Title

The Development of a General High-Fidelity Fully Coupled Fluid-Thermal-Structure Interaction Method Specialized for Turbomachinery

Author

Graff, Joseph Steven

Publication Date

2023

Peer reviewed|Thesis/dissertation

The Development of a General High-Fidelity Fully Coupled Fluid-Thermal-Structure Interaction Method Specialized for Turbomachinery

By

Joseph S. Graff
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

MECHANICAL AND AEROSPACE ENGINEERING

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Roger L. Davis, Chair

Mark Rashid

Jean-Pierre Delplanque

Committee in Charge

2023

Abstract

This work details the development effort of the in-house, high-fidelity, fully coupled, fluid-thermal-structure interaction finite volume code, MBFLO3. A three-dimensional finite volume based structural solver capable of simulating arbitrarily large strains has been integrated into the existing fluid-thermal interaction code to develop a monolithic, fully coupled, fluid-thermal-structure interaction technique. The structural solver presented is verified through two test cases using theory, two-dimensional simulations reported in literature, and commercially available software. The fully coupled, monolithic, fluid-structure interaction method is capable of simulating large structural displacements while maintaining mesh quality through the use of a Poisson smoothing technique. The FSI method is verified using Turek and Hron's [63] cylinder-flag configuration. Finally, the MBFLO3 solution procedure has been extended to handle overlaid grids and the conjugate heat transfer modeling of multiple solids using spatial second-order accurate treatments. A three-dimensional, steady, conjugate simulation of a film cooled vane with thermal barrier coating is compared with experimental data from a vane-alone experiment at the Air Force Turbine Research Facility. The numerical techniques used in the conjugate solution procedure are described and the procedure is validated against the vane-alone experimental data.

Contents

List of Figures

List of Tables

Acknowledgements

There are many individuals that have made this research project possible, I share this achievement with the excellent primary school teachers and college professors at my undergraduate and graduate institutions. This work would not have been possible without the encouragement, guidance, wisdom and support I have received from my teachers and professors. This is especially true for Professor Roger Davis, who has been a great mentor throughout my graduate career. He has shared his wisdom, encouragement and patience with me throughout this research project and has helped me grow professionally and personally.

I would like to thank the Air Force Research Laboratory Turbomachinery Branch, Turbine Engine Division, Aerospace Systems Directorate led by Dr. John Clark for their guidance and continued finical support.

I am eternally thankful for the unwavering support I received from my family, particularly my wife, my mother and father and my mother in-law and father in-law. This work would not have been possible without the love and encouragement they gave me. Finally, to my daughter Sophia thank you for bringing me the joy and determination I needed to complete this work. I did this for you.

Nomenclature

β_x	Coefficient of thermal expansion
Δt_G	Global time step
Δt_L	Local time step
δ_{ij}	Dirac delta
$\dot{x}$	Time derivative of variable x
Γ	Continuum domain
Γ_{lk}	Hourglass mode
λ	Bulk viscosity coefficient
λ	Lamé constant
μ	Fluid viscosity
μ	Shear modulus

μ_t Turbulent viscosity

ν Poisson's ratio

ω Specific turbulent dissipation rate

$\overline{\Omega_i}$ Rotational rate

$\overline{M}$ Average Mach number

$\overline{Sm_i}$ Coriolis and centripetal force vector

$\overline{Sm}$ Coriolis and centripetal momentum

$\overline{U}$ Time averaged mean flow property

$\partial\Gamma$ Continuum domain boundary

ρ Density

ρ_f Fluid density

ρ_s Solid density

$\sigma_{\eta\phi\xi}$ Transfinite interpolation partial sum

σ_{ij} Cauchy stress tensor

τ_{ij} Shear stress

τ_{ij}^R Reynolds stress tensor

ε_{ijk} Alternating tensor

$\vec{t}$ Traction vector

A Area

b_i Body force vector

B_{cv} Intensive property value of a control volume

B_{sys} Intensive property value of a control system

C_p Specific heat at constant pressure

C_v Specific heat at constant volume

C_{ijkl} Modulus tensor

CFL Courant-Friedrich-Levy condition

CS Control surface

CV Control volume

D_i Nodal displacement

E Total energy

E Young's modulous

e Specific energy

f_i force vector

f_n Theoretical frequency

F_{il}^{hg} Hourglass force

g gravitational acceleration

H Total enthalpy

h Specific enthalpy

I Area moment of inertia

I Rothalpy

k Thermal conductivity

k Turbulent kinetic energy

K_n Vibration constant

l_{le} Turbulent length scale

M Mach number

$MBFLO3$ In-house fluid-thermal-structure interaction code

n_i Unit normal

P Pressure

Pr Prandtl Number

Pr_t Turbulent Prandtl Number

Q Heat transfer

q_i Heat transfer per unit area

Q_{hg} Hourglass coefficient

R Gas constant for air

R Radius

Re Reynolds number

S Sutherland's Constant

St Strohaul number

$STAGE3$ In-house automated grid generation code

T Temperature

t time

T_s Solid temperature

u' Instantaneous flow property

u_i Velocity vector

V Volume

W Work

w uniform load

w_i Relative velocity

x_i Coordinate vector

Chapter 1

Introduction

In typical design cycles of complex aerodynamic bodies in volatile environments, the aerodynamic, aeroelastic, and aerothermal analysis and design are considered as separate analyses and disciplines. The assumption that these analyses may be treated separately can require additional design iterations to converge on a system that meets all design requirements. Continued advancements in High Performance Computing (HPC) have enabled the development of increasingly complex simulations, enabling the development fluid-thermal-structure interaction (FTSI) simulation techniques presented in this work. With continued development, the methods presented can reduce the number of design iterations needed to meet design requirements.

1.1 Multi-disciplinary Challenges of a Turbomachinery Simulation

The primary function of the turbine in a gas turbine engine is to extract energy from the flow exiting the combustor. In aviation gas turbine engines, the turbine need only extract enough power from the combustion exhaust to power the compressor and bypass fans or propellers and on-board systems while minimizing the turbine weight. In power generation applications, the turbine can be sized to extract as much power from the combustor exhaust possible. In either application, the thermodynamic efficiency of the turbine is to be maximized. The parameter that has the largest impact on the thermal efficiency of the engine is the turbine rotor inlet total temperature (TRIT). For aviation gas turbines, turbine weight is a function of the size and number of airfoils around each turbine blade-row as well as the axial gap size between blade-rows.

From a thermodynamic analysis of a Brayton engine cycle, it becomes clear that the performance of a gas turbine is directly correlated with the temperature at the inlet of the turbine. Modern gas turbines operate at inlet temperatures of 1800-2200 K which are far beyond the allowable metal temperatures. Thus to maintain acceptable fatigue life, reduce creep and maintain safety standards, the structural elements of the blades must be protected against the severe thermal environment [5]. To address these difficulties, technologies such as thermal barrier coatings (TBC) and film cooling holes have been developed. TBC is a thin, often metal plasma coating that is applied to the surface of turbine stator and rotor blades. High pressure turbine airfoils are cooled internally by bleeding cooler air from the high pressure compressor and passing that cooler air into the interior of the turbine airfoils. The internal cooling air helps to keep the interior of the airfoil at an acceptable limit through conduction

and convection. The internal cooling air is then passed to the exterior of the airfoil using holes that are distributed over the airfoil surface. The cooling air exiting the holes lays down a thin film adjacent to the surface that then acts to cool the external surface of the turbine airfoil from convection. An additional benefit of film cooling holes is added wall transpiration, which enhances the stability of the boundary layer if the holes are thoughtfully placed. Some disadvantages of film cooling holes are increased stress concentration in the vicinity of the holes and an increased complexity in manufacturing and design. A schematic of a film cooled turbine blade is shown in figure 1.1.

In addition to the complex thermal and aerodynamic environment, gas turbine engines run at thousands of revolutions per minute and experience unsteady fluid dynamic and aerothermal loads, all of which must be taken into consideration when conducting the structural analysis. To reduce turbine weight, the number of airfoils around the wheel of each blade-row and the axial gap between blade-rows are reduced. This reduction leads to increased loading on the airfoils and increased unsteady interaction between the blade-rows. It is common for the structural analysis of turbine components to consist of a part or system level modal analysis. Then, the components are tuned to avoid resonant frequencies related to the rotational rate of the engine or the aerodynamic loading. The structural analysis and aerothermal analysis are used to predict the High Cycle Fatigue (HCF) life of engine components. The HCF life of a turbine blade determines how frequently these engine components need to be replaced or serviced. Thus, predicting the HCF life of the blades accurately has a major impact on the safety and reliability of the engines.

The turbine aerodynamic team, aerothermal teams, and structures teams must balance competing interests to converge on turbine design that meets system level engine requirements. By allowing each team to conduct their own analysis, more design iterations may be neces-

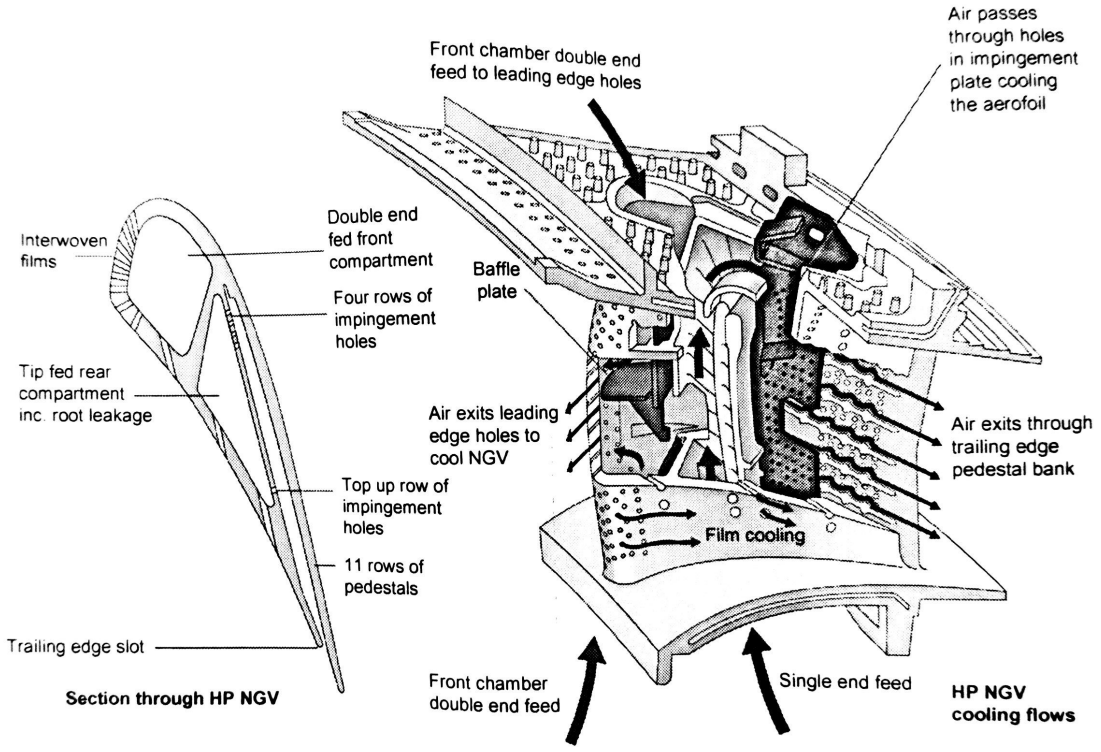


Figure 1.1: Film cooled turbine schematic. Adapted from Dixon and Hall [15]

sary before the design is finalized. Increasing the level of coupling for each of these disciplines would likely improve the design process.

In addition, the need to reduce engine weight (and cost) results in increased turbine airfoil loads and unsteady interaction between airfoil rows that increases airfoil vibration. Aeroelastic analysis is required to minimize the risk of resonance that can lead to fatigue and failure. Fully-coupled, multi-disciplinary aeroelastic analysis is needed to account for complex, non-linear effects at off-design (non-cruise) flight conditions.

1.2 Need for high fidelity turbomachinery simulations

Gas turbine engine designers seek to increase thermodynamic efficiency by raising the inlet temperature of the turbine component flow to levels beyond the material melting point of the turbine airfoils. Three dimensional conjugate heat transfer modeling of the flow and heat transfer in the turbine requires enormous computational processing power and was not considered a viable option a few decades ago. However, in recent years, computational technology has grown exponentially such that the processing power of a typical central processing unit combined with accelerators today far exceeds the capabilities of a few decades ago. Three dimensional models can provide an accurate representation of the flow and thermal fields of real turbine blades while also quantifying taking into consideration the pitch and twist of each turbine airfoil with centripetal and Coriolis forces.

1.3 Past efforts

This section will provide an overview of past efforts related to the development of the monolithic fully coupled fluid-thermal-structure interaction code specialized for turbomachinery presented.

1.3.1 Fluid-structure interaction methods

During the design and analysis cycle of an aerodynamic system, engineers must develop an adequate understanding of the aeroelastic response of the body to properly estimate the fatigue life of the system components. This involves performing analyses over an array of aerodynamic conditions that encompass all likely aerodynamic disturbances to obtain a measure of the structural response, this process can be described as a fluid-structure

interaction (FSI) analysis.

An accurate FSI simulation approach is one which will most closely model the physical aeroelastic response without the use of simplifying assumptions. In generic FSI problems, any change in the fluid domain will effect the response of the structural domain and vice versa. There are two key decisions an analyst must make when conducting an FSI simulation. First, the extent of the coupling between the fluid and solid continua. This will determine how frequently the fluid domain and the structural domain will exchange boundary condition information. Second, the selection of which numerical methods will be used to solve the governing equations of the fluid and solid continua (e.g. finite volume method). This decision will dictate if FSI analysis will be monolithic or partitioned. Monolithic solvers use the same numerical method to solve the governing equations of fluid and solid continua.

Partitioned approaches may range from fully coupled to loosely coupled. In these approaches, the fluid and solid domain governing equations are solved using different numerical schemes. Many partitioned approaches model the fluid domain using a finite volume solver, and simulate the solid response by solving the modal equations using finite elements. Examples of fully coupled partitioned methods that use a finite volume fluid solver and simulate the structural response by solving the modal dynamic equations using finite elements include: Im et. al. [31], Patel and Zha [55], Zhang et. al., [70], Doi and Alonso [16], Johnston et. al. [38], Chen et. al. [6], and Sadeghi and Liu [58]. Examples of loosely coupled or uncoupled FSI approaches include Dettmer and Peric [14], Moffat and He [49], McMullen et. al. [47], and Liu et. al. [43]. Some important advantages of partitioned approaches are the relative speed of computation, and the ease of implementation. Solving the modal dynamic equations to model the structural response allows for the structural mesh to be coarser than the fluid mesh, and for a larger solid time-step thereby reducing the numerical stiffness and computa-

tional time of the FSI method. However, high frequency behavior, modal coupling, and other non-linear behaviour will go unnoticed by the structural solver. Without a point-matched grid, a method of interpolating and communicating fluid forces to the solid nodes must be implemented. Special consideration must be given to the interpolation method to ensure the traction load is conserved across the fluid-solid boundary. Additionally, differences in numerical schemes, particularly in time-steping, can lead to an acceleration miss-match at the fluid-solid interface even if the velocity and displacements are matched [13].

