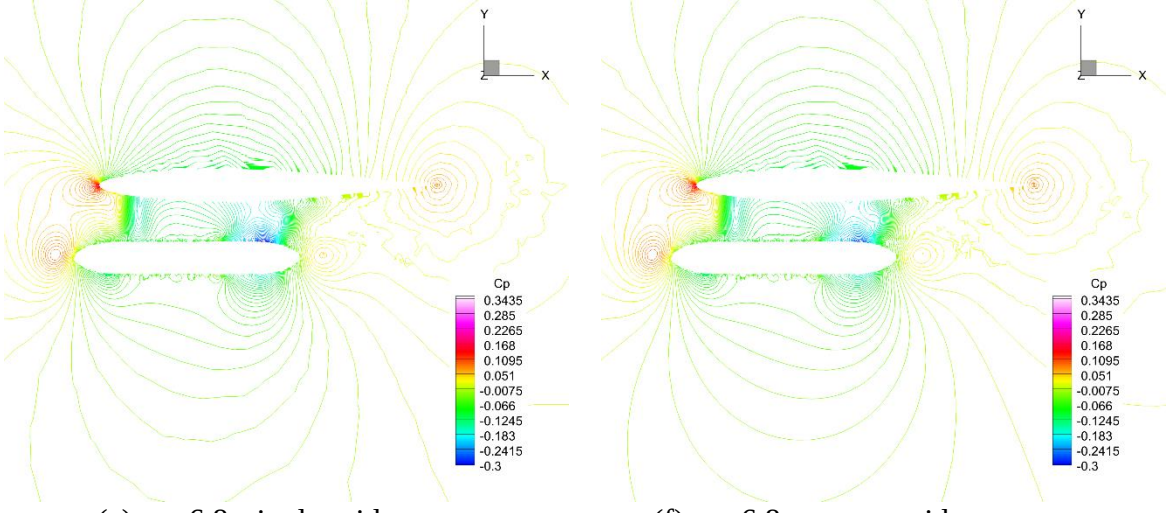
Figure V.12 C_p on the surface of the WPFS geometries



(a) z = 6.2, single grid (b) z = 6.2, overset grid

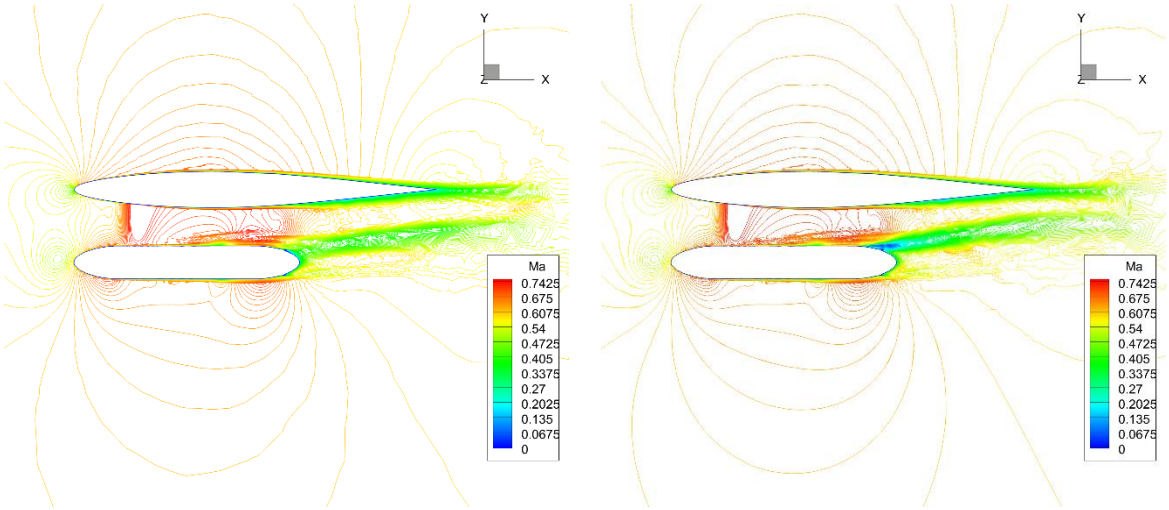


(c) z = 6.5, single grid (d) z = 6.5, overset grid



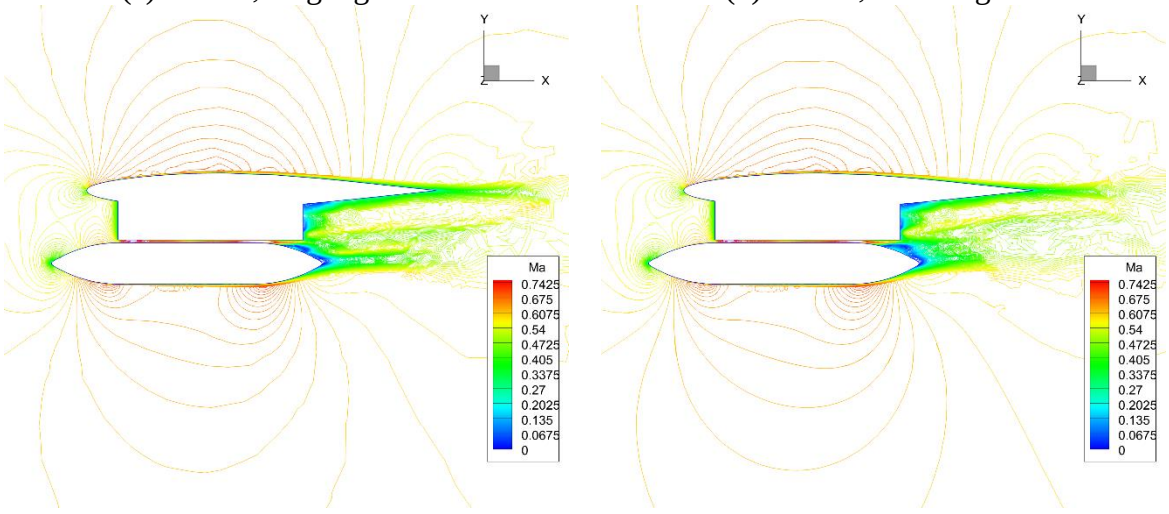
(e) z = 6.8, single grid (f) z = 6.8, overset grid