Monolithic FSI methods are necessarily fully coupled and use point-matched grids. The fluid and the structural continuum governing equations are solved using the same numerical methods and are marched in time together. Examples of monolithic approaches include Zorn and Davis [72], who used the finite volume method, Turek and Hron [63], Ramkrishnan et. al. [56], Gottfried and Fleeter [22], Han et. al. [27], and Teixeira and Awurch [61], who used the finite element method. One additional complexity of a monolithic approach is the governing equations for the fluid and solid domains are derived using Eularian and Lagrangian reference frames respectively. As a result, an Arbitrary Lagrangian Eulerian (ALE) method must be implemented to solve the governing equations of the two continua simultaneously. Another additional complexity of the monolithic approach is the maximum stable time step in the solid domain is typically much smaller than the maximum allowable fluid time step. The use of point-matched grids may make the solid domain arbitrarily more refined to meet the mesh sizing requirements of the fluid solver. This will increase the numerical stiffness and computational time of the FSI method for explicit or semi-implicit time-stepping methods. This challenge is more difficult to overcome and generally causes monolithic FSI approaches to be numerically stiffer more costly than partitioned approaches [13] [30]. Due to the algorithmic complexity and computational requirements of monolithic

solvers, many are two-dimensional.

1.3.2 Conjugate heat transfer and modeling of film cooling holes

Various turbomachinery conjugate heat transfer solution procedures have been developed over the last decade. A review of the various techniques commonly used in conjugate heat transfer analyses from highest fidelity to lowest can be found in He and Fadl [28]. The focus of current research is on high-fidelity conjugate heat transfer methods that are applicable to realistic, film-cooled turbine configurations for steady or unsteady predictions. Tightly-coupled fluid and solid solvers that use the same discretization scheme and where surface heat-fluxes from both are used to determine a unique and shared surface temperature at each time-step have become more popular recently. These procedures can be time-accurate but are computationally expensive due to the large differences in time-scales between the fluid (short) and solid (long) domains. The current procedure falls into this category and is demonstrated for "steady" conditions in which local time-steps are used.

Recent high-fidelity, conjugate heat transfer approaches for realistic, film-cooled turbine airfoils fall into one of several computational meshing categories; point-matched unstructured, point-matched structured, and Chimera/ embedded/overset (overlaid). Many of the computational investigations of heat transfer of realistic film-cooled turbines have been used to determine the computational grid requirements for grid-independence as well as to validate the conjugate solvers against experimental data.

Unstructured-grid approaches, in which combinations of point-matched tetrahedral, quadrilateral, and hexahedral grid cells are utilized to discretize the fluid and solid domains are often used due to their ability to mesh the complex interfaces between cooling-holes and

mainstream flow. An advantage of this approach is its ability to mesh complex, multi-featured geometry but often with labor and time intensive costs. In addition, inaccuracies in high-gradient regions, such as boundary layers, can occur when stretched prismatic or tetrahedral cells are used. One recent example of this type of approach was demonstrated by Wang, et. al [66] who used ANSYS ICEM CFD to generate a point-matched, unstructured grid with approximately 51 million grid points over the fluid and solid domains. The ANSYS CFX code was used to solve the governing equations including the Menter [48] SST k-ω turbulence model with the Langtry and Menter [41] $\gamma - \theta$ transition model. The focus of the investigation was to demonstrate the optimization of the film-cooling hole distribution to increase cooling effectiveness and reduce airfoil surface temperature. Kim, et. al [39] used a point-matched unstructured grid consisting of 31.5 million elements in conjunction with the ANSYS CFX multi-disciplinary solver using the Wilcox[68] k-ω turbulence model. The Hierarchical Kriging [26] method was used in conjunction with the conjugate heat transfer computations to optimize the film-cooling hole distribution. Liu, et. al [45] used a point-matched unstructured grid of 7.4 million tetrahedral elements in conjunction with the ANSYS CFX solver using the Menter[48] SST k-ω turbulence model to investigate the heat transfer of a turbine vane. Liu, et. al [44] used the STAR-CCM+ package to generate a grid of 11.7 million polyhedral cells to investigate the steady heat transfer in a turbine stator-rotor stage. Griffini, et. al [25] used the Centaur package to generate a point-matched unstructured grid of 14.2 million prismatic and tetrahedral elements with the ANSYS Fluent solver and the Walters and Cokljat [64] $k_T - k_L - \omega$ turbulence/ transition model. The change in the airfoil surface heat transfer and temperature distribution due to the clocking of combustor inlet profiles was investigated. Johnson et. al [35],[36], [37] and Ni, et. al [52], [51] used the ANSYS ICEM CFD grid package to generate a point-matched unstructured

grid of 24.2 million cells in conjunction with the ADS CFD solver and the Wilcox [68] k-ω turbulence model. Their investigations focused on heat transfer prediction with film-cooling and optimization of film-cooling hole distribution.

Recent examples of point-matched multi-block structured grid approaches to conjugate heat transfer prediction of realistic, film-cooled turbines have also been performed. The primary advantage to this approach is the use of body-fitted grid blocks that usually provide superior accuracy, especially in high gradient, viscous flow regions. The disadvantage of this approach is the difficulty of point-matching grid blocks at the intersection of differing topological features and the grid-stripping that often results. Wang and Yan [65] [66] used a point-matched multi-block structured grid using 46 million points generated with their automated Background-Based Mapping code for the flow regions and a point-matched unstructured grid for the solid regions. The ANSYS CFX multi-disciplinary solver was used to predict the flow and heat transfer with the Menter and Langtry[41] [48] turbulence/transition model. Dong and Amano [17] compared point-matched, multi-block structured grid predictions of conjugate heat transfer with point-matched unstructured grid predictions. Both computational grids were generated with the ICEM CFD package. Their multi-block structured grid consisted of 15.1 million grid points whereas the unstructured mesh used 19.3 million grid elements. The ANSYS CFX code was used with the Menter [48] SST k-w turbulence model to predict the Reynolds-Averaged Navier-Stokes (RANS) temperature and heat transfer on the film-cooled turbine vane. Large-eddy and Detached-eddy simulations were also investigated. Mangani et. al [46] used the GridPro package to generate a point-matched multi-block structured grid of 10,987 blocks with 6.7 million hexahedral cells. The OpenFoam multi-disciplinary solver was used to investigate the heat transfer and temperature distribution of a film-cooled vane using a variety of turbulence models. Goormans-Francke

et. al [21] used their own point-matched, multi-block, structured grid procedure to generate a mesh for a realistic film-cooled turbine vane/blade stage including 1000 cooling holes, 150 pin fins, 250 turbulators, impingement jets, and recessed blade tip with 154 million grid points and 12,316 blocks. The FINETM/Turbo code with the k-ϵ turbulence model with wall functions was used to predict the heat transfer on a coarse-grid model of 21.8 million points and 12,000 blocks in an initial simulation.

The third approach to modeling film-cooling flows in realistic turbine configurations consists of Chimera, overset/overlaid, and/or embedded multi-block structured grids. The advantage of this approach is the ability to create complex geometrical models automatically and/or fairly quickly that can be easily used in optimization processes. A disadvantage of this approach is the non-conservative interpolation between overset grids that can cause error. Castillon et. al [4] used a Chimera, overset grid package they developed that interfaces directly with their elsA multidisciplinary solver. The mesh to predict the conjugate heat transfer for a film-cooled turbine vane consisted of 38.7 million grid points. Lad and He [40] used an Immersed Mesh Block overset grid method to investigate the cooling flows for a film-cooled nozzle guide vane in which each row of cooling hole was independently meshed with its own airfoil internal plenum and associated boundary conditions, cooling hole, and mainstream interface grid blocks. The internal plenum(s) of the guide vane were not meshed and conjugate heat transfer was not investigated. However, this work does demonstrate the use of embedded/overset grids and the highlights the related numerical issues.

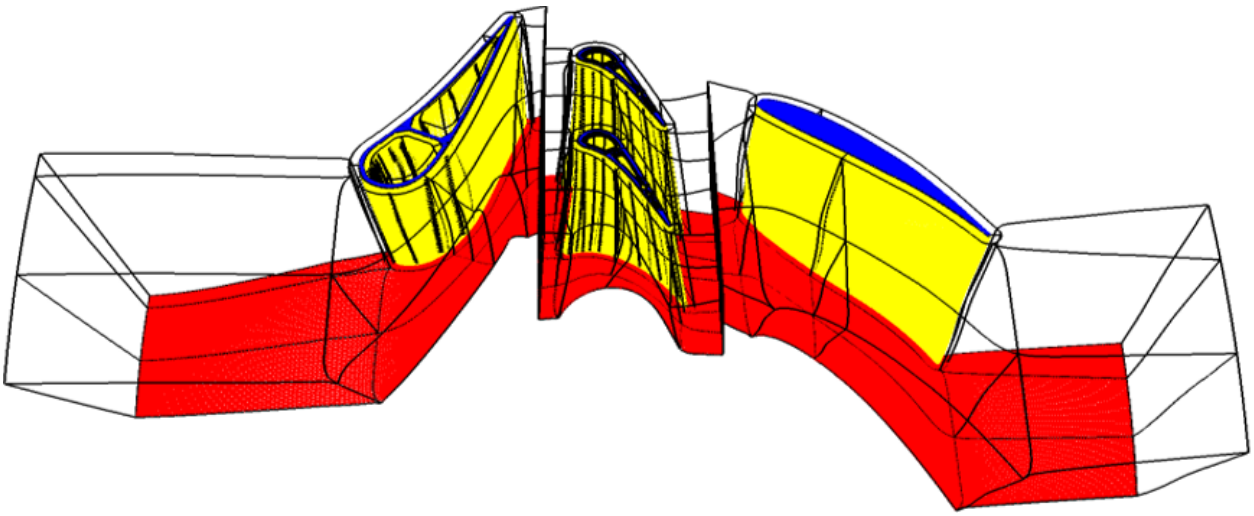


Figure 1.2: $1\frac{1}{2}$ stage high pressure turbine computational grid example.

1.4 Dissertation research objectives

The primary goal of this research is to continue development of in-house high fidelity CFD code MBFLO3 [2],[3], [9], [8],[1],[42] to create a monolithic fluid-thermal-structural interaction method capable of simulating extremely complex turbomachine geometry. At previous stages of development, improvements to MBFLO3 have been validated with a $1\frac{1}{2}$ stage film cooled turbine with thermal barrier coating [42]. The goal of this research is to add the capabilities needed to consider a fluid-thermal-structure simulation of the $1\frac{1}{2}$ stage turbine with film cooling holes and thermal barrier coatings, and other complex geometries. An example computational grid of the $1\frac{1}{2}$ stage turbine is shown in figure 1.2. This example computational grid has over 3000 computational blocks and over 150 million grid points.

To achieve this goal, four major code development efforts must be undertaken. First, the development of a method for modeling film cooling holes as overlaid/over-set grids. Second, improve generality and accuracy of the conjugate heat transfer solver to support complex turbomachine geometry. Third, the creation of a finite volume based structural solver that

uses the same numerical methods to solve the solid continuum governing equations as the fluid solver. Fourth, the development of a fully coupled fluid-structure interaction method. Due to the turbomachine geometries to be analysed, each of these code development efforts must be efficient and general.

The following chapters will discuss the governing equations for the fluid and solid continua, and the numerical methods and techniques implemented to solve them. Then, the simulation results which verify or validate each of the four stated research goals will be presented. Finally, the current development state of MBFLO3 will be discussed and the next steps to complete the goal of simulating a fluid-thermal-structure interaction on a $1\frac{1}{2}$ stage high pressure turbine with film cooling holes and thermal barrier coatings will be addressed.

Chapter 2

Governing Equations

The equations of conservation of mass, momentum, and energy for fluid and solid domains are derived with the assumption that the domains may be treated as a continuum. A continuum can be established if the variation in properties within a volume is so smooth that differential calculus can be applied to quantify the changes of properties within the domain. A continuum body can be continually sub-divided into infinitesimal elements with each element having the same properties as the bulk of the material.

2.1 Fluid Continuum

The conservation of mass, momentum, and energy integral equations are derived in this section. Physical changes to conserved variables are tracked through the fluid continuum through the use of control volumes and Reynolds transport theorem. Through the use of Reynolds transport theorem the changes in extensive and intensive properties. An extensive propriety is dependent on the mass of a given system, while an intensive propriety is

independent of the system mass.

2.1.1 Reynolds Transport Theory

Before the derivation of the fluid continuum conservation equations, it is useful to consider a tool capable of tracking changes in conserved quantities as a system of fluid passes through a control volume. Consider the control volume and system shown in figure 2.1. In this system, an arbitrary control volume is subject to an arbitrary flow field. Let B be any property of the fluid, and define the intensive value of property B as $\beta = dB/dm$. At time t_0 the control volume boundary and system boundary are coincident.

$$B_{sys}(t) = B_{cv}(t) = \int_{CV} \beta dm = \int_{CV} \beta \rho dV \tag{2.1}$$

At time $t_i = t_0 + \Delta t$ the fluid system has moved to occupy region II and III, while the control volume remains fixed in space. Changes in the fluid property can be tracked by $B_{sys} = B_{cv}(t + \Delta t) + B_{II}(t + \Delta t) - B_I(t + \Delta t)$. Then, changes to the extensive property as the limit $\Delta t \to 0$, resulting in equation 2.2.

$$\frac{\partial B_{CV}}{\partial t} = \frac{\partial}{\partial t} \int_{CV} \beta \rho dV \tag{2.2}$$

Next, consider the intensive property flux through the control surfaces of the control volume.

$$\dot{B} = \int_{CS} \beta \rho V_i n_i dA \tag{2.3}$$

Finally, the total time rate of change of the fluid is given by the Reynolds Transport Theorem

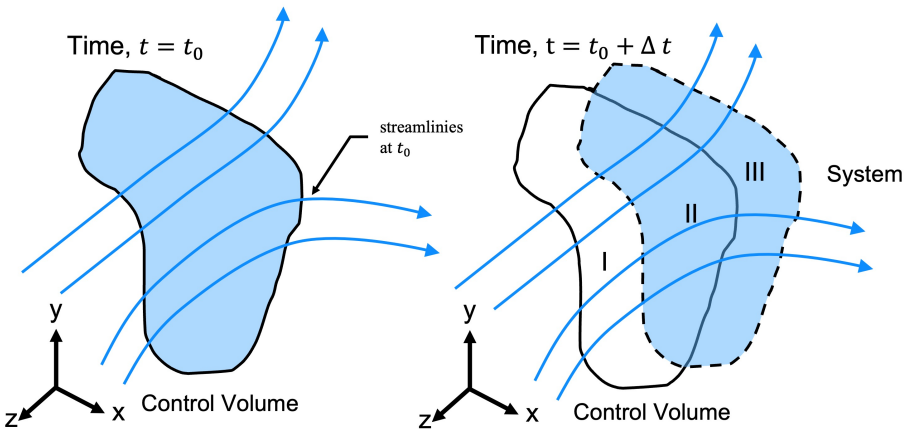


Figure 2.1: Control volume and system changes with time.

in equation 2.4.

$$\frac{DB_{sys}}{Dt} = \frac{\partial}{\partial t}\int_{CV} \beta\rho dV + \int_{CS} \beta\rho u_i dA \tag{2.4}$$

The objective of Reynolds Transport Theorem is to relate the rate of change of an extensive fluid property to the time variations of the fluid property within a given control volume.

2.1.2 Conservation of Mass

The Reynolds transport theorem given in equation 2.4 defines a relationship between the rate of change of an extensive property in a system and control volume and control surface integrals. To evaluate the rate of change of mass in a system, the dummy variable B will be replaced with system mass, m.

$$\left(\frac{dm}{dt}\right)_{syst} = \frac{\partial}{\partial t}\int_{CV} \rho dV + \int_{CS} \rho u_i dA \tag{2.5}$$

The physical axiom that mass cannot be created or destroyed sets any time rates of change

of system mass to be 0. The result is the integral form of the conservation of mass equation 2.6.

$$0 = \frac{\partial}{\partial t}\int_{CV} \rho dV + \int_{CS} \rho u_i dA \tag{2.6}$$

The conservation of mass equation may also be considered in differential form by considering an infinitesimal control volume with dimensions $dx \times dy \times dz$. The result is equation 2.7

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \tag{2.7}$$

2.1.3 Conservation of Momentum

The conservation of linear momentum equation is Newton's second law, $F_i = ma_i$, applied to a fluid in a control volume. The forces acting on a control volume are due to pressure, gravitational loads, and shear stress. The integral form of the conservation of momentum equations can be obtained by defining the dummy variable in equation 2.4 as $B_i = mu_i$ and $\beta = dB_i/dm = u_i$. In the current work, the densities of the fluids considered are small, therefore gravitational loads are neglected, but it is left in equation 2.8 for the sake of completeness.

$$\sum F_i = \int_{CS} \rho \frac{\partial \tau_{ij}}{\partial x_j} dA - \int_{CS} P dA + \int_{CV} \rho g_i dV = \frac{\partial}{\partial t}\Big(\int_{CV} \rho u_i dV\Big) + \int_{CS} \rho u_j u_i dA \tag{2.8}$$

To obtain the differential form of the conservation of momentum equation, consider Newton's Second Law but replace the acceleration vector with the total acceleration of a given particle,

given in equation 2.9.

$$a_i = \frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_i} \tag{2.9}$$

The forces acting on a differential control volume fluid particle remain the same, pressure, gravitational and shear loads. The resulting equation is the Navier-Stokes equation 2.10. The viscous stress tensor τ_{ij} is discussed at length in section 2.1.5.

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.10}$$

2.1.4 Conservation of Energy

The conservation of energy equation is the first law of thermodynamics, shown in equation 2.11, applied to a fluid control volume. Where $\frac{D}{Dt}$ is the material derivative, E is the total system energy, $\dot{Q}$ is the rate of heat transfer into the system, and $\dot{W}$ is the work being done by the system.

$$\frac{DE}{Dt} = \dot{Q} - \dot{W} \tag{2.11}$$

Reynolds transport theorem may again be applied here to obtain the integral form of the energy equation. In this case, the dummy variable B may be replaced with energy E, and $\beta = \frac{dE}{dm} = e$. The left hand side of equation 2.11 The total energy change within a control volume an be captured by integrating the time rate of change of energy within a control volume plus the energy flux through control surfaces.

$$\frac{DE}{Dt} = \frac{\partial}{\partial t}\Big(\int_{CV} \rho E dV\Big) + \int_{CS} \rho u_j u_i dA = \int_{CS} \rho u_i \tau_{ij} dA - \int_{CS} \rho q_j dA \tag{2.12}$$

The differential form of the conservation of energy equation can be obtained by considering

an infinitesimal volume. Let $\dot{Q}$ be defined by Fourier's Law in equation 2.13, where q_j is the vector of heat flow per unit area, k is the coefficient of thermal conductivity.

$$q_j = -k\frac{\partial T}{\partial x_j} \tag{2.13}$$

The thermal conductivity of a fluid is defined in equation 2.14.

$$k = \frac{\mu C_p}{Pr} \tag{2.14}$$

Combining equations 2.13 and 2.14, the heat transfer rate into the volume is described in equation 2.15.

$$\dot{Q} = -\frac{\partial q_j}{\partial x_j} = \frac{\partial}{\partial x_j}\Big(\frac{\mu}{Pr}\frac{\partial h}{\partial x_j}\Big) \tag{2.15}$$

The rate of work done on the fluid volume is dependent on the velocity, shear stresses and pressure, per equation 2.16.

$$\dot{W} = -\frac{\partial w_j}{\partial x_j} = -\frac{u_i(\tau_{ij} - p)}{\partial x_j} \tag{2.16}$$

Combining equations 2.11, 2.15, and 2.16, expanding the material derivative on the left hand side of 2.11, and including the definition of enthalpy from equation 2.17, the differential form of the energy equation is shown in equation 2.18.

$$H = \frac{E + P}{\rho} \tag{2.17}$$

$$\frac{\partial E}{\partial t} + \frac{\partial (u_j E)}{\partial x_j} = \frac{\partial}{\partial x_j}\Big[(u_i(\tau_{ij} - p) + \Big(\frac{\mu}{Pr}\frac{\partial h}{\partial x_j}\Big)\Big] \tag{2.18}$$

2.1.5 Viscous Stress Tensors

The viscous stress tensor τ_{ij} is included in the differential and integral forms of the momentum and energy equations, 2.8, 2.10, 2.12, 2.18, but has not been discussed in detail yet. All fluids considered in this analysis are Newtonian, therefore stress in the fluid is dependent only on the viscosity and velocity gradient. The general shear stress of a Newtonian fluid is shown in equation 2.19, where λ is the result of Stoke's hypothesis, $\lambda = -\frac{2}{3}\mu$ [67].

$$\tau_{ij} = \mu\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) + \lambda\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{2.19}$$

The viscosity of a fluid is dependent on the temperature per Sutherland's Law, equation 2.20. Where $\mu_{ref} = 1.716 \times 10^{-5}\frac{kg}{ms}$ and $T_{ref} = 273.15K$ are gas dependent user inputs, and $S = 110.4K$

$$\mu = \mu_{ref}\Big(\frac{T}{T_{ref}}\Big)^{3/2}\frac{T_{ref} + S}{T + S} \tag{2.20}$$

2.1.6 The Navier Stokes Equations

Substituting terms from equations 2.19 into 2.10, the Navier Stokes equations are formed in equation 2.21. The Navier-Stokes equations are non-linear partial differential equations

with no general solution.

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\Big[\mu\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{ij}\Big] \tag{2.21}$$

The Navier Stokes 2.21 equations, the conservation of mass equation 2.7, and the conservation of energy equation 2.18 can be written in conservative form, as shown in equations 2.22 through 2.25, where ρ is the fluid density, u, v, and w are the x, y, and z velocity components. H_o is the total enthalpy, E_o is the total energy, and p is the static pressure. The F_v, G_v, and H_v arrays contain values of shear stresses that can be found by expanding equations 2.21 and 2.18 with shear stress tensor definition from equation 2.19. The conservative formulation can be interpreted as the time rate of change of conserved quantities U is equal to the inviscid flux in primary variables for a coordinate direction, F_i, G_i, H_i minus the viscous flux F_v, G_v, H_v. The Conservative formulation of the governing equations is convenient to demonstrate the relationship between conserved quantities.

$$\frac{\partial U}{\partial t} = -\frac{\partial (F_i + F_v)}{\partial x} - \frac{\partial (G_i + G_v)}{\partial y} - \frac{\partial (H_i + H_v)}{\partial z} \tag{2.22}$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_o \end{bmatrix} F_i = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u H_o \end{bmatrix} F_v = - \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau xz \\ \tau_{hx} \end{bmatrix} \tag{2.23}$$

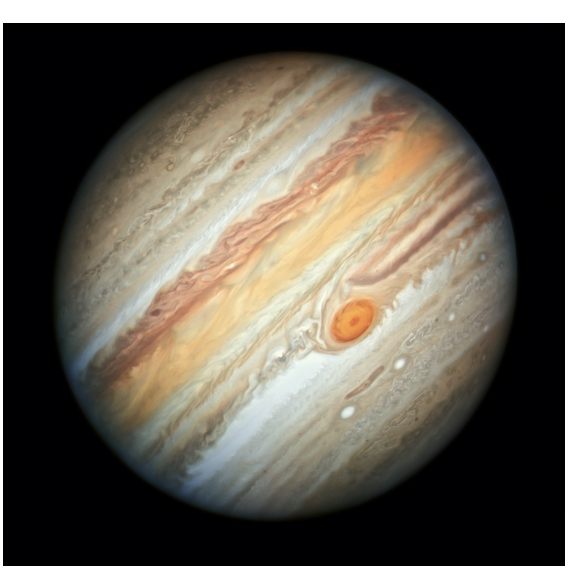
$$G_i = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho v H_o \end{bmatrix} G_v = - \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau yz \\ \tau_{hy} \end{bmatrix} \tag{2.24}$$

$$H_i = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho w H_o \end{bmatrix} H_v = - \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau zz \\ \tau_{hz} \end{bmatrix} \tag{2.25}$$

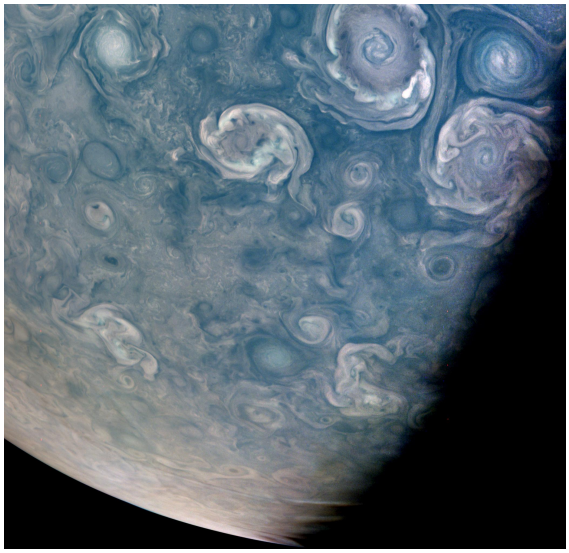
2.1.7 Turbulence

A beautiful demonstration of turbulence can be seen in the clouds of Jupiter, figure 2.2. The clouds show a spatially varying flow with superimposed three-dimensional random circulations and fluctuations that are self sustaining and enhance mixing, diffusion, entrainment and dissipation [67]. Large and small vortical structures can be also observed. This is the so called "turbulence energy cascade" which asserts that turbulence exists at various length scale and large vortical structures contain more fluid energy and momentum than smaller vorticies.

White [67] lists five characteristics of turbulence:



(a) Hubble Space Telescope Portrait of Jupiter



(b) Jupiter's north pole from NASA's Juno Mission

Figure 2.2: Turbulence examples in Jupiter's clouds.

1. *Fluctuations* in pressure, velocity, and temperature (when heat transfer is present). Velocity fluctuates in all three coordinate directions. Fluctuations of each property are superimposed on the mean value of each property.

2. *Eddies* or fluid packets of many sizes that intermingle and fill the shear layer.

3. *Random* variations in fluid properties which have a particular form. Each property has a specific continuous energy spectrum that drops off to zero in the smallest eddies.

4. *Self-sustaining* motion. Once triggered, turbulent flow can maintain itself by producing new eddies to replace those lost by viscous dissipation.

5. *Mixing* that is much stronger than that due to laminar (molecular) action. Turbulent eddies actively move about in three dimensions and cause rapid diffusion of mass, momentum, and energy.

The first characteristic of turbulence is the foundation for turbulence modeling. The assertion that fluctuations of pressure, velocity, and temperature can be super-imposed on the mean values of fluid flow allow the mean value and the fluctuations to be decoupled from one-another, as demonstrated in figure 2.3.

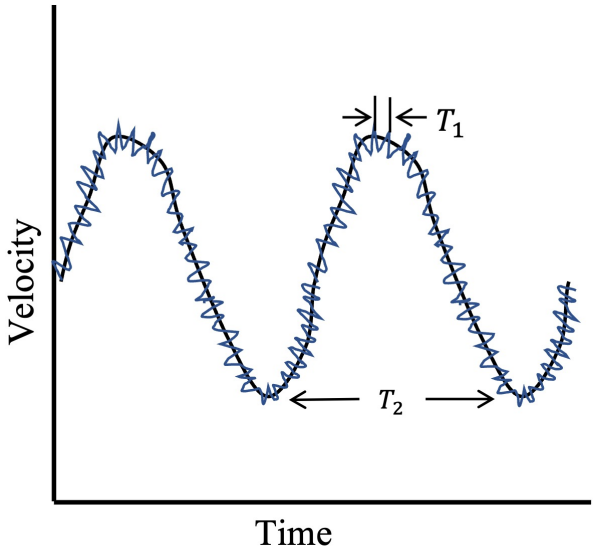


Figure 2.3: Example of turbulent flow variation form mean flow.

The perturbations of properties in the fluid flow can be separated from the mean flow as shown in equation 2.26. Where $\overline{U}$ is the time averaged mean flow property, and $u'(x_i, t)$ are the fluid fluctuations.

$$u(x_i, t) = \overline{U}(x_i) + u'(x_i, t) \tag{2.26}$$

where,

$$\overline{U}(x_i, t) = \frac{1}{T}\int_t^T u(x_i, t)dt \quad for \quad T_1 << T << T_2 \tag{2.27}$$

It is important to understand what turbulent length scales can be safely ignored when decoupling the mean flow properties from the perturbations. One could imagine choosing a

value of T_2 that is so large the mean flow path in figure 2.3 becomes the perturbation and the mean flow path becomes a flat line.

Reynolds Averaged Navier Stokes (RANS) Equations

The Navier-Stokes equations, equation 2.21, are non-linear partial differential equations for which only a handful of analytical solutions exist. As such, it is important to consider some simplifications to these equations to make it possible to obtain meaningful results. One simplification is to decouple the mean flow parameters from the fluctuations, as done in equation 2.26. For incompressible flows, equation 2.26 may be substituted into the differential forms of the conservation of mass equation 2.7, and conservation of momentum equation 2.21. Before that process is discussed, correlations for the multiplication of two time averaged quantities must be discussed.

Consider the multiplication of multiple time averaged quantities $\overline{AB}$, that can be separated into fluctuating and mean properties, per equation 2.26. The result is shown in equation 2.28. The product of a mean flow parameter and a perturbation is zero, however the product of two perturbations is non-zero if A and B are correlated. Two properties are correlated if a change in A could cause a change in B and vice versa.

$$\overline{AB} = \overline{(A + A')(B + B')} = \overline{AB + AB' + A'B + A'B'} = AB + \overline{A'B'} \tag{2.28}$$

A similar process may be conducted for a triple product, shown in equation 2.29.

$$\overline{ABC} = ABC + \overline{B'C'}A + \overline{A'C'}B + \overline{A'B'}C + \overline{A'B'C'} \tag{2.29}$$

Equation 2.26 may now be substituted into the incompressible form of the conservation of mass and momentum equations, resulting in the Reynolds Averaged Navier Stokes (RANS) equations, 2.30, 2.31, and 2.32.

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{2.30}$$

$$\rho\frac{\partial U_i}{\partial t} + \rho U_j\frac{\partial U_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\Big(\tau_{ji} - \tau_{ji}^R\Big) \tag{2.31}$$

where, τ_{ji}^R is the Reynolds Stress tensor.

$$\tau_{ji}^R = -\overline{\rho u_j' u_i'} \tag{2.32}$$

The Reynolds stress tensor is symmetric. Therefore it has been traditionally thought that an additional six unknowns have been introduced through the averaging process [18] [67] [68]. However, recent mathematical analyses challenge this assertion. Sun [60] has concluded that only three independent unknowns, u_i', are introduced due to the Reynolds stress term. Whether the number of unknowns introduced is three or, new equations need to be developed to close the RANS equations. The closure of the RANS equations has been a topic of intense research for decades [41] [48] [68] [67]. One method of closure is discussed in 2.1.7.