Figure V.13 C_p contours on various span-wise locations



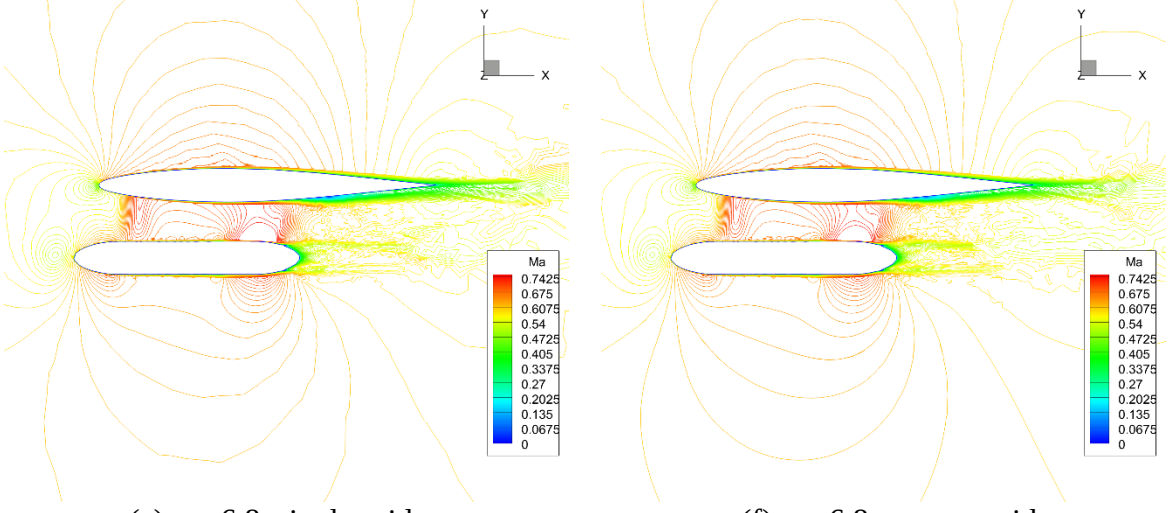
(a) z = 6.2, single grid

(b) z = 6.2, overset grid



(c) z = 6.5, single grid

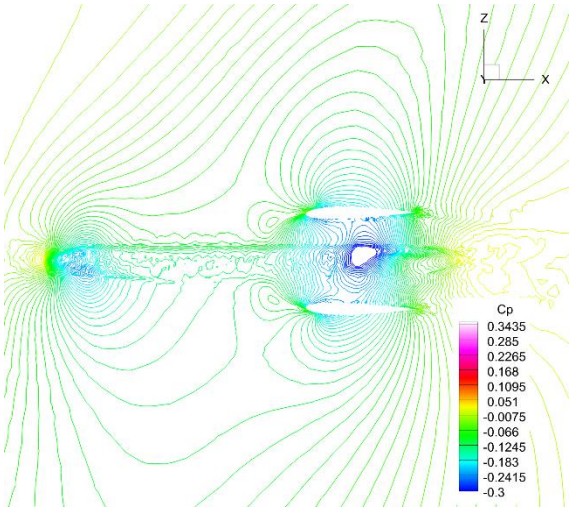
(d) z = 6.5, overset grid



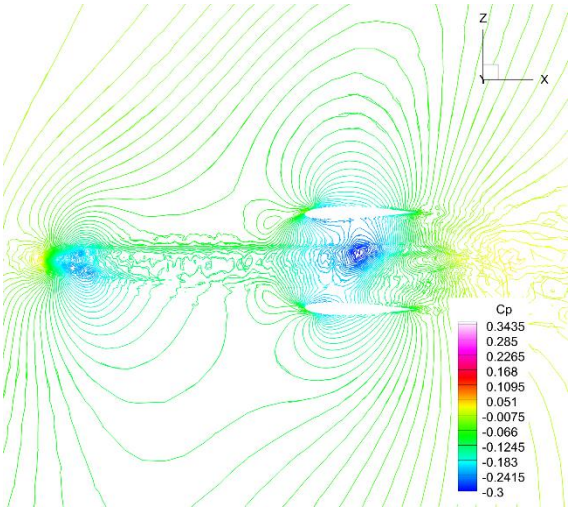
(e) z = 6.8, single grid

(f) z = 6.8, overset grid

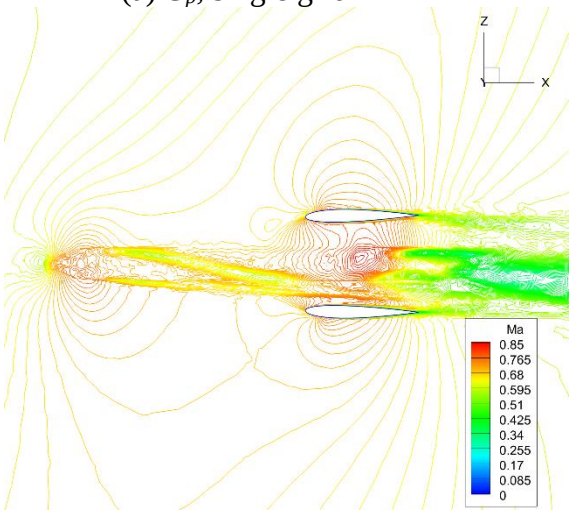
Figure V.14 Mach contours on various span-wise locations



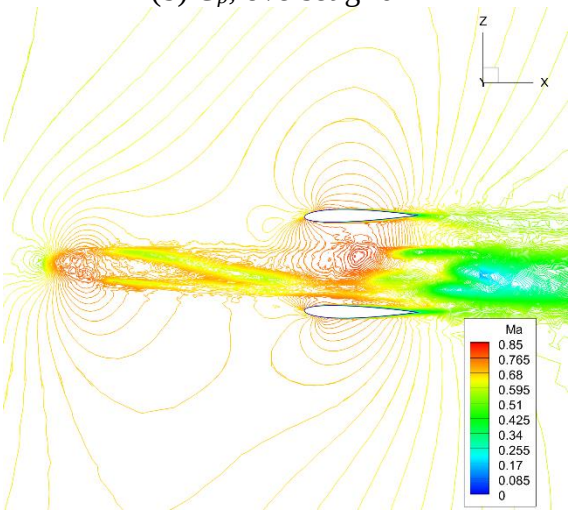
(a) C_p, single grid



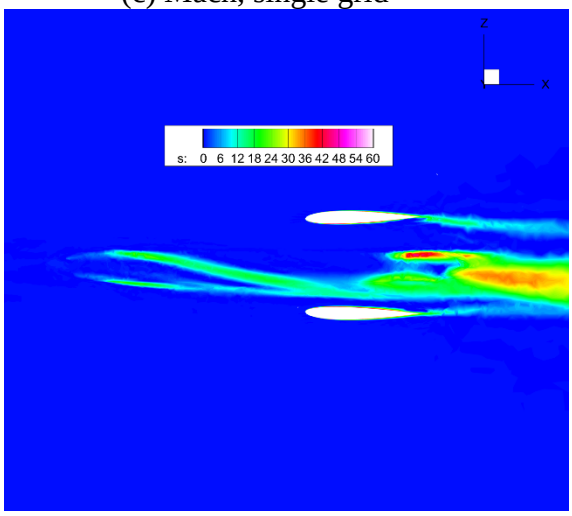
(b) C_p, overset grid



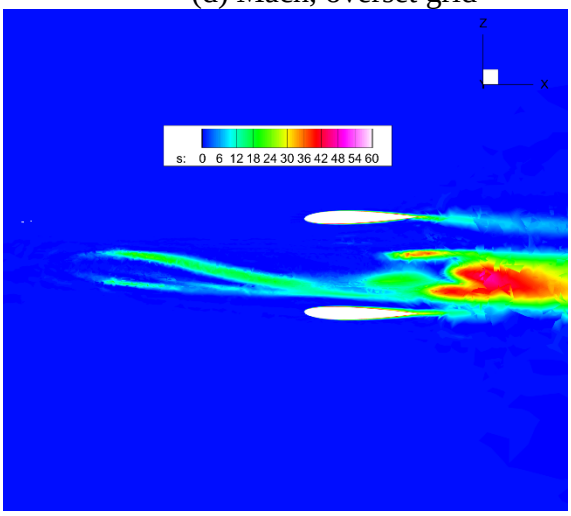
(c) Mach, single grid



(d) Mach, overset grid



(e) Entropy, single grid



(f) Entropy, overset grid

Figure V.15 Mach, C_p and entropy contours in the gap between the pylon and store (y = -1.24)

Figure V.16 shows C_p plots at various azimuthal locations of the store. $\phi = 0°$ and $\phi = 180°$ locations are the top and bottom of the store, respectively, $\phi \in (0°, 180°)$ is the outboard side of the store, and $\phi \in (180°, 360°)$ is the inboard side of the store. Again excellent agreement is found in most azimuthal locations. Some discrepancies at locations $\phi = 0°$ and $\phi = 330°$, as shown in Figure V.16(a, l), indicate the larger separation area towards the rear part of the store predicted by the overset simulation.

Figure V.17 illustrates C_p plots at various y-locations of the pylon. The location y = -0.45 is towards the wing, y = -0.8 is approximately the middle of the pylon, and y = -1.15 is towards the pylon-store gap. Figure V.18 shows C_p plots at various z-locations (span-wise locations) of the wing. The location z = 6.2 and z = 6.8 are 0.3 from the store centerline towards the inboard and outboard of the wing, respectively. Again excellent agreement is observed between two simulations. Note, the oscillations in the C_p plots for both the single and overset grid simulations are due to the linear representations of the geometries while using quadratic elements for the solutions.

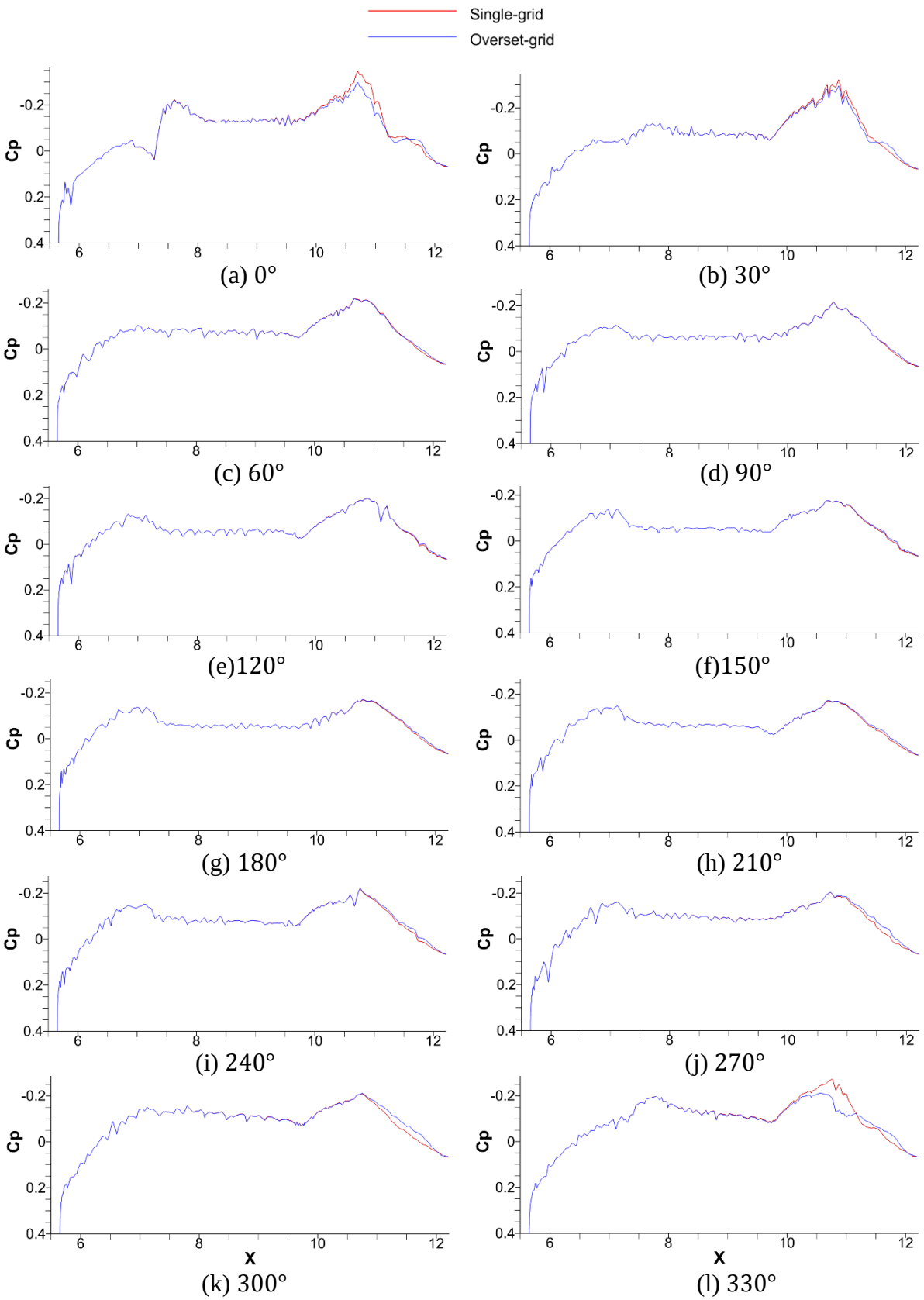
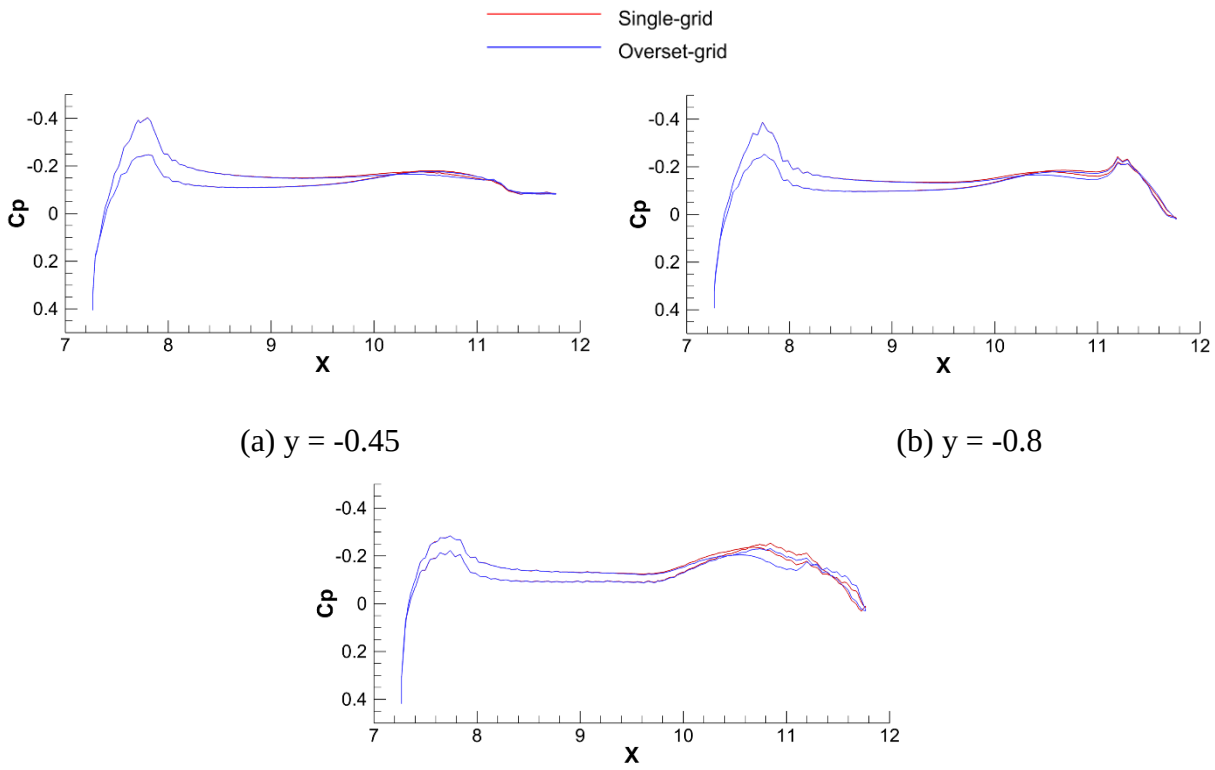


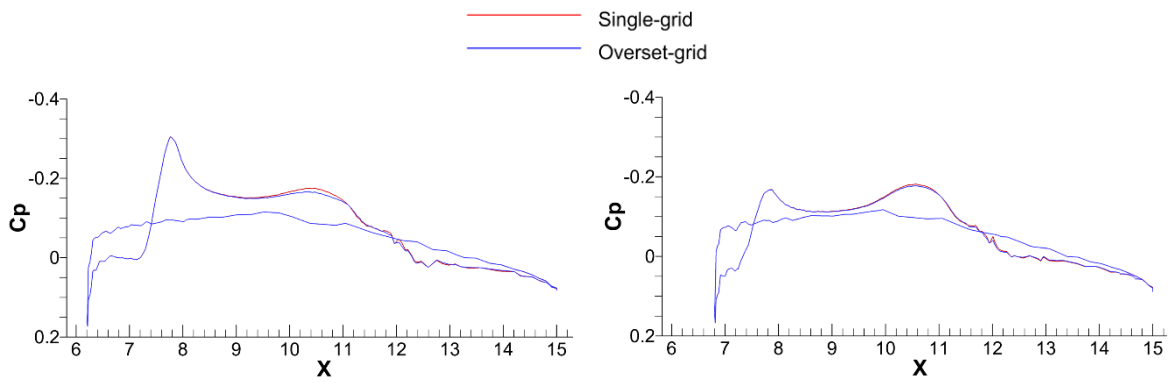
Figure V.16 C_p plots at various azimuthal locations on the store



(a) y = -0.45 (b) y = -0.8

(c) y = -1.15

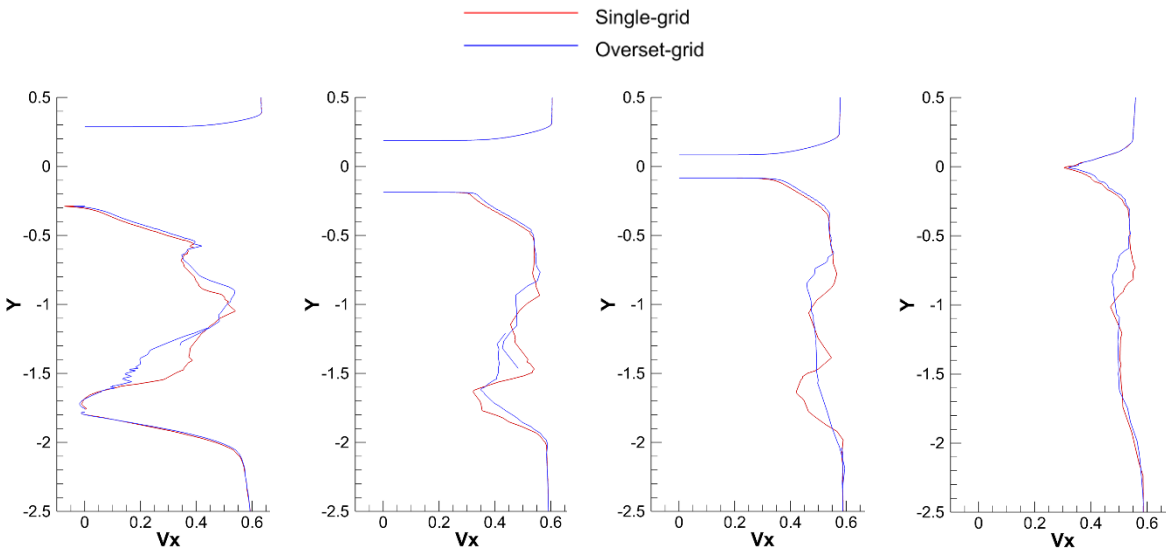
Figure V.17 C_p plots on inboard/outboard sides of the pylon



(a) z = 6.2 (b) z = 6.8

Figure V.18 C_p plots at various span-wise locations on the wing

Figure V.19 shows the x-velocity profiles along y-direction at various x-locations downstream of the store. Note that the wing is installed at the y = 0 plane, and the store is installed at the y = -1.77 plane, and the trailing edge of the store is at location (x, y, z) = (12.22, -1.77, 6.5). As show in the plots, the main discrepancies of the x-velocity profiles occur between the wing and the store. For locations of x = 12.22 and x = 13.22, the discrepancies are again due to the larger separation area on the store predicted by the overset simulation. Once again, as is evident from Figure V.9(f), for other downstream locations the coarser spacing (overset grid for the store has a coarser spacing downstream of the store than the single grid), as well as the disparities in cell sizes between the wing grid and the store grid at the overset boundaries, are also the causes of the discrepancies in the velocity profiles.



(a) x = 12.22 (b) x = 13.22 (c) x = 14.22 (d) x = 15.22

Figure V.19 x-velocity profiles along y-direction at various x-locations downstream of the store

V.2.2 Sinusoidally Pitching Wing

The forced pitching oscillation of an ONERA M6 wing is simulated to examine the three-dimensional overset methodology for moving boundary problems. The wing is pitching about the z-axis located at approximately 60% of the root chord length, and is analytically prescribed as

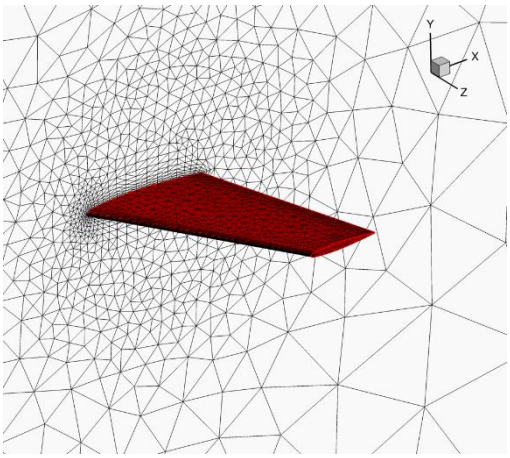
$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t) \tag{V.3}$$

where the mean angle-of-attack α_m and amplitude-of-oscillation α_0 are 2.89° and 2.41°, respectively. The non-dimensional circular frequency ω is defined in terms of the reduced frequency k and the freestream Mach number M_∞ as $\omega = 2kM_\infty$. In the present results, k and M_∞ are 0.0808 and 0.6, respectively. Furthermore, analytic grid velocities are utilized based on the prescribed motion, and the non-dimensional time step is selected to have approximately 75 time instances per pitching cycle.