Farve Averaging

When considering the compressible form of the Navier-Stokes equations, the fluid temperature and density may also fluctuate with time. As a result, a slightly more sophisticated version of Reynolds averaging must be conducted. The method of density-weighted averaging proposed by Farve [18] is shown in equation 2.33. With this formulation, the momentum

per unit volume, ρu_i, becomes the dependent variable, rather than the velocity.

$$\overline{u}_i = \frac{1}{\overline{\rho}} \lim_{T \leftarrow \infty} \frac{1}{T} \int_{t_0}^{t+\infty} \rho(x_i, \tau) u(x_i, \tau) d\tau \tag{2.33}$$

Applying Reynolds averaging to equation 2.33 results in equation 2.34.

$$\overline{\rho}\tilde{u}_i = \overline{\rho} U_i + \overline{\rho' u_i'} \tag{2.34}$$

Taking equation 2.34 and substituting it into the conservation of mass, equation 2.7, conservation of momentum, equation 2.21, and conservation of energy 2.18, results in the Farve averaged conservation of mass, equation 2.35, conservation of momentum, equation 2.36, and conservation of energy, equation 2.37. Full details of the Farve averaging process can be found in [18] and [68].

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\overline{\rho}\tilde{u}_i) = 0 \tag{2.35}$$

$$\frac{\partial(\overline{\rho}\tilde{u}_i)}{\partial t} + \frac{\partial(\overline{\rho}\tilde{u}_j\tilde{u}_i)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\Big(\overline{\tau_{ji}} + \tau_{ji}^R\Big) \tag{2.36}$$

$$\frac{\partial \overline{\rho} E}{\partial t} + \frac{\partial(\overline{\rho}\tilde{u}_j H)}{\partial x_j} = \frac{\partial}{\partial x_j}\Big[\tilde{u}_i \hat{\tau_{ji}} + \Big(\frac{\mu}{Pr} + \frac{\mu_T}{\mathrm{Pr}_T}\Big)\frac{\partial \tilde{h}}{\partial x_j}\Big] \tag{2.37}$$

where Pr and Pr_T are the laminar and turbulent Prandtl numbers, $\frac{c_p \mu_T}{k_f}$, k_f is the thermal conductivity of the fluid, and $\hat{\tau_{ji}}$ is defined in equation 2.38.

$$\hat{\tau_{ji}} = \tau_{ji} + \tau_{ji}^R = (\mu + \mu_T)\Big[\frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big) - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ji}\Big] \tag{2.38}$$

As seen with the incompressible RANS equations in section 2.1.7, preforming mass averaging

introduces additional unknowns related to the Reynolds stress tensor into the system of fluid governing equations. Additionally, some method of correlating the mean total energy E with the temperature must be established. If the simulated fluid is considered an ideal gas, the equation of state can be used to correlate E with T, shown in equation 2.39.

$$P = \overline{\rho} R \tilde{T} \tag{2.39}$$

Wilcox's Two Equation k-ω model

One method of closing the mass averaged RANS equations is Wilcox's [68] k-ω turbulence model. k is the turbulent kinetic energy and ω is the specific dissipation rate. This turbulence model is well posed to capture highly vortical flows and the k-ω model can be applied to a boundary layer without modification. This, along with high grid density near a no slip boundary, allows for an accurate simulation of flow physics in the boundary layer without the use of wall functions. Equations 2.40 and 2.41 are the two equations of the k-ω turbulence model, with constants $\alpha = 0.52$, $\beta = 0.075$, $\beta* = 0.09$, $\sigma = \sigma* = 0.5$, and $\sigma_d = 0.125$. The turbulent viscosity, μ_T is defined in equation 2.43.

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho w_j k)}{\partial x_j} = \Big(\tau_{ij} - \frac{2}{3}\rho k \delta_{ij}\Big)\frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j}\Big(\Big(\mu + \frac{\rho k}{\omega}\Big)\frac{\partial k}{\partial x_j}\Big) \tag{2.40}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho w_j \omega)}{\partial x_j} = \alpha\frac{\omega}{k}\Big(\tau_{ij} - \frac{2}{3}\rho k \delta_{ij}\Big)\frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega^2 + \sigma_d \frac{\rho}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j}\Big(\Big(\mu + \frac{\rho k}{\omega}\Big)\frac{\partial \omega}{\partial x_j}\Big) \tag{2.41}$$

$$l_{le} = \frac{\sqrt{k}}{\omega} \tag{2.42}$$

$$\mu_t = \rho l_{le} \sqrt{k} \tag{2.43}$$

2.1.8 Coriolis and Centripetal forces

The developed computational fluid dynamics code is intended to be used for turbomachinery. The conservation equations are solved in the airfoil-relative frame of reference frame. In rotor blade passages of turbomachinery the grid is rotating per MBFLO3 sliding grid algorithm developed by Lee [42]. The rotation of the blades will impart additional coriolis and centripetal forces that must be considered in when solving the fluid conservation of momentum equations. The coriolis and centripetal effects of a rotating grid can be calculated using 2.44, where $\overline{\Omega}$ is the rotational rate, $\overline{R}$ is fluid cell radius, ε_{ijk} is the alternating tensor, $\hat{e}_k$ and $\hat{e}_j$ are unit vectors in the k and j direction respectively.

$$\overline{Sm_i} = 2\rho\varepsilon_{ijk}\overline{\Omega_i}u_j\hat{e}_k + \rho\overline{\Omega_i}\,\overline{\Omega_j}\,\overline{R_i}\hat{e}_j - \rho\overline{\Omega_i}^2\,\overline{R_k}\hat{e}_k \tag{2.44}$$

Equation 2.44 is added to the right hand side of the conservation of momentum equations 2.30.

2.1.9 Effect of a Moving Control Volume

The fluid domain governing equations are derived in a stationary Eulerian reference frame. If a fluid control volume or control volume boundary move with some non-zero velocity v_i, the governing equations will need to be slightly modified to facilitate this movement. A relative velocity term w_i is defined in equation 2.45. Note that the fluid velocity u_i can be the Reynolds averaged velocity, the control volume velocity, v_i, could be caused by some deformation such as aeroelastic movement of the fluid boundary. The simple equation 2.45 is the basis for connecting the fluid domain Eulerian reference frame with the solid domain Lagrangian reference frame, and the Arbitrary Eulerian-Lagrangian (ALE) method used in

the fluid-structure interaction procedure presented in this work.

$$w_i = u_i - v_i \tag{2.45}$$

2.1.10 Rotating Reference Frames

When conducting a turbomachinery analysis, it is advantageous to consider two frames of reference, an absolute non-rotating frame, and a relative frame that rotates with the blade. As previously mentioned, the relative frame is used to quantify the flow properties the rotor blade is experiencing. This formulation gives rise to several new terms such as relative total pressure, total temperature, total enthalpy and relative Mach number. All of these quantities differ from the absolute total pressure, total temperature, total enthalpy and absolute Mach number by the subtraction of terms related to the rotational speed of the rotor blades. The relative Mach number is calculated by taking the relative velocity and dividing it by the speed of sound. The relative total enthalpy is often referred to as rothalpy, which is a contraction of its formal name, is defined in equation 2.46. Rothalpy is constant throughout adiabatic rotating machinery at a fixed radius, which is analogous to total enthalpy being constant across non-rotating machinery [15].

$$I = H - \frac{1}{2}(\Omega R)^2 = \frac{E + P}{\rho} \tag{2.46}$$

2.1.11 The Implemented Fluid Governing Equations

The Unsteady Reynolds Averaged Navier Stokes Equations (URANS) solved using numerical methods discussed in chapter 3 are as follows.

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho w_i)}{\partial x_i} = 0 \tag{2.47}$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial \rho(w_j u_i)}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} - \overline{Sm_i} \tag{2.48}$$

$$\frac{\partial E}{\partial t} + \frac{\partial(\rho w_j I)}{\partial x_j} = \frac{\partial}{\partial x_j}\Big(u_i \tau_{ij} + \Big(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_T}\Big)\frac{\partial h}{\partial x_j} - \frac{\partial(p v_j)}{\partial x_j}\Big) \tag{2.49}$$

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho w_j k)}{\partial x_j} = \Big(\tau_{ij} - \frac{2}{3}\rho k \delta_{ij}\Big)\frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j}\Big(\Big(\mu + \frac{\rho k}{\omega}\Big)\frac{\partial k}{\partial x_j}\Big) \tag{2.50}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho w_j \omega)}{\partial x_j} = \alpha \frac{\omega}{k}\Big(\tau_{ij} - \frac{2}{3}\rho k \delta_{ij}\Big)\frac{\partial u_i}{\partial x_j} - \beta^* \rho k \omega^2 + \sigma_d \frac{\rho}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j}\Big(\Big(\mu + \frac{\rho k}{\omega}\Big)\frac{\partial \omega}{\partial x_j}\Big) \tag{2.51}$$

$$l_{le} = \frac{\sqrt{k}}{\omega} \tag{2.52}$$

$$\mu_T = \rho l_{le} \sqrt{k} \tag{2.53}$$

$$\overline{Sm_i} = 2\rho \varepsilon_{ijk} \overline{\Omega_i} u_j \hat{e}_k + \rho \overline{\Omega_i}\,\overline{\Omega_j}\,\overline{R_i}\hat{e}_j - \rho \overline{\Omega_i}^2\,\overline{R_k}\hat{e}_k \tag{2.54}$$

$$Pr_T = \frac{c_p \mu_T}{k} \tag{2.55}$$

2.2 Solid Continuum

When developing a fluid-structure interaction technique, one typically has freedom deciding which governing equations will be utilized for the solid continuum. Many researchers [6] [16] [31] [38] [55] [70] [58] choose to use the modal equations as solid governing equations. This

decision removes the possibility of the developed solver to be monolithic and may reduce capability of the structural solver to resolve high frequency behavior, modal coupling and other non-linear behavior. Furthermore, the intent of this work is to develop a fully coupled, monolithic, fluid-thermal-structural simulation technique. In the present work, The momentum conservation equations are written in terms of the Piola-Kirchhoff stresses. The stress tensor is related to the Lagrangian strain tensor through the St. Venant-Kirchhoff constitutive relationship. The governing equations for the solid continuum come from elasticity, where there are three fundamental components of elasticity: mathematical relationships linking stresses and forces without consideration of the continuum displacement, mathematical relationships describing the continuum deformations without consideration of the forces that caused these displacements, and a constitutive relationship relating the loads to the displacements.

2.2.1 Equilibrium Relationships

Consider a three dimensional solid continuum of any shape as shown in figure 2.4, subject to an arbitrary body force acting on the domain, Γ, and an external load acting on the boundary, $\partial\Gamma$ in static equilibrium. Let the continuum be cut at an arbitrary plane into two pieces Γ_1 and Γ_2. To preserve equilibrium, a force must be imposed on Γ_1 prescribing a displacement $\vec{x}$ acting on an infinitesimal area $|S|$ with a unit normal $\hat{n}$. This force F represents the contact forces domain Γ_2 imparts upon Γ_1. It is convenient to represent this force vector as a traction vector, the boundary which it acts on can be tracked instead of the area which is acted upon. This can be achieved if the force vector is divided by the infinitesimal area, which will result in the traction vector, $\vec{t}$, which is dependent only on the

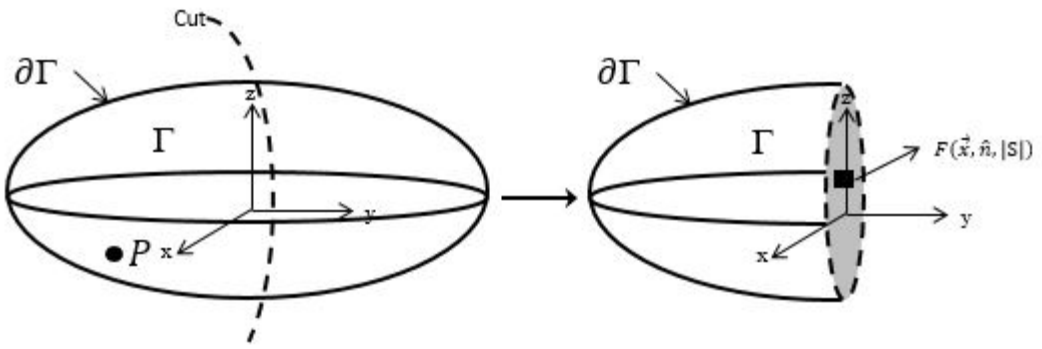


Figure 2.4: Three dimensional Solid Continuum subject to external loads

position and the unit normal.

$$\vec{t} = \lim_{|S|\to 0} \frac{F(\vec{x}, \hat{n}, |S|)}{|S|} = \vec{t}(\vec{x}, \hat{n}) \tag{2.56}$$

The traction vector can be defined in the current coordinate system by considering an infinitesimal tetrahedron in equilibrium, shown in figure 2.5. For a traction acting on the face of the outward normal, a resisting traction can be applied along each coordinate direction, resulting in equation 2.57.

$$\vec{t}(\vec{x}, \hat{n}_1) + \vec{t}(\vec{x}, \hat{n}_2) + \vec{t}(\vec{x}, \hat{n}_3) = \vec{t}(\vec{x}, \hat{n}) \tag{2.57}$$

To obtain the desired result, let us define a tensor σ_{ji}, such that $\vec{t_n} = t_i = \sigma_{ji} \cdot n_i$. Note that from this definition the traction is understood to act in the i direction on an infinitesimal area S which has an area normal in the j direction. Now consider that the continuum domain presented in figure 2.4 is subject to a body force per unit mass ρb_i and traction forces and is

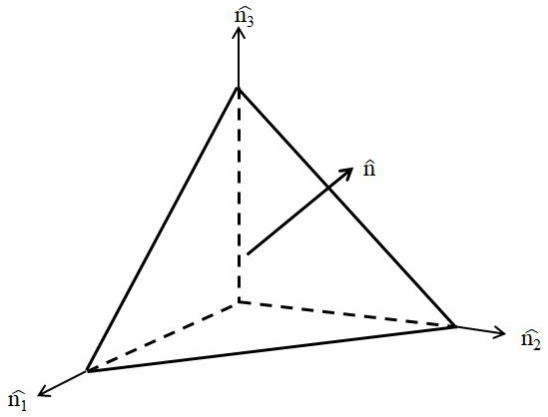


Figure 2.5: Infinitesimal in equilibrium tetrahedron subject to normal traction loads

in equilibrium. Applying Newton's Second Law, $\vec{F} = m\vec{a}$ the equilibrium equation becomes:

$$\int_{\partial\Gamma} \sigma_{ji} \cdot n_j dA + \int_{\Gamma} \rho b_i dV_{ol} = 0 \tag{2.58}$$

To obtain a partial differential equation instead of an integral equation for the equilibrium equation, the Gauss' Divergence theorem will be applied. Note that Gauss' divergence theorem states that for any sufficiently smooth vector field $\vec{f}(\vec{x})$ with components $f_i(\vec{x})$ the following applies:

$$\int_{\Gamma} \frac{\partial f}{\partial x_i} dV_{ol} = \int_{\partial\Gamma} f_i n_i dA \tag{2.59}$$

Applying Gauss' theorem to equation 2.58 will result in the following:

$$\int_{\Gamma} (\frac{\partial \sigma_{ji}}{\partial x_j} + \rho b_i) dV_{ol} = 0 \tag{2.60}$$

For the above statement to be true, the intagrand must be equal to zero which will result in the desired relationship between forces and stresses completing the first pillar of solid

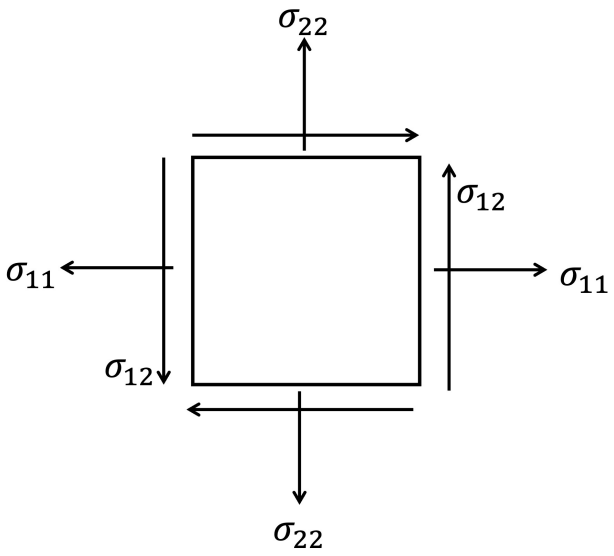


Figure 2.6: Planar 2D element

mechanics and elasticity theory.

$$\frac{\partial \sigma_{ji}}{\partial x_j} + \rho b_i = 0 \tag{2.61}$$

Before moving forward to the second pillar of solid mechanics it is necessary to further characterize the stress tensor σ_{ji}. Consider a two dimensional infinitesimal area in equilibrium with tractions applied as shown in figure 2.6. For this object to be in static equilibrium the stress tensor σ_{ji} is symmetric. Even if the planar element is not in static equilibrium, the Cauchy stress is always symmetric.