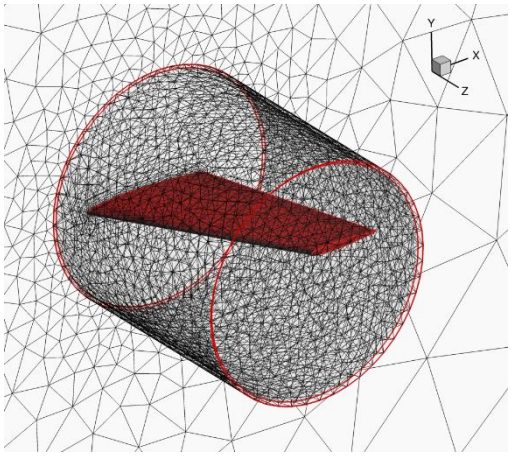
The single grid and overset grids used for comparisons are depicted in Figure V.20. Two configurations of overset grids are investigated. The first configuration in Figure V.20(b) has a pre-cut hole in the stationary background grid and, thus, hole cutting is not performed for this configuration. The second overset configuration in Figure V.20(c) has no hole in the background grid, and dynamic hole cutting is performed at each time instance. In all simulations, linear P_1 tetrahedral elements are utilized.

Note that for the second overset configuration, although the overset boundary of the wing grid is a perfect cylinder, the axis of the cylinder is not the rotational axis of the wing grid. Moreover, the background grid also rotates clock-wise about the same rotational axis as the wing grid at approximately 6° per time step. Therefore, different overlapping between the wing grid and background grid occurs at each time instance and, thus, different nodes are excluded from

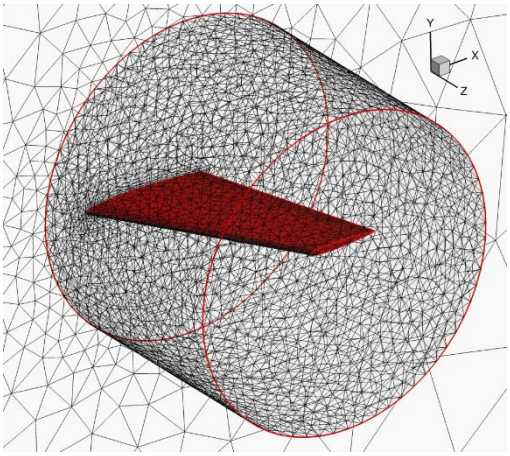
simulation at each time instance. This configuration is a quantitative examination of the potential error introduced by the solution re-initialization approach described in section III.5. Also note that, although the invalid cells have not been blanked out, as shown in Figure V.21, these cells are isolated from the rest of the cells included in the simulation and, thus, will not influence the computed value of the lift coefficient.



(a) Single grid



(b) Overset grids, pre-cut



(c) Overset grids, dynamic cut

Figure V.20 Single and overset grids used for the sinusoidally pitching ONERA M6 wing

Shown in Figure V.22 are the time histories of the coefficient of lift for the single grid and the two overset grid simulations. As seen excellent agreement is observed between the different configurations and, therefore, indicates that for this simulation the accuracy is not diminished by using the overset grid approach or the approach for solution re-initialization.

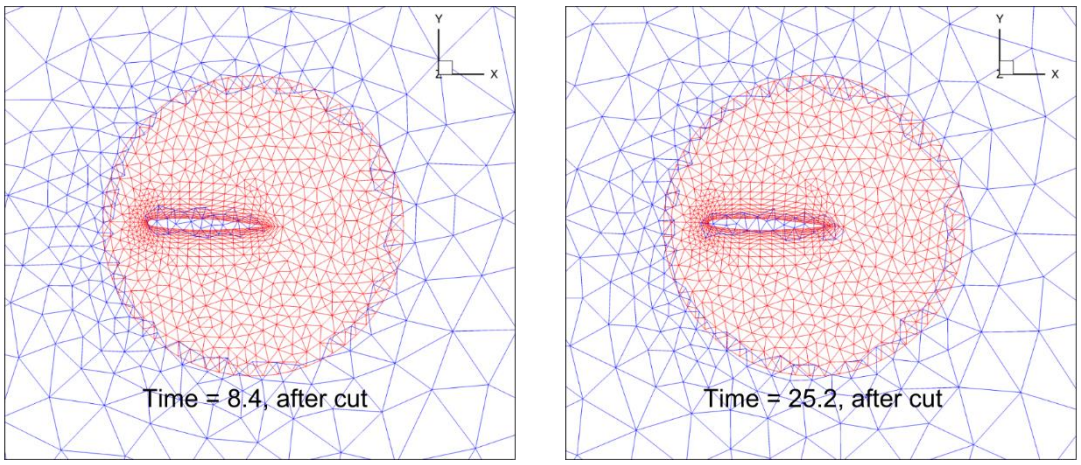


Figure V.21 Overset grids for the ONERA M6 wing after hole cutting at various time instances

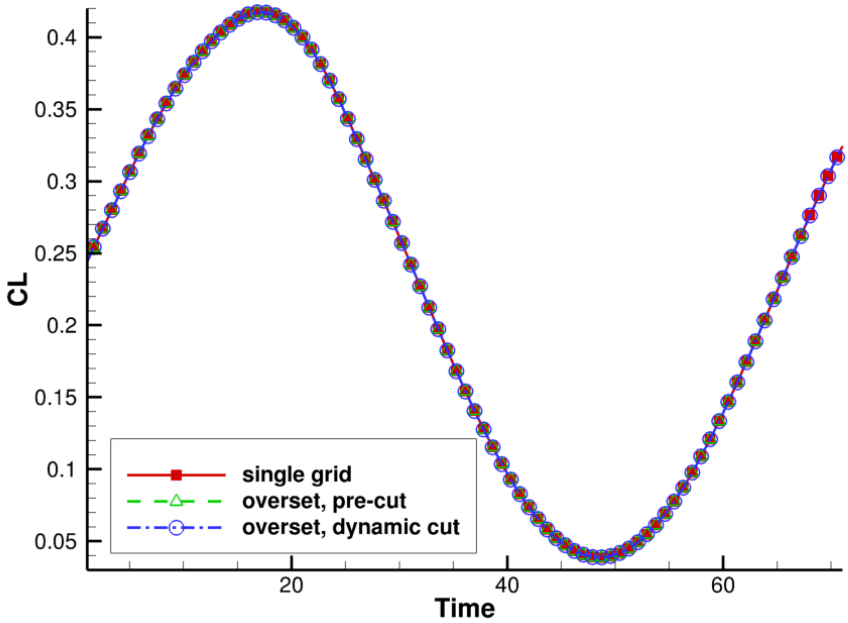


Figure V.22 Time histories of C_L for the sinusoidally pitching ONERA M6 wing

V.2.3 Laminar Wing/Finned-Store Separation

One of the primary interests for the development of overset grid capabilities is the simulation of unsteady moving boundary problems such as store separation. This case simulates the unsteady viscous flow for a wing/finned-store (WFS) configuration, in which the store has been released and is moving in a prescribed motion. The dimensions of the WFS configuration is depicted in Figure V.23, noting that the geometry of the store is the same as that of the WPFS configuration used in section V.2.1. The wing once again has 45° of leading edge sweep and a NACA-64A010 airfoil section. However, it has a different span from the WPFS configuration.

The flow is assumed laminar, with a Reynolds number of 10^3. The freestream has zero degree of angle-of-attack, and Mach number of 0.6. The prescribed trajectory of the store is depicted in Figure V.24.