If the solid continuum shown in figure 2.4 is not in static equilibrium and has an outside force acting upon it, the above analysis is still valid. The force acting on a continuum element can be added to the right hand side of equation 2.61 as an acceleration a_i, to create the solid momentum equation 2.62.

$$\frac{\partial \sigma_{ji}}{\partial x_j} + \rho b_i = \rho a_i \tag{2.62}$$

2.2.2 Displacement Relationships

The particle P in the continuum domain from figure 2.4 can be subjected to some displacement field described by $\vec{u}(\vec{x}) = P(\vec{x}_{new}) - P(\vec{x}_0)$, where $\vec{x}_0$ represents the initial positions of the particle P. Assume the displacement of the particle P is due entirely to local stretching and shearing, and no rigid body movement is occurring. Imagine a two dimensional element who's initial geometry looks similar to that of figure 2.6. As the stress components σ_{ji} are increased, the square will begin to deform into a rhombus. It is reasonable to expect that the material strain will also be a rank two tensor to enable some future direct mapping between stress and strain. Let ε_{ij} be a symmetric rank two tensor, which will quantify the local stretching and shearing of material. Without limiting this analysis to small deformations the strain is defined in equation 2.63. In this definition of strain, D_i is the displacement of a material element and X_i is the undeformed reference coordinate. Strain due to thermal expansion/contraction of a material may also be considered by adding $\delta_{kk}\beta_x(T_S - T_R)$ to the right hand side of equation $eqn : strain$, where β_x is the coefficient of thermal expansion T_s is the solid continuum temperature and T_R is the reference/initial temperature of the solid.

$$E_{ij} = \frac{1}{2}\left(\frac{\partial D_i}{\partial X_j} + \frac{\partial D_j}{\partial X_i} + \frac{\partial D_k \partial D_k}{\partial X_i \partial X_j}\right) \tag{2.63}$$

Looking at equation 2.63, the strain of any given point within a continuum can be calculated given a displacement field. But it is not clear how to find a displacement field given the strain field, since there are six partial deferential equations that define the strain and only three unknowns in the displacement field. Restrictions must be placed on the strain field so meaningful displacements can be determined from the strain field. Consider a known small strain strain field is imposed on the body, so the higher order terms in equation 2.63 can be

discarded. In this strain field the displacement of point b may be written as:

$$u_i^b = u_i^a + \int_\beta \frac{\partial u_i}{\partial x_j} l_j \, ds \tag{2.64}$$

Given that any rank two tensor, such as the displacement gradient, may be represented as the sum of the symmetric and anti-symmetric parts the following relation will be used to simplify equation 2.64 ε_{ij} and the anti-symmetric part ω_{ij}.

$$\frac{\partial u_i}{\partial x_j} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) + \frac{1}{2}(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}) = \varepsilon_{ij} + \omega_{ij} \tag{2.65}$$

Let $\xi(s)$ be the function which defines the curve of interest, then the tangent vector $\vec{l} = \frac{\partial \xi(s)}{\partial s}$, then equation 2.64 becomes:

$$u_i^b = u_i^a + \int_\beta (\varepsilon_{ij} + \omega_{ij}) \frac{\partial \xi_j(s)}{\partial s} ds \tag{2.66}$$

Now let $\xi_j = \xi_j + x^a$, this substitution will allow us to relate the strain vector with the position of point a. Once this substitution is made, equation 2.64 can be further expanded using integration by parts which will yield the following:

$$u_i^b = u_i^a + \omega_{ij}^a (x_j^b - x_j^a) + \int_\beta (\varepsilon_{ij} l_j + \frac{\partial \omega_{ij}}{\partial x_k}(x_j^b - \xi_j) l_k) ds \tag{2.67}$$

It would be more convenient to represent the anti-symmetric part of the spatial gradient of the displacement ω_{ij} in terms of the strain, so we use the following substitution:

$$\frac{\partial \omega_{ij}}{\partial x_k} = \frac{\partial \varepsilon_{ik}}{\partial x_j} - \frac{\partial \varepsilon_{jk}}{\partial x_i} \tag{2.68}$$

Substitution of equation 2.68 into equation 2.67 gives the following:

$$u_i^b = u_i^a + \omega_{ij}^a(x_j^b - x_j^a) + \int_\beta (\varepsilon_{ij} l_j + (\frac{\partial \varepsilon_{ik}}{\partial x_j} - \frac{\partial \varepsilon_{jk}}{\partial x_i})(x_j^b - \xi_j) l_k) ds \tag{2.69}$$

Equation 2.69 quantifies the displacement of an arbitrary point b given the rotation and displacement of a point a. This result is only physically meaningful if and only if the line integral over β is path independent.

This observation is rooted in physics because the displacement field should depend only on the imposed strain field and should not change depending on the path which is used to calculate the displacements. It is important to note that this restriction requires that the strain field be continuous and that the domain must be simply connected, therefore special care must be taken near material discontinuities such as cracks. This observation will impose some restrictions on the intagrand in equation 2.69 as follows:

$$\frac{\partial}{\partial x_l}(\varepsilon_{ij} + (\frac{\partial \varepsilon_{ik}}{\partial x_j} - \frac{\partial \varepsilon_{jk}}{\partial x_i})) = \frac{\partial}{\partial x_k}(\varepsilon_{il} + (\frac{\partial \varepsilon_{il}}{\partial x_j} - \frac{\partial \varepsilon_{jl}}{\partial x_i})) \tag{2.70}$$

Expanding equation 2.70 will result in the so called Compatibility equations:

$$\frac{\partial \varepsilon_{ij}}{\partial x_k \partial x_l} + \frac{\partial \varepsilon_{kl}}{\partial x_i \partial x_j} - \frac{\partial \varepsilon_{ik}}{\partial x_j \partial x_l} - \frac{\partial \varepsilon_{jl}}{\partial x_i \partial x_k} = 0 \tag{2.71}$$

These equations are restrictions on the strain field to ensure that any strain field is unique and will therefore result in a single value of displacement. Equation 2.71 has four free indices which result in a total of 81 possible values. However, many of these are identically zero. There are actually only six distinct equations which result from different combinations of $ijkl = 1123$ and $ijkl = 1122$.

2.2.3 Constitutive Relationships

Now we will derive a relationship between stress and strain. It is conceptually helpful to think of a material element as a linear spring analogous to Hooke's Law for a spring $\vec{F} = k\vec{x}$, where k is the spring constant, $\vec{F}$ is the force required to displace the spring by $\vec{x}$, so the work done by displacing a spring is $W = \frac{1}{2}kx^2$. A similar definition of work can be made to impose an arbitrary strain field on a material element. First, consider the principle of the conservation on energy and take the time rate of work done by the stress on a material element to be equal to the time rate of a function of strain only, which is mathematically represented in equation 2.72. This equation is valid for any elastic material because it is founded on the principle of the conservation of energy.

$$\sigma_{ji} * \dot{\varepsilon}_{ij} = \frac{\partial W}{\partial t}(\varepsilon_{ij}) \Rightarrow (\sigma_{ji} * -\frac{\partial W}{\partial \varepsilon_{ij}})\dot{\varepsilon}_{ij} = 0 \Rightarrow \sigma_{ij} = \frac{\partial W}{\partial \varepsilon_{ij}} \tag{2.72}$$

Where the work is defined analogously to the work done displacing a spring following Hooke's Law and C_{ijkl} is a rank four modulus tensor with the symmetries $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}$ which result in 21 distinct moduli.

$$W = \frac{1}{2}C_{ijkl}\varepsilon_{ij}\varepsilon_{kl} \tag{2.73}$$

Many materials of interest, such as metals, are isotropic so they have no preferred loading direction, so the 21 moduli can be reduced to two moduli. We will define the modulus tensor as follows:

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + +\mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{2.74}$$

Where λ is the $Lame'$ constant, μ is the shear modulus which can be defined in terms of the elastic modulus E and Poisson's ratio ν.

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{2.75}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{2.76}$$

Limiting the analysis to isotropic linear elasticity two distinct moduli exist resulting in the following constitutive relationship.

$$\sigma_{ij} = \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij} + 2\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{2.77}$$

2.2.4 Solid Heat Conduction Equation

Consider a small finite cube shown in figure 2.7. For simplicity, let the top, bottom, back and front surfaces be adiabatic so that there is no heat flux at those boundaries. $\dot{Q}$ is the heat flux, $\dot{G}$ is the heat generation in side the small finite cube.

Conducting an energy balance of the finite cube from figure 2.7 results in the following formulation:

$$\left[\dot{Q}_{x+\Delta x}\right] - \left[\dot{Q}_x\right] + \left[\dot{G}\right] = \frac{\Delta E}{\Delta t} \tag{2.78}$$

Take the limit as $\Delta x \to 0$ and $\Delta t \to 0$. Then, substitute the definition of heat conduction to be $\dot{Q} = k\frac{dT}{dx}$, where k is the thermal conductivity, and T is temperature. The energy stored within the finite cube can be reformulated to be $dE = c_p \rho dT$, where c_p is the specific

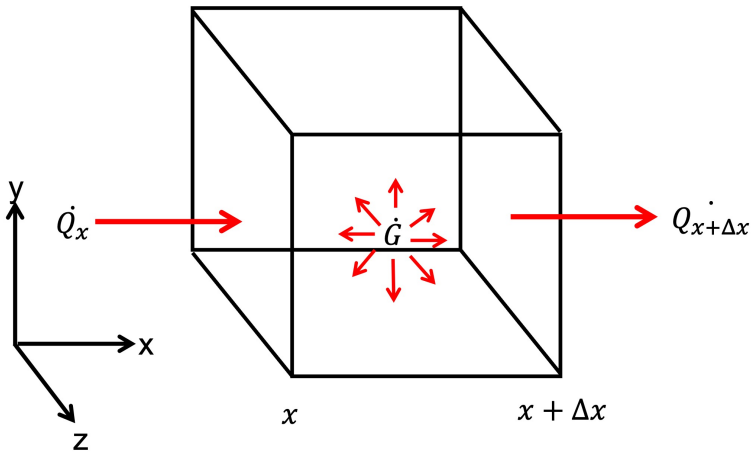


Figure 2.7: Finite cube with one dimensional heat transfer.

heat capacity and ρ is the material density, resulting in equation 2.79, where g is the heat generation.

$$\frac{d}{dx}\left[k\frac{dT}{dx}\right] + g = \rho c_p \frac{dT}{dt} \tag{2.79}$$

This equation 2.79 can be readily expanded into three dimensions by considering removing the simplifying assumption that the back, front, top and bottom faces of the finite block in figure 2.7. This results in the general heat conduction equation for an isotropic material in equation 2.80.

$$\frac{\partial T}{\partial x_j}\left[k_s\frac{\partial T}{\partial x_j}\right] = \rho_s C_{ps}\frac{\partial T}{\partial t} \tag{2.80}$$

2.2.5 The Implemented Governing Equations

Two configurations are defined for each solid cell, an instantaneous domain, denoted by x_i, and a referential configuration, denoted by X_j. The instantaneous domain reflects the actual shape and deformations applied to the current grid cell whereas the referential domain is an

undeformed cell used for integration purposes. These two domains are related to each other by the deformation gradient, F_{ij}, which is essentially a Jacobian array that allows coordinate mapping between the two domains, as shown in equation 2.81. The configuration array is related to the displacement gradient by equation 2.82, where I is the identity matrix.

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \tag{2.81}$$

$$\frac{\partial D_i}{\partial X_j} = F_{ij} - I_{ij} \tag{2.82}$$

The desired quantities in the solid blocks are the displacement, D_i, and velocity, v_i, of the nodes. The definition of the strain field, equation 2.83 allows for large deformations and thermal strain; where β_s is the coefficient of thermal expansion, T_S is the temperature of the solid and T_R is the prescribed reference temperature which is an input value. Two stress tensors are utilized to describe the physical state of the solid material, the 1^{st} and 2^{nd} Piola-Kirchhoff stresses. The 2^{nd} Piola-Kirchhoff stress, S_{ij}, is related to the strain through the St. Venant-Kirchhoff constitutive relationship, as shown in equation 2.86. The 1^{st} and the 2^{nd} Piola-Kirchhoff stresses are related using the configuration array as shown in equation 2.87. The 1^{st} Piola-Kirchhoff stress, P_{ij}, allows a direct mapping of forces to the referential configuration. This connection is exploited to relate the solid momentum equation with the stress field as described in equation 2.89. The Lamé parameters λ and μ are related to Young's Modulus, E, and Poisson's Ratio, ν, as denoted in equations 2.84 and 2.85. Finally, the 2^{nd} Piola-Kirchhoff stress and the Cauchy stress, σ_{ij}, are related by equation 2.88.

$$E_{ij} = \frac{1}{2}\left(\frac{\partial D_i}{\partial X_j} + \frac{\partial D_j}{\partial X_i} + \frac{\partial D_k \partial D_k}{\partial X_i \partial X_j}\right) + \beta_x (T_S - T_R) \tag{2.83}$$

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{2.84}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{2.85}$$

$$S_{ij} = \lambda \delta_{ij} E_{kk} + 2\mu E_{ij} \tag{2.86}$$

$$P_{ik} = F_{ij} S_{jk} \tag{2.87}$$

$$\sigma_{ik} = \frac{1}{|F|} P_{ij} F_{kj} \tag{2.88}$$

$$\rho_s \frac{\partial v_i}{\partial t} - \frac{\partial P_{ij}}{\partial X_j} = f_i \tag{2.89}$$

$$\frac{\partial T}{\partial x_j}\left[k_s \frac{\partial T}{\partial x_j}\right] = \rho_s C_{ps} \frac{\partial T}{\partial t} \tag{2.90}$$

These equations are solved using numerical methods discussed in chapter 3.

Chapter 3

Numerical Methodology

This chapter will discuss the numerical methods that are implemented in the MBFLO3 code package. The methods presented are used to solve the governing equations for the fluid and solid continua and maintain the numerical stability. The algorithms used to develop unique features such as the implementation overlaid/overset grid to model film cooling holes and the fluid-structure interaction will be discussed.