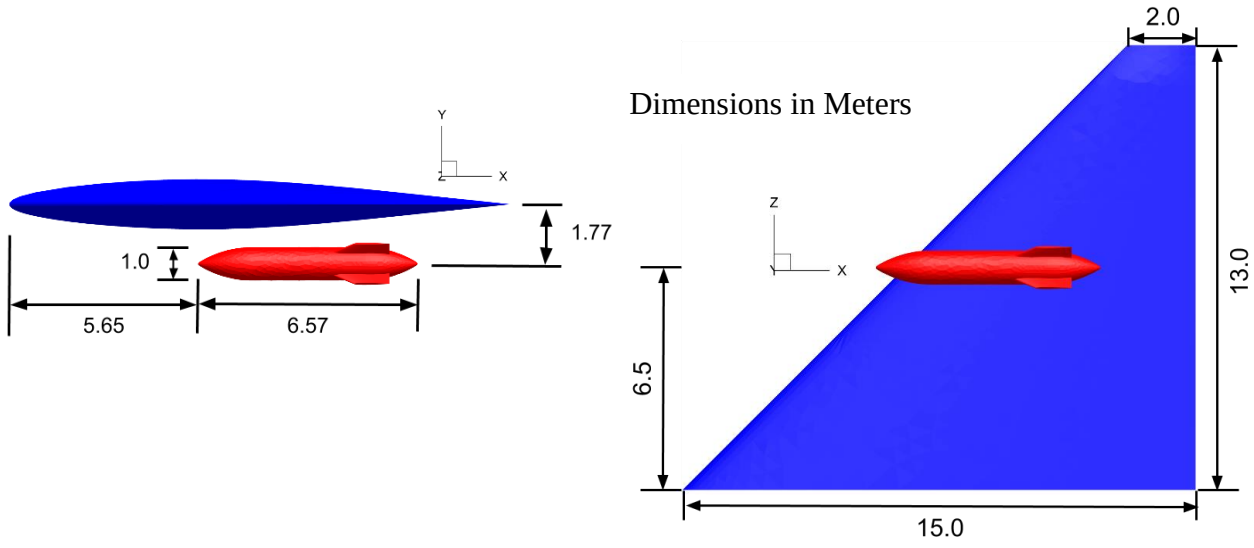


Figure V.23 The WFS geometries

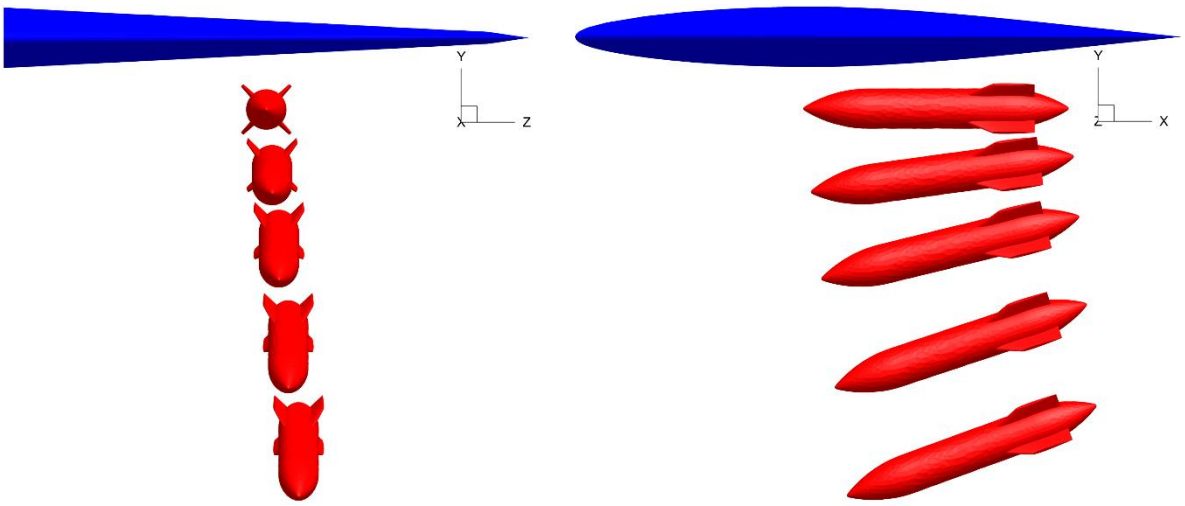


Figure V.24 Prescribed trajectory of the store

The same meshing strategy as the WPFS case is used for the WFS case. As shown in Figure V.25, a body fitted grid is generated for the wing, which extends to the far field. Another body fitted grid is generated for the finned-store. Both grids have a viscous spacing of 2×10^{-2} normal to the wall. Quadratic P_2 tetrahedral elements are used in the simulation, but all element boundaries are once again linear. That is, no mesh curving is performed. The grids contain 365,125 P_2 tetrahedral elements and 506,030 nodes, and the simulation is carried out using 68 CPU cores.

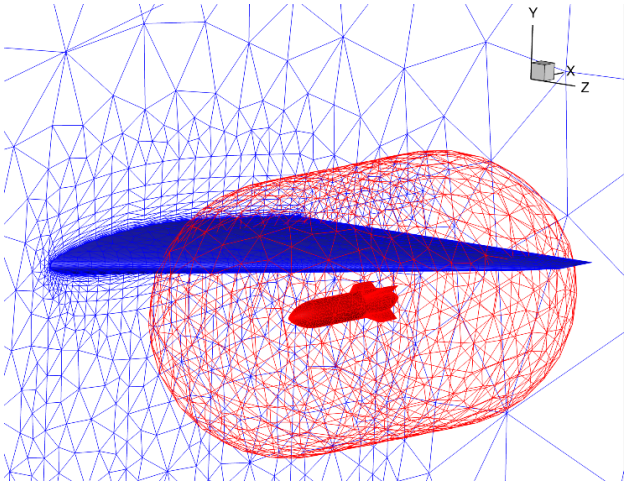
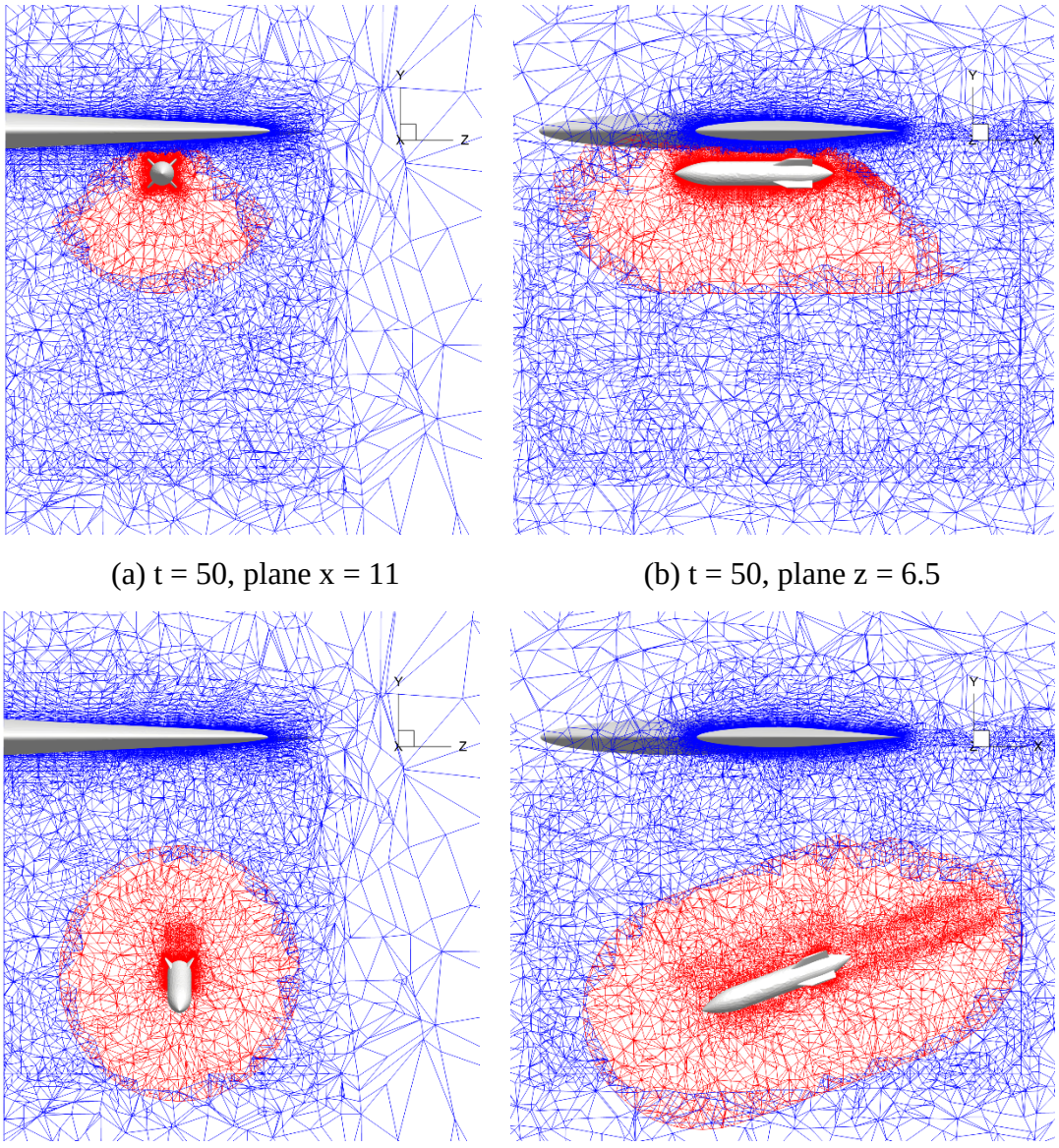


Figure V.25 Mesh for the WFS configuration

The overset grids after hole cutting for the first and last time instance of the simulation are depicted in Figure V.26. Note that like the WPFS case, each quadratic tetrahedral element is divided into four linear tetrahedral and two linear pyramid elements for visualization purpose.



(a) t = 50, plane x = 11 (b) t = 50, plane z = 6.5

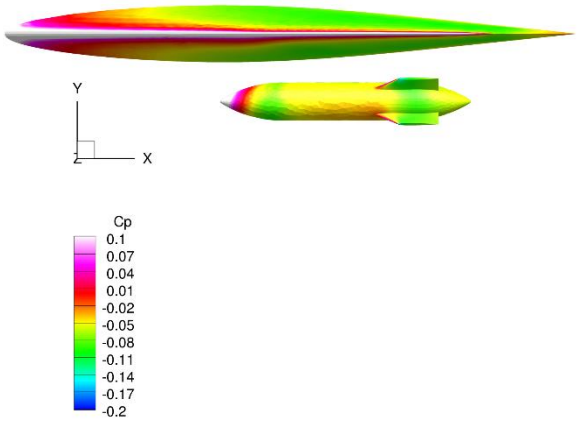
(c) t = 2000, plane x = 11 (d) t = 2000, plane z = 6.5