3.1 MBFLO3 Overview

MBFLO3 is an in-house monolithic, fully coupled, finite volume based, fluid-thermal-structure interaction code. It is written almost entirely in FORTRAN90 and has massively parallel capabilities using Message Passing Interface (MPI), OpenMP, and OpenACC, and is specialized for turbomachinery. MBFLO3 was originally created by Bozinoski and Davis [3] [2]. Various improvements have been made by many contributors including Bates [1], Gottlieb [9] and Lee [42]. MBFLO3 is a complex collection of nearly 200 subroutines. A highly simplified

flow chart for the program is shown in figure 3.1.

3.1.1 Cell Centered Control Volume Technique

The Lax-Wendroff integration method used in MBFLO3, implemented by Bozinoski [2], is combined with Ni's distribution scheme [50] as described and applied by Dannenhofer [10] and Davis et. al. [12]. The Lax-Wendroff technique is a spatially second order accurate time marching scheme which is based upon a Taylor Series expansion, equation 3.1. The implementation of the Lax-Wendroff integration with Ni's distribution scheme are best understood with a simple one dimensional example. Let $F(U)$ be some function applied to a vector of conserved variables U defined in equation 2.23.

$$\delta U_i = U_i^{n+1} - U_i^n = \Big(\frac{\partial U}{\partial t}\Big)_i^n \Delta t + \Big(\frac{\partial^2 U}{\partial t^2}\Big)_i^n \frac{\Delta t^2}{2} + H.O.T. \tag{3.1}$$

$$\frac{\partial U}{\partial t} = -\frac{\partial F(U)}{\partial x} \tag{3.2}$$

Equation 3.1 describes how the change of an independent variable, U, between time steps can be broken up into first-order and second-order time rate changes. Using equation 3.2, the first order changes at a nodal point i can be found using a central differencing scheme. For simplicity, consider a uniform grid with grid spacing $\Delta x_{i-1/2} = \Delta x_{i+1/2} = \Delta x$. Note that$\Delta x_{i-1/2}$ and $\Delta x_{i+1/2}$ indicate cell centered quantities.

$$\Big(\frac{\partial U}{\partial t}\Big)_i^n = \frac{1}{2}\Big[\Big(\frac{\partial U}{\partial t}\Big)_{i+1/2}^n + \Big(\frac{\partial U}{\partial t}\Big)_{i-1/2}^n\Big] \tag{3.3}$$

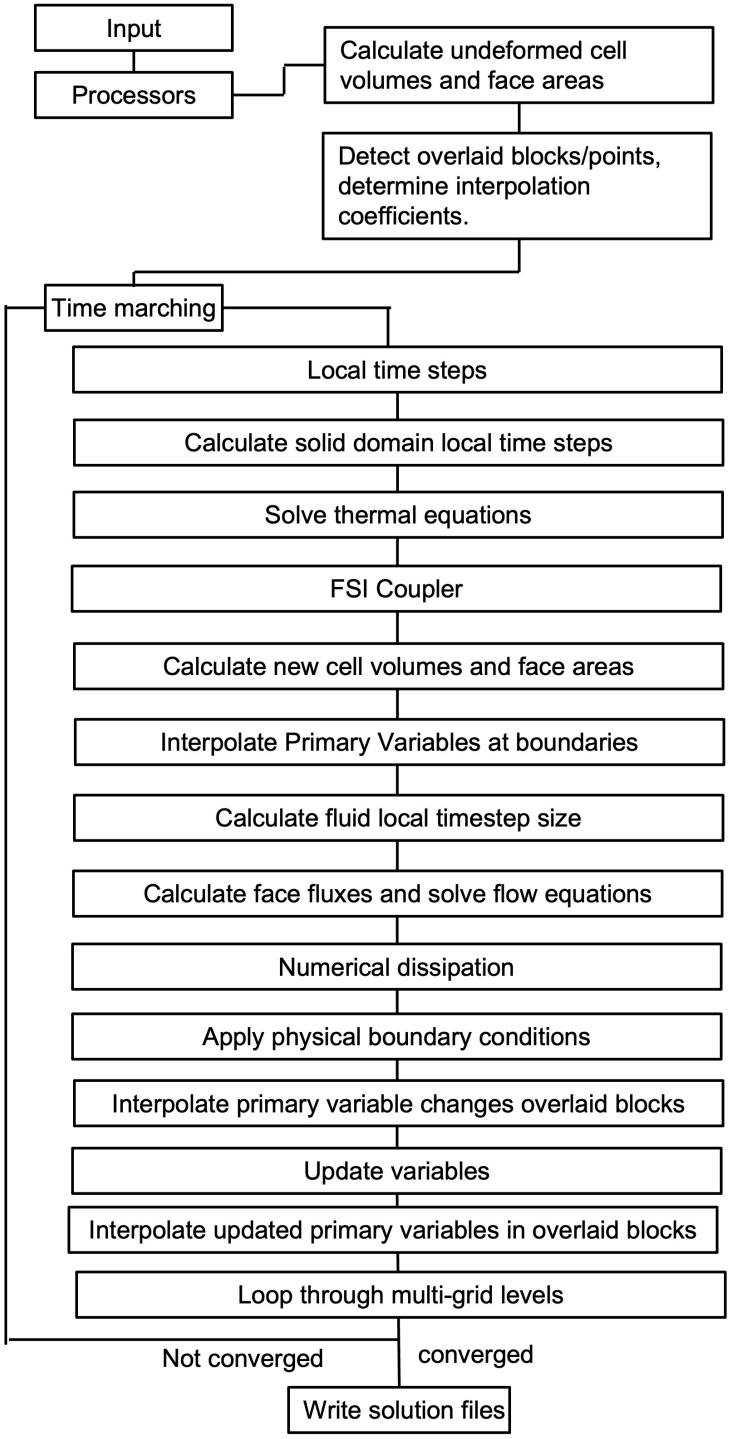


Figure 3.1: Simplified MBFLO3 Flow Chart

where,

$$\begin{aligned}\Big(\frac{\partial U}{\partial t}\Big)^{n}_{i+1/2} &\cong \frac{F^{n}_{i} - F^{n}_{i+1}}{\Delta x_{i+1/2}} \\ \Big(\frac{\partial U}{\partial t}\Big)^{n}_{i-1/2} &\cong \frac{F^{n}_{i-1} - F^{n}_{i}}{\Delta x_{i-1/2}}\end{aligned} \tag{3.4}$$

The second order changes can be found using a similar approach by expanding 3.2 and using 3.3.

$$\frac{\partial^2 U}{\partial t^2} = -\frac{\partial}{\partial x}\frac{\partial F}{\partial U}\frac{\partial U}{\partial t} \tag{3.5}$$

where,

$$-\frac{\partial}{\partial x}\frac{\partial F}{\partial U}\frac{\partial U}{\partial t} \cong -\frac{1}{\Delta x_i}\Big[\Big(\frac{\partial F}{\partial U}\Big)^{n}_{i+1/2}\Big(\frac{\partial U}{\partial t}\Big)^{n}_{i+1/2} - \Big(\frac{\partial F}{\partial U}\Big)^{n}_{i-1/2}\Big(\frac{\partial U}{\partial t}\Big)^{n}_{i-1/2}\Big] \tag{3.6}$$

By combining equations 3.1, 3.3 and 3.6 the Lax-Wendroff time marching scheme for the simple one dimensional problem is shown in equation 3.7.

$$\delta U_i = \big(F^{n}_{i-1} - F^{n}_{i+1}\big)\frac{\Delta t}{2\Delta x} + \Big[\Big(\frac{\partial F}{\partial U}\Big)^{n}_{i-1/2}\big(F^{n}_{i-1} - F^{n}_{i}\big) - \Big(\frac{\partial F}{\partial U}\Big)^{n}_{i+1/2}\Big(F^{n}_{i} - F^{n}_{i+1}\Big)\Big]\frac{\Delta t^2}{2\Delta x^2} \tag{3.7}$$

As written in equation 3.7, the number of numerical operations become large when scaled to two or three-dimensions. However, by implementing Ni's [50] distribution formulas and the above steps, the predicted changes at a nodal points can be found in a single step. The first order changes based on the cell centered terms are shown in equation 3.8. The second

order changes are shown in equation 3.9.

$$\begin{aligned} \Delta U_{i-1/2} &= \big(F^n_{i-1} - F^n_i\big)\frac{\Delta x}{\Delta t} \\ \Delta U_{i+1/2} &= \big(F^n_i - F^n_{i+1}\big)\frac{\Delta x}{\Delta t} \end{aligned} \tag{3.8}$$

$$\Delta F \equiv \Big(\frac{\partial F}{\partial U}\Big)\Delta U \tag{3.9}$$

The predicted changes at the nodal points can be calculated in a single step by combining equations 3.8, 3.9 with 3.7, resulting in equation 3.10. Upon examination of equation 3.10, it can be observed that the changes experienced at nodal point i is influenced by changes up and down stream of the node.

$$\delta U_i = \frac{1}{2}\Big[\Delta U_{i-1/2} + \frac{\Delta t}{\Delta x}\Delta F_{i-1/2}\Big] + \frac{1}{2}\Big[\Delta U i + 1/2 - \frac{\Delta t}{\Delta x}\Delta F_{i+1/2}\Big] \tag{3.10}$$

With the above findings, Ni [50] developed a control volume centered integration method that consists of three basic steps. First, the governing equation are approximated at the cell center, $x_{i+1/2}$ using equation 3.11.

$$\Delta U_{i+1/2} = \big(F^n_i - F^n_{i-1}\big)\frac{\Delta t}{\Delta x} \tag{3.11}$$

Second, the corrections made to point i and $i+1$ are determined with the use of Ni's

distribution formulas, equation 3.12.

$$\begin{aligned}(\delta U_i)_{i+1/2} &= \frac{1}{2}\Big[\Delta U_{i+1/2} - \frac{\Delta t}{\Delta x}\Delta F_{i+1/2}\Big] \\ (\delta U_{i+1})_{i+1/2} &= \frac{1}{2}\Big[\Delta U_{i+1/2} - \frac{\Delta t}{\Delta x}\Delta F_{i+1/2}\Big]\end{aligned} \tag{3.12}$$

Third, the primary variables are updated, using equation 3.13.

$$\begin{aligned}U_i^{n+1} &= U_i^n + \delta U_i \\ \delta U_i &= \big(\delta U_{i+1}\big)_{i-1/2} + \big(\delta U_i\big)_{i+1/2}\end{aligned} \tag{3.13}$$

The integration procedure described above is second order accurate in time and space. This procedure is also explicit, so the time step size is governed by the Currant-Friedrichs-Lewi (CFL) condition, shown in equation 3.14, where W is the wave speed [50].

$$\Delta t = \frac{\Delta x}{|W|} \tag{3.14}$$

The solid momentum and heat conduction equations are diffusion like, and do not contain first derivative convection like terms that require second order changes for stability. The changes in solid node velocity due to displacement are calculated using equation 3.8, and are then distributed from the cell centers to the nodes.

3.1.2 Time Marching

The point-implicit dual time step method formulated by Jameson[32] is used for unsteady simulations in the current study, described in equation 3.15; where n is the global time step, m is the inner iteration index, and w represents a primary variable. This method is marched

implicitly by a specified value of Δt_G, but the equations are solved at each global time step iteratively using a local time step of Δt_L until the residual drops below a tolerance. Equation 3.15 is second order accurate in time.

$$w^{m+1} = F(w^m)\Delta t_L + \frac{-3w^m + 4w^n - w^{n-1}}{2\Delta t_G}\Delta t_L + w^m \tag{3.15}$$

Each grid cell has an independent local time step size based on a specified Courant-Friedrichs-Lewy (CFL) number, the geometry of the cell, and continuum types and properties. The fluid and solid time-steps are different and are based upon the physics and numerical stability considerations for each domain. For the fluid domain, the local time-step, equation 3.16, uses two stability considerations; the first is related to local wave speed c, and the second related to the grid cell minimum Reynolds number α, where V_{ol} is cell volume, μ_{total} is the sum of the laminar viscosity and an approximation to the numerical viscosity due to numerical smoothing, α is the cell minimum cell Reynolds number, and c is the speed of sound.

$$\Delta t_L = CFL_f \ min\Big(\frac{\Delta L}{V_{ol} + c + \frac{\alpha\mu_{total}}{\Delta L}}\Big) \tag{3.16}$$

The solid domain local time-step, equation 3.17, is dependent on the wave speed and the cell dimensions. V_{ol} is cell volume, E is the modulus of elasticity, ρ is the material density, A_{is}, A_{js}, A_{ks} are one half of the root-sum-square of the cell face areas, and i, j, k subscripts refer to the three-dimensional grid directions.

$$\Delta t_L = CFL_s \ min\Big(\frac{V_{ol}}{2\sqrt{\frac{E}{\rho}}(A_{is} + A_{js} + A_{ks})}\Big) \tag{3.17}$$

The solid momentum equation 2.89 is a second order partial differential equation in terms of displacement. Therefore, a second time integration is necessary to determine the nodal displacement. This is completed using equation 3.18, which is a second order accurate backward difference equation in time; where m and n are the inner and outer iteration indices respectfully, D is the nodal displacement, Δt_G is the global time step, and v is the nodal velocity.

$$D^{m+1} = \frac{2v^{m+1}\Delta t_G + 4D^n - D^{n-1}}{3} \tag{3.18}$$

The dual time stepping method presented in equation 3.15 is the chosen method of time-stepping because the point-implicit method allows for the simulation to be executed with a larger global time step. The global time step may be two to four orders in magnitude larger than the explicit time step given by equation 3.17. As will demonstrated in section 4.1.1, the dual time stepping method can typically reduce simulation time by 25%.

3.1.3 Numerical Smoothing Techniques

Numerical smoothing techniques are implemented in the fluid and solid blocks to maintain numerical stability and capture complex fluid flow structures in the fluid domain.

Fluid Domain

Jameson smoothing [33] is implemented in the MBFLO3 flow solver to capture shock waves and reduce solution decoupling. The result of equation 3.19 is added to the primary variable updates for the current inner iteration. Equation 3.22 is used to determine the 4^{th} difference smoothing coefficient, where $f(\tau_{ij}) = \sqrt{S_{xy}^2 + S_{xz}^2 + S_{yz}^2}$ is the damping function that is

responsible for shutting off the smoothing in viscous flow regions, $\alpha^{(2)} = 0.1$, $\alpha^{(4)} = 0.01$, $C_{av} = 1 \times 10^{-5}$. More details on the implementation of this smoothing method can be found in Bozinoski [3].

$$F_{i+\frac{1}{2}} = F_{i+\frac{1}{2}} - d_{i+\frac{1}{2}} \tag{3.19}$$

$$d_{i+\frac{1}{2}} = \epsilon^{(2)}(U_{i+1} - U_i) - \epsilon^{(4)}(U_{i+2} - 3U_{i+1} + 3U_i - U_{i-1}) \tag{3.20}$$

$$\epsilon^{(2)} = \alpha^{(2)} dp_i \tag{3.21}$$

$$\epsilon^{(4)} = \alpha^{(4)} \Big[min\Big(1, \frac{1}{max(C_{av}, f(\tau_{ij})}\Big)\Big]^2 \tag{3.22}$$

$$dp_i = max(\Delta P_{i-1}, \Delta P_i, \Delta P_{i+1}) \tag{3.23}$$

$$\Delta P_i = \Big| \frac{P_{i-1} - 2P_i + P_{i+1}}{P_{i-1} + 2P_i + P_{i+1}} \Big| \tag{3.24}$$

Solid Domain Smoothing Techniques

Due to the second order accurate discretization of the solid domain using finite volumes, a numerical issue arises called hourglassing. This phenomena is similar to the odd even decoupling which can occur in central-difference and finite-difference finite volume solvers. An example of hourglassing is shown in figure 3.2. This section will discuss the two solid domain smoothing techniques that aim to prevent hourglassing without significantly impacting the solution. The two smoothing methods considered are hourglass stiffeniing/smoothing proposed by Gottfried and Fleeter [22] and Flanagan and Belytschko [19], and a modified version of Jameson's [33] second and fourth difference smoothing techniques.