Figure V.26 Overset grids after hole cutting at various time instances

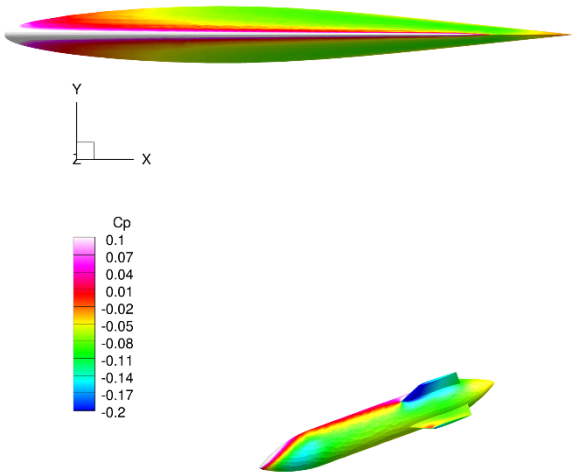
The reference length L_{ref} is chosen to be the diameter of the store, which is 1 meter, the reference velocity V_{ref} is the speed of sound in the freestream, and the reference time is defined as $T_{ref} = L_{ref}/V_{ref}$. The non-dimensionalized time step used in this unsteady simulation is 10, and 200 time iterations are computed, with the non-dimensionalized time range of $t \in (0, 2000)$. A steady-state simulation of the flow about the WFS geometries with the store in captive position is carried out and used as the initial solution $(t = 0)$ for the unsteady wing/store separation simulation. The store rotates about the z-axis at 71% of the centerline of the store. The translating and rotating motion of the store is prescribed as a function of the non-dimensional time t as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = 0.5 * \begin{pmatrix} 4 \times 10^{-7} \\ -4 \times 10^{-6} \\ 4 \times 10^{-7} \end{pmatrix} * t^2$$
$$angle = \ 5 \times 10^{-9} * t^3 - 15 \times 10^{-6} * t^2 \qquad \text{(V.4)}$$

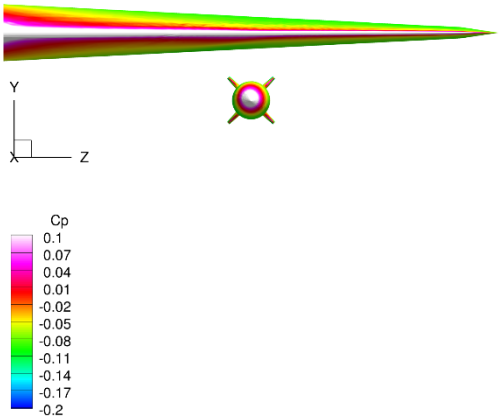
Figure V.27 depicts the shaded C_p contours on the WFS geometries from different vantage points at two time instances during the store motion. Figure V.28 shows C_p and Mach contours at various time instances on a stream-wise plane (x = 11) passing through the wing and the fins of the store. The interference effects between the fined store and the wing can be clearly seen in the earlier time instances. Illustrated in Figure V.29 are, once again, the C_p and Mach contours, however, at span-wise planes inboard (z = 6.5), middle (z = 6.89), and outboard (z = 7.3) of the store centerline at various time instances during the motion. At all the time instances, smooth transitions of the contours across the overset boundaries are observed.



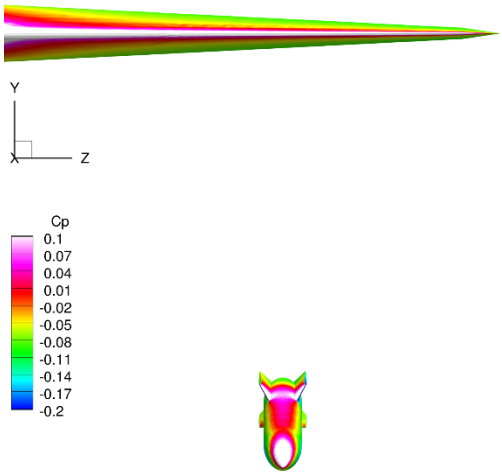
(a) t = 50, -z view



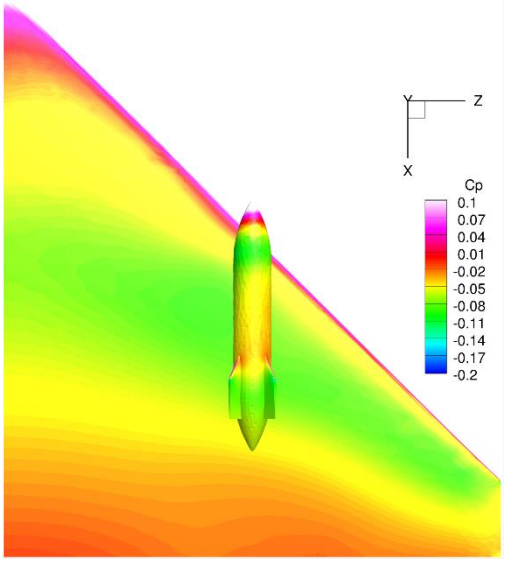
(b) t = 2000, -z view



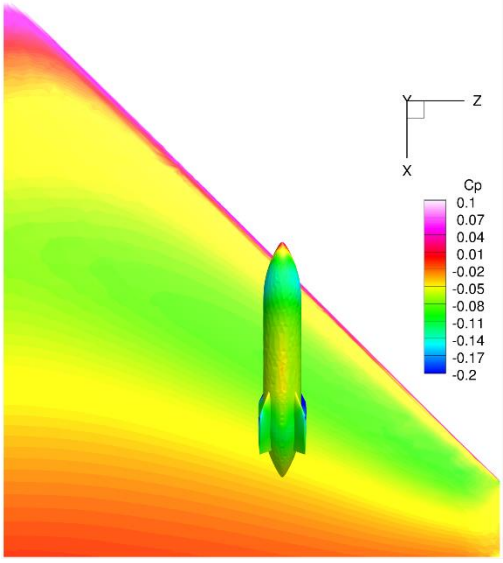
(c) t = 50, +x view



(d) t = 2000, +x view

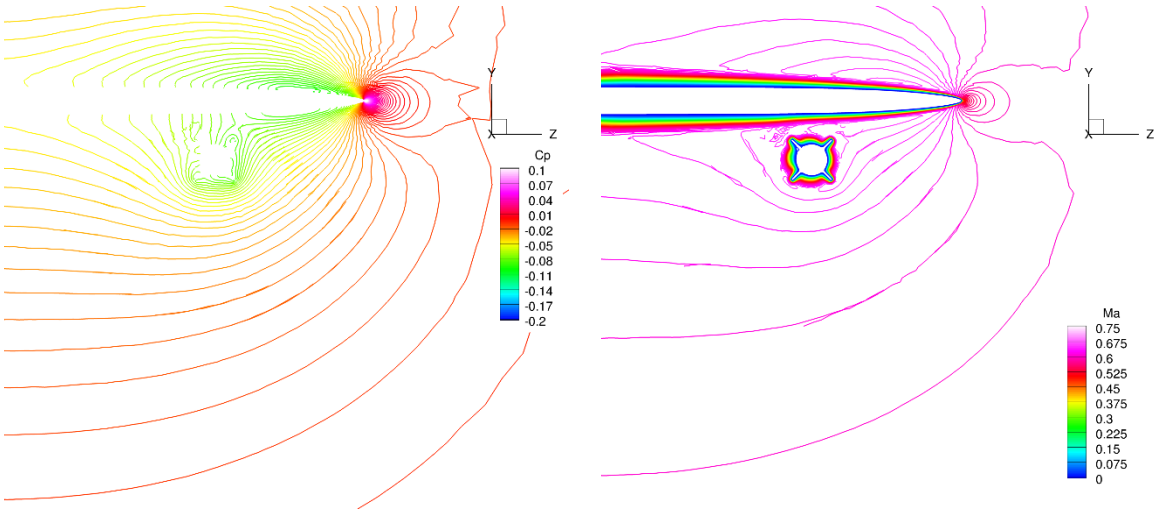


(e) t = 50, +y view



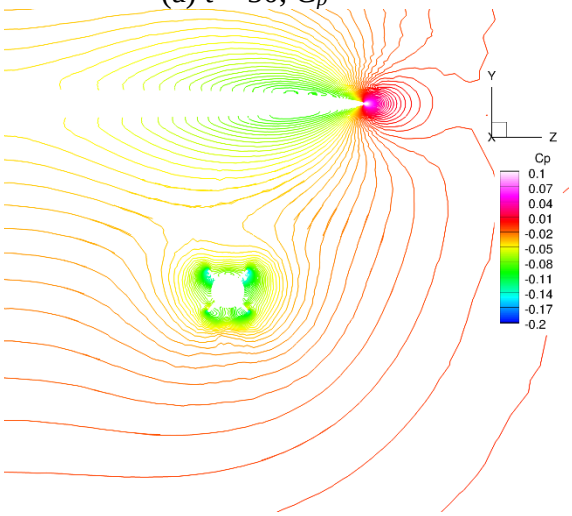
(f) t = 2000, +y view

Figure V.27 C_p on the surface of the wing and store at various time instances

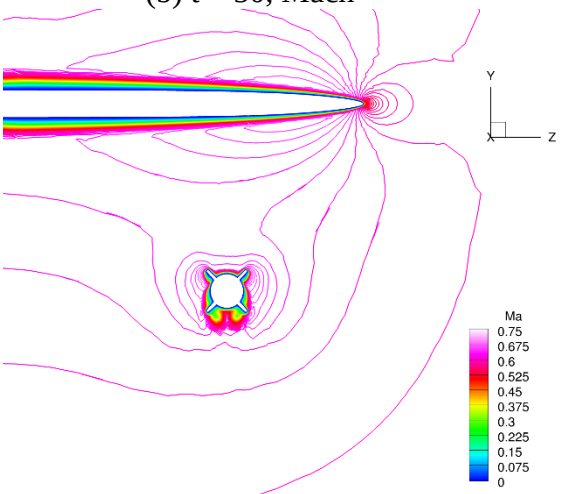


(a) t = 50, C_p

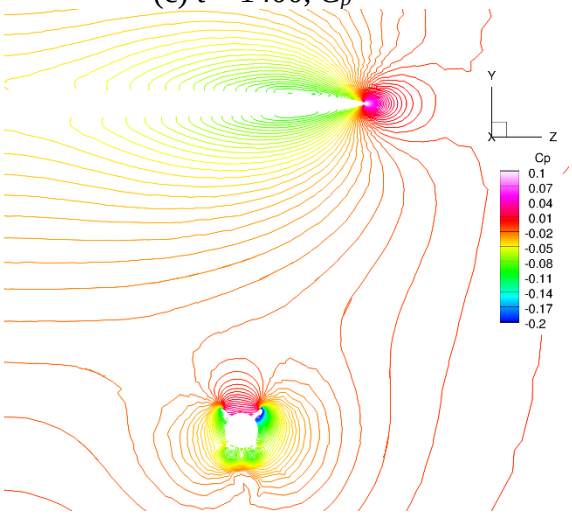
(b) t = 50, Mach



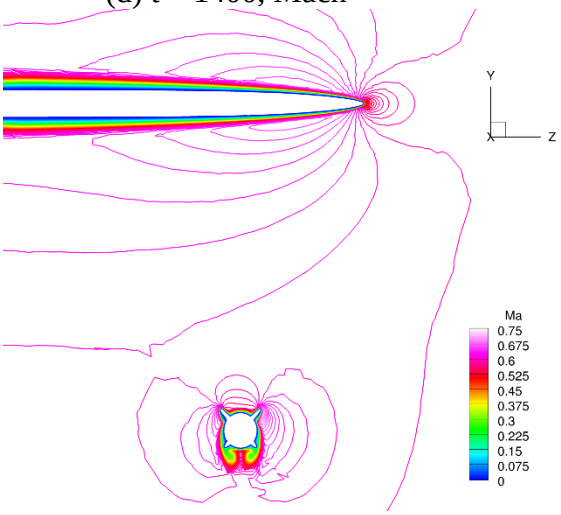
(c) t = 1400, C_p



(d) t = 1400, Mach

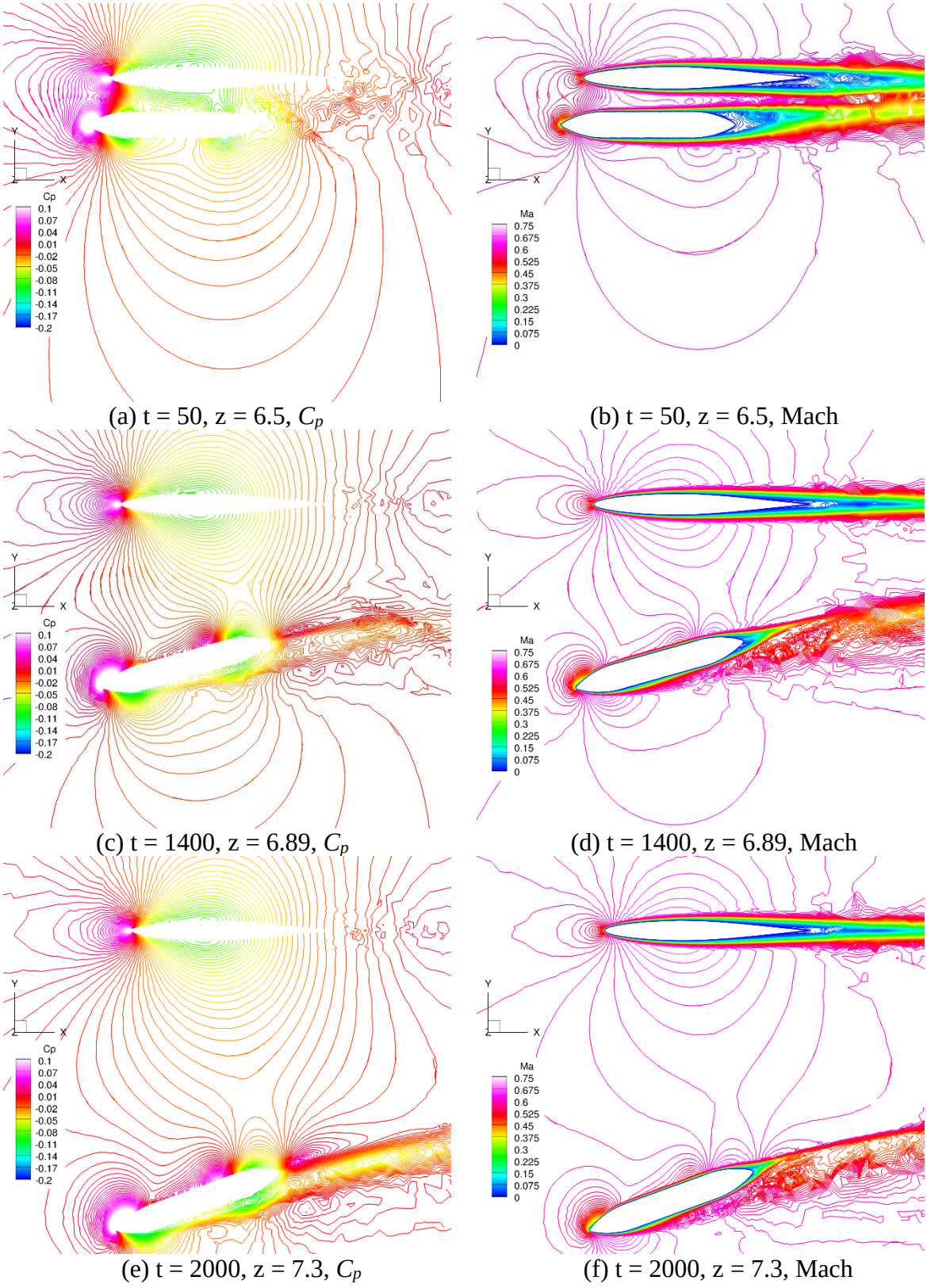


(e) t = 2000, C_p



(f) t = 2000, Mach

Figure V.28 C_p and Mach contours on plane x = 11 at various time instances



(a) t = 50, z = 6.5, C_p

(b) t = 50, z = 6.5, Mach

(c) t = 1400, z = 6.89, C_p

(d) t = 1400, z = 6.89, Mach

(e) t = 2000, z = 7.3, C_p

(f) t = 2000, z = 7.3, Mach

Figure V.29 C_p and Mach contours on various z planes at various time instances

CHAPTER VI

CONCLUSION

VI.1 Summary and Contributions

In this research, the viability of using an overset grid approach within a streamline/upwind Petrov-Galerkin formulation for unsteady, viscous, moving boundary simulations has been established. Using the method of manufactured solutions, the order of accuracy has been verified up to fourth order for inviscid and laminar flow in the two-dimensional implementation, and up to third order for inviscid flow in the three-dimensional framework.

A MPI-based parallel three-dimensional overset grid assembly framework has been developed. Additionally, a novel cell selection approach named elliptic hole cutting (EHC) based on solutions to the Poisson equation has been introduced. Like implicit hole cutting, EHC is able to yield a mesh in favor of certain mesh quality metrics by devised corresponding source terms for the Poisson equation. However, the cell selection for EHC is continuous, and similar to the approach based on the distance function. Additionally, if parallelization is needed, the modified EHC is a viable choice.

Two modifications to the ILU(k) preconditioner, which consider both intra-grid and inter-grid linearization and are referred to as “triangular inter-grid ILU(k)” and “Jacobi inter-grid ILU(k)” respectively, have been constructed. Compared with the intra-grid ILU(k), which only considers the intra-grid linearization, both of these modifications effectively reduce the number of

GMRES search directions. This reduction indicates the importance of considering inter-grid linearization in the preconditioner. The "Jacobi inter-grid ILU(k)", which utilizes an intra-grid ILU(k) inside Jacobi iterations, can be extended as a parallel preconditioner on distributed memory machines. This may be viewed as making the analogy between intra/inter-grid linearization with intra/inter-computational-node linearization, due to the similarity in the structures of their Jacobian matrices.

The potential benefits of using adaptation in overset grid simulations are explored by combining the overset with an adaptive mesh refinement approach. As discussed, the need to resolve flow phenomena that propagate in an unsteady simulation, when the movement of the bodies is not known a priori, and/or when grids have disparate cell volumes at the overset boundaries, the ability to adaptively refine is critical for solution accuracy.

Finally, overset grid results are presented for several inviscid and viscous, steady-state and time-dependent moving boundary simulations with linear, quadratic, and cubic elements. For the two-dimensional cases and the three-dimensional WPFS case, verification of the overset grid results are performed by comparing with single grid solutions. The single grid solver has been previously validated in the literature. The two-dimensional triangular edge passing an airfoil, and the store separation in the three-dimensional WFS case, exemplify examples where overset capabilities are required to perform the simulation.

VI.2 Recommendations for Future Work

1) The test cases in this research do not involve strong discontinuities in the flow field. Extending the current overset grid capabilities with shock-capturing schemes would significantly expand the range of applications.

2) For the overset grids with zero overlapping, the overset interface has different discretizations from the grids on two sides of the interface. The numerical fluxes integrated on the two sides of the interface will not be locally conservative. While the order of accuracy of the solutions is confirmed by the method of manufactured solutions, the mechanism for global conservation is not completely understood. It would be interesting to investigate the order of accuracy of the numerical fluxes on the overset boundaries in the zero overlapping situations, and develop a numerical flux integration method for overset boundaries that guarantees desired order of accuracy of flux boundary conditions.

3) Due to the difference in the computation and communication pattern between the flow solver and the overset grid assembly, domain decomposition is difficult, if not impossible, to be load-balanced for the flow solver and overset grid assembly at the same time. Furthermore, for overset grid simulations of moving bodies, the initial domain decomposition is certain to create load imbalance over time. A dynamic load balancing framework for distributed memory machines would be a valuable asset.

4) Develop mesh morphing capability for stabilized finite element formulation, and combine with the overset approach, for the simulations of fluid structure interactions with the structures deforming elastically.