Hourglass Stiffening and Smoothing

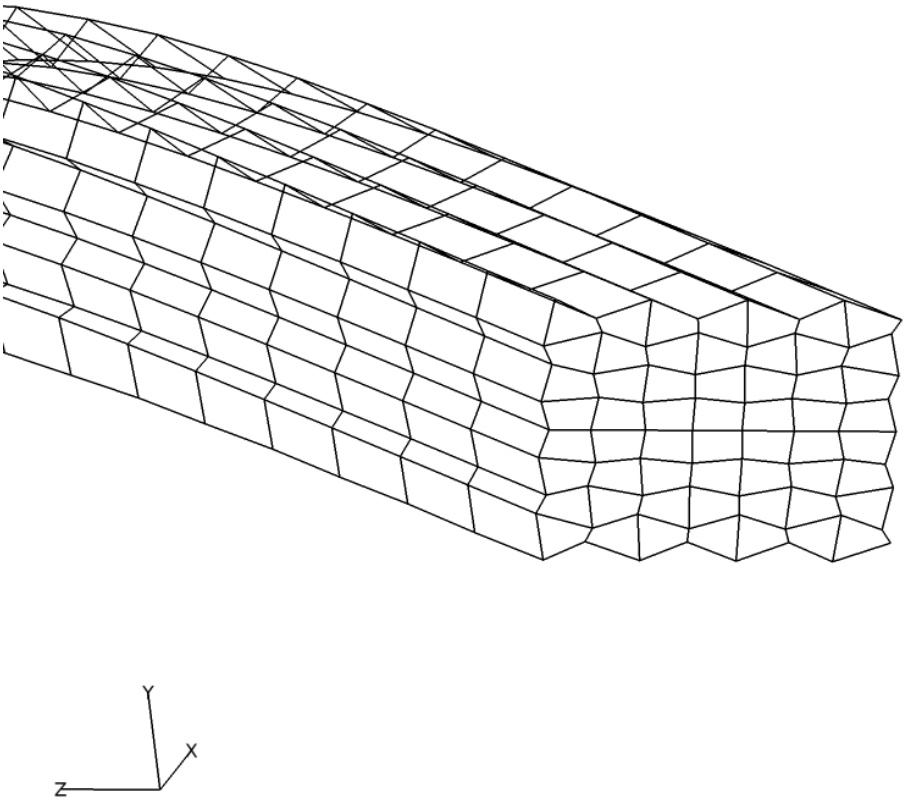


Figure 3.2: Hourglassing example.

For hexahedral cells, there are eight deformation modes; four of the eight deformation modes pertain to rigid body motion, expansion/contraction, and two shear modes that are all permissible modes of deformation. However, four hourglass modes exist which represent unconstrained motion that can lead to the destruction (decoupling) of the solution. Each hourglass mode has a corresponding basis vector denoted by Γ_{lk}, listed in table 3.1. Hourglass modes are represented pictorially in figure 3.3. To prevent the destruction of the solution, a fic-

titious force is introduced at each solid node to discourage the unconstrained movement. The hourglass force is described by equation 3.27; where F_{il}^{hg} is the artificial smoothing force, h_{ij} is the hourglassing velocity/displacement and is measure of how much a node is allowed to move in an unconstrained manner, and Q_{hg} is the hourglass coefficient. The hourglass displacement and hourglass velocity are shown in equations 3.25 and 3.26 respectfully. Hourglass stiffening uses the displacement based definition of h_{lk} in equation 3.25, while hourglass smoothing uses the velocity based definition of h_{lk} in equation 3.26. The hourglass force is ultimately added to the right hand side of equation 2.89 as an additional nodal force. MBFLO3 has the capability to use either hourglass stiffening or hourglass smoothing. Gottfried and Fleeter[22] recommended Q_{hg} to be between 0.05 and 0.15.

$$h_{lk} = \frac{D_{il}\Gamma_{lk}}{\Delta t_L} \tag{3.25}$$

$$h_{lk} = u_{il}\Gamma_{lk} \tag{3.26}$$

$$F_{il}^{hg} = -\rho V^{2/3}(0.25Q_{hg}A + 25Q_{hg}^2|h_{ik}|)h_{ik}\Gamma_{lk} \tag{3.27}$$

Table 3.1: Hourglass modes notation for equations 3.25 and 3.27[8]

	Local Node Number							
Deformation Mode	1	2	3	4	5	6	7	8
Hourglass 1 Γ_{l1}	-1	-1	1	1	1	1	-1	-1
Hourglass 2 Γ_{l2}	-1	1	1	-1	1	-1	-1	1
Hourglass 3 Γ_{l3}	1	-1	1	-1	1	-1	1	-1
Hourglass 4 Γ_{l4}	1	-1	1	-1	-1	1	-1	1

2^{nd} and 4^{th} Difference Smoothing

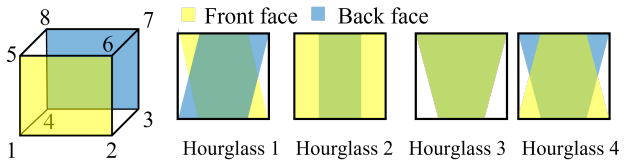


Figure 3.3: Description of Hourglass Deformation modes.

Hourglass stiffening reduces the maximum stable time step the solid solver may take. The smoothing control parameter Q_{hg} and the maximum stable time step are inversely related [19]. An additional method of smoothing has been investigated to be used in conjunction with hourglass stiffening/smoothing to enable larger global time steps and increased numerical stability. A modified 2^{nd} and 4^{th} difference smoothing technique formulated by Jameson [33] has been implemented. Zorn [71] tried using the 2^{nd} and 4^{th} differencnig equations instead of hourglass stiffening and found these methods to be overly dissapative for use with a structural solver. Rather than apply the 2^{nd} and 4^{th} difference smoother to the entire solid domain, as Zorn [71] did, only the nodal displacements or velocities of free surfaces are smoothed.

The 2^{nd} difference smoothing implemented within the structural solver uses equations 3.28 and 3.1.3. The 4^{th} difference smoothing uses equations3.28 and 3.1.3. The implementation of these equations is verified using the cantilever beam subject to a gravitational load gravity case in 4.1.1.

$$b_i^s = b_i - d_i \tag{3.28}$$

$$d_i = \epsilon(U_{i+1} - U_i) \tag{3.29}$$

$$d_i = \epsilon(U_{i+1} - U_i) - \epsilon(U_{i+2} - 3U_{i+1} + 3U_i - U_{i-1}) \tag{3.30}$$

3.1.4 Physical Boundary Conditions

MBFLO3 is capable of simulating a multitude of boundary conditions. This section will focus on the boundary condition types and improvements that were developed specifically for this effort. For additional details on the development and implementation of MBFLO3 boundary conditions see [3], [42], [1] and [9].

Conjugate Heat Transfer Boundary Conditions

The temperature field of solid blocks is modeled using the heat conduction equation, shown in equation 2.80. The heat conduction equation is solved using the same numerical approach as the fluid solver. Temperature and temperature dependent values are message passed from fluid blocks to neighboring solid blocks at every inner iteration. This approach requires point matched and overset grids to be generated at all fluid-solid interfaces. To minimize the interblock communication, only the temperature of the first two fluid nodes off of the solid surface, and the thermal conductivity of the fluid, are message passed to the solid block. This information is stored in the solid block's ghost nodes.The fluid thermal conductivity is calculated at every inner iteration using equation 3.31, where C_{pf} is the thermal capacity of the fluid at a constant pressure, μ is the dynamic viscosity and Pr is the Prantdl number.

$$k_f = \frac{C_{pf}\mu}{Pr} \tag{3.31}$$

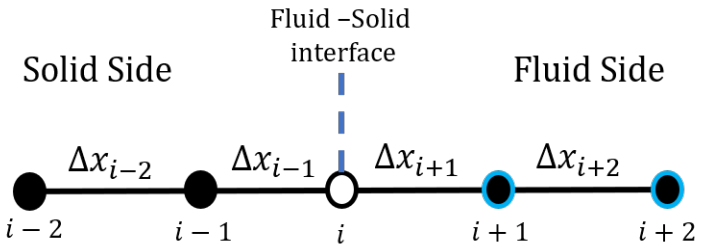


Figure 3.4: Central differencing stencil

There are two physical requirements for a conjugate interface condition. First, to satisfy energy conservation, the heat flux on either side of the interface is constant, as described in equation 3.32. Where k_f and k_s are the thermal conductivity of the fluid and solid respectively. Indices $i+$ and $i-$ represent right and left hand limits at the fluid-solid interface. Second, the temperature on either side of the fluid-solid boundary must be the same.

$$k_s \frac{\partial T}{\partial x}\Big|_{i-} = k_f \frac{\partial T}{\partial x}\Big|_{i+} \tag{3.32}$$

The central differentiating formulas implemented in MBFLO3 can be derived using Lagrange Polynomials, equation 3.33. The Lagrange interpolation formula for $n+1$ points will result in an N^{th} order accurate numerical scheme.

$$\phi(x) = \sum_{j=0}^{n} \prod_{k=0, j\neq k}^{n} \frac{x - x_k}{x_j - x_k} \phi_j \tag{3.33}$$

The first order accurate central differencing scheme can be derived by setting $n = 1$, and $n = -1$ resulting in two Lagrange polynomials shown in equations 3.34, and 3.35.

$$\phi(x) = \frac{(x - x_1)}{(x_0 - x_1)} \phi_0 + \frac{(x - x_0)}{(x_1 - x_0)} \phi_1 \tag{3.34}$$

$$\phi(x) = \frac{(x - x_{-1})}{(x_{-1} - x_0)}\phi_{-1} + \frac{(x - x_{-1})}{(x_0 - x_{-1})}\phi_0 \tag{3.35}$$

Then, the derivative of $\frac{\partial \phi}{\partial x}\big|_{x_0}$ can be evaluated. Consider the first order forward difference derivative, equation 3.36, as the fluid side of the boundary, and the first order backward difference derivative, equation 3.37, as the solid side of the boundary.

$$\frac{\partial \phi}{\partial x}\Big|_{x_0} = -\frac{\phi_0}{\Delta x_{i+1}} + \frac{\phi_{i+1}}{\Delta x_{i+1}} \tag{3.36}$$

$$\frac{\partial \phi}{\partial x}\Big|_{x_0} = -\frac{\phi_0}{\Delta x_{i-1}} + \frac{\phi_{i-1}}{\Delta x_{i-1}} \tag{3.37}$$

Then, $T_n k_m$ is substituted for ϕ_n, where k_m is the thermal conductivity of the node's assigned continuum. Finally, the two quantities are set equal to each other and the temperature at the interface is solved for, shown in equation 3.38, where k_s and k_f is the thermal conductivity of the solid and fluid respectively; T_s and T_f the temperatures at the solid and fluid nodes respectively.

$$T_0 = \frac{\frac{k_f T_f^{i+1}}{\Delta x_{i+1}} + \frac{k_s T_s^{i-1}}{\Delta x_{i-1}}}{\frac{k_f}{\Delta x_{i+1}} + \frac{k_s}{\Delta x_{i-1}}} \tag{3.38}$$

The second order accurate central differencing scheme can be derived by setting $n = 2$, and $n = -2$ in equation 3.33. The result is two Lagrange polynomials shown in equations 3.39, and 3.40.

$$\phi(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\phi_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\phi_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\phi_2 \tag{3.39}$$

$$\phi(x) = \frac{(x-x_{-1})(x-x_{-0})}{(x_{-2}-x_{-1})(x_{-2}-x_0)}\phi_{-2} + \frac{(x-x_{-2})(x-x_0)}{(x_{-1}-x_{-2})(x_{-1}-x_0)}\phi_{-1} + \frac{(x-x_{-1})(x-x_{-2})}{(x_0-x_{-1})(x_0-x_{-2})}\phi_0 \tag{3.40}$$

Consider the second order forward difference derivative, equation 3.41, as the fluid side of the boundary, and the second order backward difference derivative, equation 3.42, as the solid side of the boundary. Evaluating $\frac{\partial\phi}{\partial x}\big|_{x_0}$ of equations 3.39, and 3.40 results in equations 3.41 and 3.42.

$$\frac{\partial\phi}{\partial x}\Big|_{x_0} = -\frac{2\Delta x_{i+1} - \Delta x_{i+2}}{\Delta x_{i+1}(\Delta x_{i+1} + \Delta x_{i+2})}\phi_0 + \frac{\Delta x_{i+1} - \Delta x_{i+2}}{\Delta x_{i+1}\Delta x_{i+2}}\phi_{i+1} - \frac{\Delta x_{i+1}}{(\Delta x_{i+1} + \Delta x_{i+2})\Delta x_{i+2}}\phi_{i+2} \tag{3.41}$$

$$\frac{\partial\phi}{\partial x}\Big|_{x_0} = \frac{2\Delta x_{i-1} + \Delta x_{i-2}}{\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_{i-2})}\phi_0 - \frac{\Delta x_{i-1} + \Delta x_{i-2}}{\Delta x_{i-1}\Delta x_{i-2}}\phi_{i-1} + \frac{\Delta x_{i-1}}{(\Delta x_{i-1} + \Delta x_{i-2})\Delta x_{i-2}}\phi_{i-2} \tag{3.42}$$

Then, $T_n k_m$ is substituted for ϕ_n, where k_m is the thermal conductivity of the node's assigned continuum. Finally, the two quantities are set equal to each other and the temperature at the interface is solved for, shown in equation 3.43, where k_s and k_f is the thermal conductivity of the solid and fluid respectively; T_s and T_f the temperatures at the solid and fluid nodes respectively.

$$T_0 = \frac{\frac{\Delta x_{i+1}+\Delta x_{i+2}}{\Delta x_{i+1}\Delta x_{i+2}} k_f T_f^{i+1} - \frac{\Delta x_{i+1}}{(\Delta x_{i+1}+\Delta x_{i+2})\Delta x_{i+2}} k_f T_f^{i+2} + \frac{\Delta x_{i-1}+\Delta x_{i-2}}{\Delta x_{i-1}\Delta x_{i-2}} k_s T_s^{i-1} - \frac{\Delta x_{i-1}}{(\Delta x_{i-1}+\Delta x_{i-2})\Delta x_{i-2}} k_s T_s^{i-2}}{\frac{2\Delta x_{i-1}+\Delta x_{i-2}}{\Delta x_{i-1}(\Delta x_{i-1}+\Delta x_{i-2})} + \frac{2\Delta x_{i+1}+\Delta x_{i+2}}{\Delta x_{i+1}(\Delta x_{i+1}+\Delta x_{i+2})}} \tag{3.43}$$

Fluid-Structure Boundary Conditions

The fluid force imparted onto the solid surface is defined in equation 3.44, where τ_{ij} and P are the fluid shear stress and pressure at the solid surface respectively, P_{ref} is a user specified reference pressure at the solid surface, δ_{ij} is the Kronecker delta, and dA_i is the cell face area.

$$F_i = \int (\tau_{ij} - (P - P_{ref})\delta_{ij}) dA_i \tag{3.44}$$

To apply this boundary condition, the shear stresses and cell face areas- that are stored at cell centers- must be distributed to the fluid side nodes, as shown in figure 3.5. The distributed quantities and nodal pressure are then message passed to the solid where the integral equation 3.44 is evaluated. The fluid force, along with any body forces and the hourglass smoothing force, makeup the right hand side of the solid momentum equation 2.89.