5) Using h-, p- and hp-adaptation in combination with overset grid simulations offers many benefits. In a dynamic overset moving body simulation without mesh adaptation, flow

phenomena will evolve with time with inadequate resolution, and the disparities between grids may become pronounced. Extending the current three-dimensional overset capabilities with adaptive refinement may mitigate these issues.

REFERENCES

1. Benek, J. A., Steger, J. L., and Dougherty, F. C. "A Flexible Grid Embedding Technique with Application to the Euler Equations," *6th Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 1983.

2. Steger, J. L., Dougherty, F. C., and Benek, J. A. "A Chimera Grid Scheme," *Advances in Grid Generation*, Vol. 5, 1983, pp. 59-69.

3. Benek, J. A., Buning, P. G., and Steger, J. L. "A 3-D Chimera Grid Embedding Technique," *7th Computational Physics Conference*. American Institute of Aeronautics and Astronautics, 1985.

4. Robert, M. "Computations of the Unsteady Flow about a Generic Wing/Pylon/Finned-Store Configuration," *Astrodynamics Conference*. American Institute of Aeronautics and Astronautics, 1992.

5. Robert, M., and Andrew, W. "Unsteady Aerodynamic Simulation of Static and Moving Bodies using Scalable Computers," *14th Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 1999.

6. Wey, T. C. "The Applications of an Unstructured Grid Based Overset Grid Scheme to Applied Aerodynamics," *Eighth International Mesh Round Table*. South Lake Tahoe, CA, 1999, pp. 163-169.

7. Fumiya, T., Kazuhiro, N., Yasushi, I., Toshiyuki, I., and Yuichi, S. "Flow Simulation of NAL Experimental Supersonic Airplane/Booster Separation using Overset Unstructured Grids," *38th Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2000.

8. Rainald, L., Dmitri, S., Hong, L., and Ravi, R. "Overlapping Unstructured Grids," *39th Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2001.

9. Ralph, N., and David, B. "Improvements to SUGGAR and DiRTlib for Overset Store Separation Simulations," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2009.

10. Rogers, S. E., Suhs, N. E., and Dietz, W. E. "PEGASUS 5: An Automated Preprocessor for Overset-Grid Computational Fluid Dynamics," *AIAA Journal*, Vol. 41, No. 6, 2003, pp. 1037-1045. doi: 10.2514/2.2070

11. Eliot, W. Q., and Marilyn, S. "Advanced Data Transfer Strategies for Overset Computational Methods," *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2015.

12. Cristian, N., Dimitri, M., and Jay, S. "An Overset Unstructured Mesh Discontinuous Galerkin Approach for Aerodynamic Problems," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2011.

13. Galbraith, M. C., Orkwis, P. D., and Benek, J. A. "Extending the Discontinuous Galerkin Scheme to the Chimera Overset Method," *20th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2011.

14. Galbraith, M. C., Benek, J. A., Orkwis, P. D., and Turner, M. G. "A Discontinuous Galerkin Chimera Scheme," *Computers & Fluids*, Vol. 98, 2014, pp. 27-53. doi: http://dx.doi.org/10.1016/j.compfluid.2014.03.014

15. Galbraith, M., Knapke, R., Orkwis, P., and Benek, J. "A Discontinuous Galerkin Chimera Scheme with Implicit Artificial Boundaries," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2013.

16. Galbraith, M. C., Orkwis, P. D., and Benek, J. A. "Hole Cutting of Curved Discontinuous Galerkin Chimera Overset Meshes using a Direct Cut Method," *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2013.

17. Cuong, N., Moro, D., and Jaime, P. "Navier-Stokes Solution using Hybridizable Discontinuous Galerkin Methods," *20th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2011.

18. Allmaras, S. R., Johnson, F. T., and Spalart, P. R. "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," *Seventh International Conference on Computational Fluid Dynamics*. Big Island, Hawaii, 2012.

19. Erwin, J. T., Wang, L., Anderson, W. K., and Kapadia, S. "High-Order Finite-Element Method for Three-Dimensional Turbulent Navier-Stokes," *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2013.

20. Masud, A., and Hughes, T. J. R. "A Space-Time Galerkin/Least-Squares Finite Element Formulation of the Navier-Stokes Equations for Moving Domain Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 146, No. 1–2, 1997, pp. 91-126. doi: http://dx.doi.org/10.1016/S0045-7825(96)01222-4

21. van der Vegt, J. J. W., and van der Ven, H. "Space–Time Discontinuous Galerkin Finite Element Method with Dynamic Grid Motion for Inviscid Compressible Flows: I. General Formulation," *Journal of Computational Physics*, Vol. 182, No. 2, 2002, pp. 546-585. doi: http://dx.doi.org/10.1006/jcph.2002.7185

22. Tezduyar, T. E. "Finite Element Methods for Fluid Dynamics with Moving Boundaries and Interfaces," *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004.

23. Donea, J., Giuliani, S., and Halleux, J. P. "An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Interactions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 33, No. 1, 1982, pp. 689-723. doi: http://dx.doi.org/10.1016/0045-7825(82)90128-1

24. Persson, P. O., Bonet, J., and Peraire, J. "Discontinuous Galerkin Solution of the Navier–Stokes Equations on Deformable Domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 17–20, 2009, pp. 1585-1595. doi: http://dx.doi.org/10.1016/j.cma.2009.01.012

25. Dimitri, M., and Cristian, N. "On the Geometric Conservation Law for High-Order Discontinuous Galerkin Discretizations on Dynamically Deforming Meshes," *46th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2008.

26. Gressier, J., Villedieu, P., and Moschetta, J.-M. "Positivity of Flux Vector Splitting Schemes," *Journal of Computational Physics*, Vol. 155, No. 1, 1999, pp. 199-220. doi: http://dx.doi.org/10.1006/jcph.1999.6337

27. Wang, C., and Liu, J. "Positivity Property of Second-Order Flux-Splitting Schemes for the Compressible Euler Equations," *Discrete and Continuous Dynamical Systems - Series B*, Vol. 3, No. 2, 2003, pp. 201-228. doi: 10.3934/dcdsb.2003.3.201

28. van Leer, B. "Flux-Vector Splitting for the Euler Equations," *Eighth International Conference on Numerical Methods in Fluid Dynamics: Proceedings of the Conference, Rheinisch-Westfälische Technische Hochschule Aachen, Germany, June 28 – July 2, 1982*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982, pp. 507-512.

29. Anderson, W. K., Thomas, J. L., and Van Leer, B. "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal*, Vol. 24, No. 9, 1986, pp. 1453-1460. doi: 10.2514/3.9465

30. Anderson, W. K., Thomas, J. L., and Rumsey, C. L. "Extension and Application of Flux-Vector Splitting to Calculations on Dynamic Meshes," *AIAA Journal*, Vol. 27, No. 6, 1989, pp. 673-674. doi: 10.2514/3.10163

31. Saad, Y., and Schultz, M. H. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856-869. doi: doi:10.1137/0907058

32. Nakahashi, K., Togashi, F., and Sharov, D. "Intergrid-Boundary Definition Method for Overset Unstructured Grid Approach," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2077-2084. doi: 10.2514/2.869

33. Robert, M. "Object X-Rays for Cutting Holes in Composite Overset Structured Grids," *15th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2001.

34. YikLoon, L., and James, B. "Implicit Hole Cutting - A New Approach to Overset Grid Connectivity," *16th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2003.

35. Roget, B., and Sitaraman, J. "Robust and Efficient Overset Grid Assembly for Partitioned Unstructured Meshes," *Journal of Computational Physics*, Vol. 260, 2014, pp. 1-24. doi: http://dx.doi.org/10.1016/j.jcp.2013.12.021

36. Druyor, C. T., Karman, S. L., and Jones, W. "A Survey of Overset Domain Assembly Methods," *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2015.

37. Erwin, J. T. "Stabilized Finite Elements for Compressible Turbulent Navier-Stokes." Ph.D. Dissertation, University of Tennessee at Chattanooga, Chattanooga, TN, 2013.

38. Noack, R. "A Direct Cut Approach for Overset Hole Cutting," *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2007.

39. Ahrabi, B. R., Anderson, W. K., and Newman, J. C. "High-Order Finite-Element Method and Dynamic Adaptation for Two-Dimensional Laminar and Turbulent Navier-Stokes," *32nd AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 2014.

40. Galbraith, M. C. "A Discontinuous Galerkin Chimera Overset Solver." Ph.D. Dissertation, University of Cincinnati, Cincinnati, OH, 2013.

41. Robert, M. "On the Spatial and Temporal Accuracy of Overset Grid Methods for Moving Body Problems," *12th Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, 1994.

42. Salari, K., and Knupp, P. "Code Verification by the Method of Manufactured Solutions." Sandia National Laboratories, TR-SAND2000-1444, Albuquerque, NM, 2000.

43. Landon, R. H. "NACA 0012 Oscillatory and Transient Pitching," *Compendium of Unsteady Aerodynamic Measurements*. AGARD-R-702, 1982, pp. 3.3-3.25.

VITA

Chao Liu was born in March of 1986 in Nanchang, Jiangxi, China, to Renji and Xiaoping Liu. He graduated from Nanchang No.1 High School in 2003, where he developed his interest in mechanics. He earned a Bachelor of Science degree in Aerospace Engineering from Beijing University of Aeronautics and Astronautics in 2007. He continued his education at Beijing University of Aeronautics and Astronautics, serving as a graduate research assistant and earned a Master of Science degree in Aerospace Engineering in 2010. In August 2011, Chao moved to Chattanooga, Tennessee, accepting a graduate research assistantship from the University of Tennessee at Chattanooga SimCenter. Chao graduated with a Ph.D. in Computational Engineering in May of 2016.