3.1.5 Fluid-Structure Interaction Process Flow Chart

A simplified version of the fluid-structure interaction process is shown in figure 3.6.

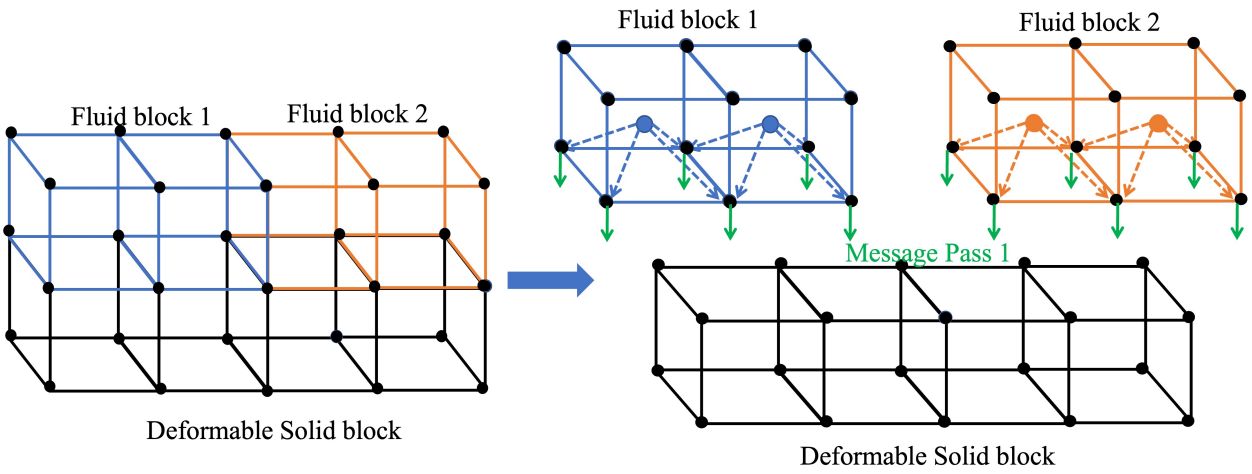


Figure 3.5: Application of Fluid-Solid boundary condition

Solid Solver Solution Process

The structural solver solution process is outlined in figure 3.7. The structural solver may be run independently or with fluid and/or thermal solver depending on user inputs.

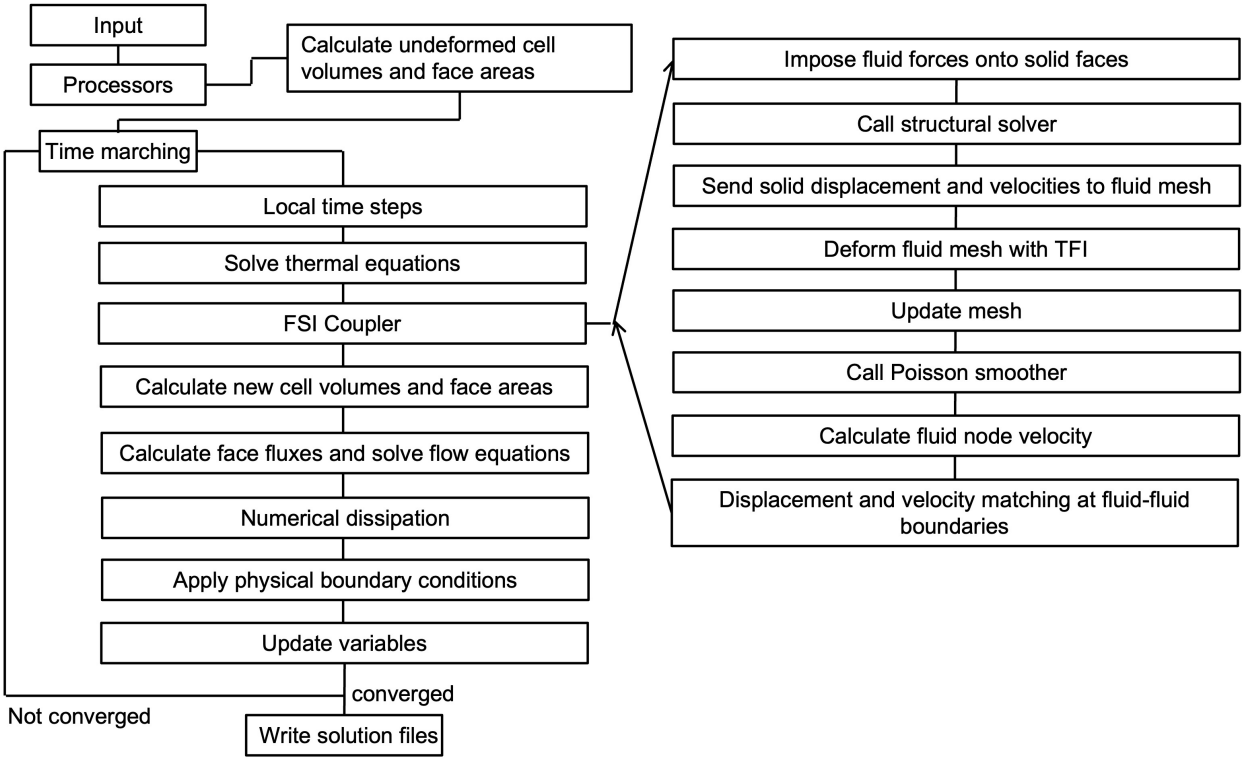


Figure 3.6: Fluid-structure interaction flow chart.

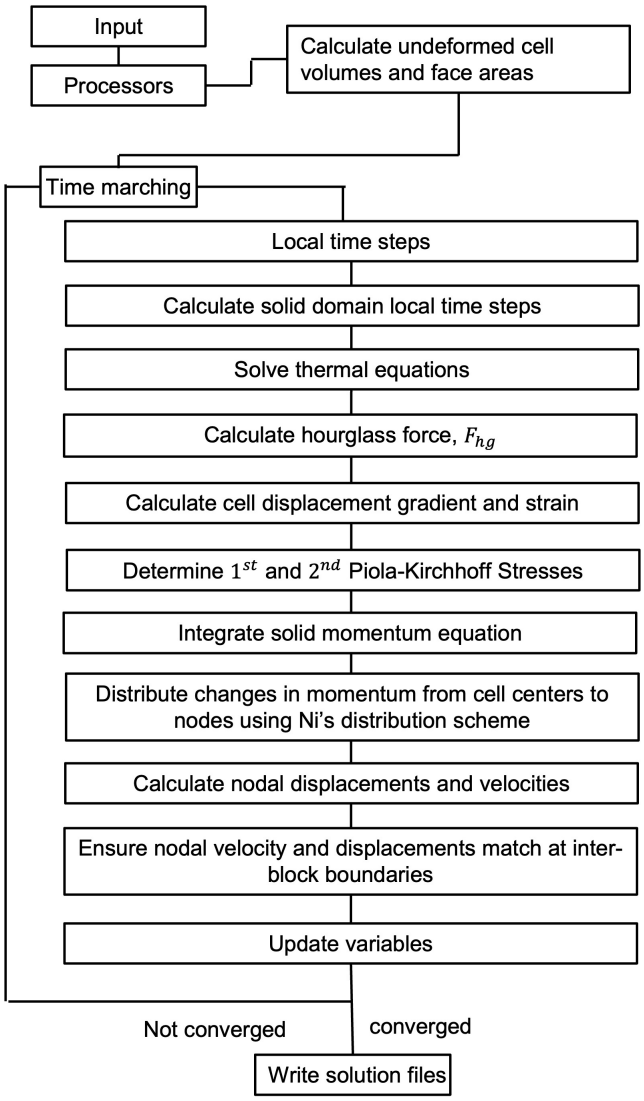


Figure 3.7: Structural solver flow chart

3.1.6 Mesh Motion Techniques

To facilitate a fully coupled fluid-structure interaction simulation, the fluid grid must respond to the solid deformations at every inner iteration. This process consists of six steps: first, deformable solid node displacement is message passed to the adjacent fluid sub-faces, ensuring the grid stays point matched at all fluid-structure boundaries. Second, the fluid

node displacements are distributed through the block. Third, fluid node coordinates are updated by adding the distributed displacement values to the undeformed grid. Fourth, the fluid grid is smoothed via a Poisson smoother to preserve orthogonality. Fifth, the fluid nodal velocity is updated using a second order backward difference given in equation 3.45, where m is the inner iteration index, n is the outer iteration index, and Δt_{Glob} is the global time-step. Finally, the nodal coordinates and velocity are averaged at inter-block boundaries of deformable fluid blocks. This final step is necessary because all deformable fluid blocks are allowed to deform independently. After deformation, all cell face areas and volumes are re-calculated and the fluid solver computes updates to the primary variables. Steps two, four and five merit further discussion.

$$v_n^{m+1} = \frac{3D_n^{m+1} - 4D_n + D_{n-1}}{2\Delta t_{Glob}} \tag{3.45}$$

Distributing Surface Nodal Displacements

Fluid displacements are updated every inner iteration, therefore it is important that the method of displacement distribution is computationally inexpensive. The chosen method for displacement distribution involves the solution of eight algebraic equations. Fluid node displacements are calculated using linear partial sum equations, listed in equations 3.46-3.53. Equations 3.46-3.48 perform linear interpolation along the edges of deformable sub-faces. Equations 3.49-3.51 update displacement on deformable sub-faces the nodal displacements inside of a deformable fluid block are updated using equation 3.52. Finlay the partial sums from each equation are combined using equation 3.53. Displacements are updated on edges, then faces, then the interior nodes are updated.

$$\sigma_\xi(\xi,\phi,\eta) = (1-\xi)D(0,\phi,\eta) + \xi D(1,\phi,\eta) \tag{3.46}$$

$$\sigma_\phi(\xi,\phi,\eta) = (1-\phi)D(\xi,0,\eta) + \phi D(\xi,1,\eta) \tag{3.47}$$

$$\sigma_\eta(\xi,\phi,\eta) = (1-\eta)D(\xi,\phi,0) + \eta D(\xi,\phi,1) \tag{3.48}$$

$$\sigma_{\xi\phi}(\xi,\phi,\eta) = -(1-\xi)(1-\phi)D(0,0,\eta) - \xi(1-\phi)D(1,0,\eta) - (1-\xi)\phi D(0,1,\eta) - \xi\phi D(1,1,\eta) \tag{3.49}$$

$$\sigma_{\xi\eta}(\xi,\phi,\eta) = -(1-\xi)(1-\eta)D(0,\phi,0) - \xi(1-\eta)D(1,\phi,0) - (1-\xi)\eta D(0,\phi,1) - \xi\phi D(1,\phi,1) \tag{3.50}$$

$$\sigma_{\phi\eta}(\xi,\phi,\eta) = -(1-\phi)(1-\eta)D(\xi,0,0) - \phi(1-\eta)D(\xi,1,0) - (1-\phi)\eta D(\xi,0,1) - \phi\eta D(\xi,1,1) \tag{3.51}$$

$$\begin{aligned}\sigma_{\xi\phi\eta}(\xi,\phi,\eta) =& (1-\xi)(1-\phi)(1-\eta)D(0,0,0) + (1-\xi)(1-\phi)\eta D(0,0,1) + (1-\xi)\phi(1-\eta)D(0,1,0) \\ &+ (1-\xi)\phi\eta D(0,1,1) + \xi(1-\phi)(1-\eta)D(1,0,0) + \xi(1-\phi)\eta D(1,0,1) \\ &+ \xi\phi(1-\eta)D(1,1,0) + \xi\phi\eta D(1,1,1)\end{aligned} \tag{3.52}$$

$$D(\xi,\phi,\eta) = \sigma_\xi(\xi,\phi,\eta)+\sigma_\phi(\xi,\phi,\eta)+\sigma_\eta(\xi,\phi,\eta)+\sigma_{\xi\phi}(\xi,\phi,\eta)+\sigma_{\xi\eta}(\xi,\phi,\eta)+\sigma_{\phi\eta}(\xi,\phi,\eta)+\sigma_{\xi\phi\eta}(\xi,\phi,\eta) \tag{3.53}$$

To demonstrate the capability of this simple method, consider a cube of arbitrary dimensions with a two-Dimensional sinusoidal displacement is applied to the top surface while the bottom surface remains fixed. The undeformed mesh is shown in figure 3.8 and the deformed cube is shown in figure 3.9. Upon examination of figure 3.9, one can observe the displacement varies smoothly from the top surface to the bottom surface, but the mesh does loose some orthogonality. There are several limitations to the transfinite interpolation scheme as

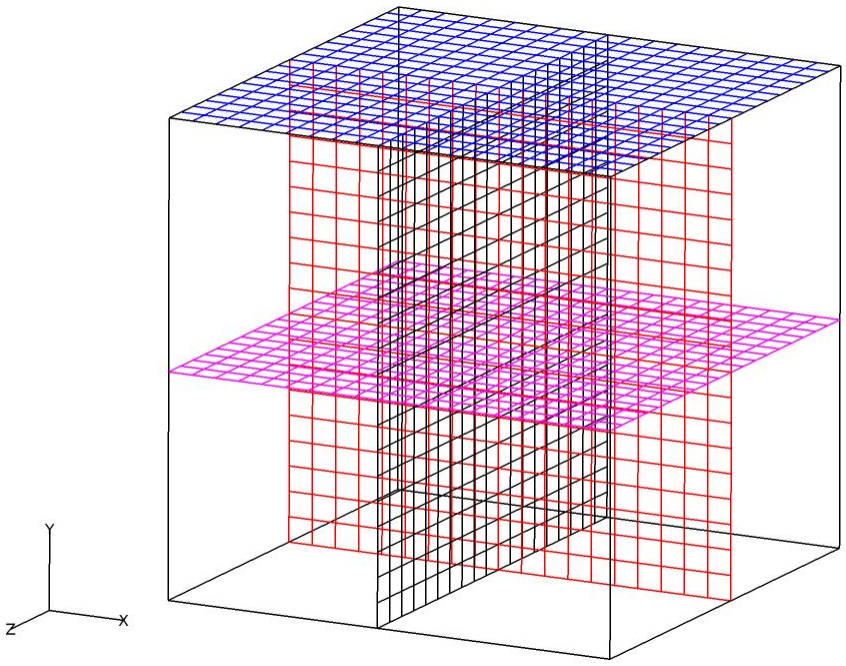


Figure 3.8: Undeformed cube.

it is currently implemented within MBFLO3, the most significant being that mesh orthogonality is lost near surfaces when large displacements occur, which reduces, or destroys, the quality of the solution. For this reason, fluid-structure interaction simulations with large displacements undergo Poisson smoothing (discussed in sub section 3.1.6). Furthermore, mesh deformation does not transfer into neighboring computational blocks which may be required in configurations with large deflections.

This method of displacement distribution is fairly simple to implement in a single block case. However, when parallelism and the multi-block data structure of MBFLO3 is introduced, identifying which edges/sub-edges and faces/sub-faces are deformable becomes algorithmically complex.

Identifying Deformable Edges

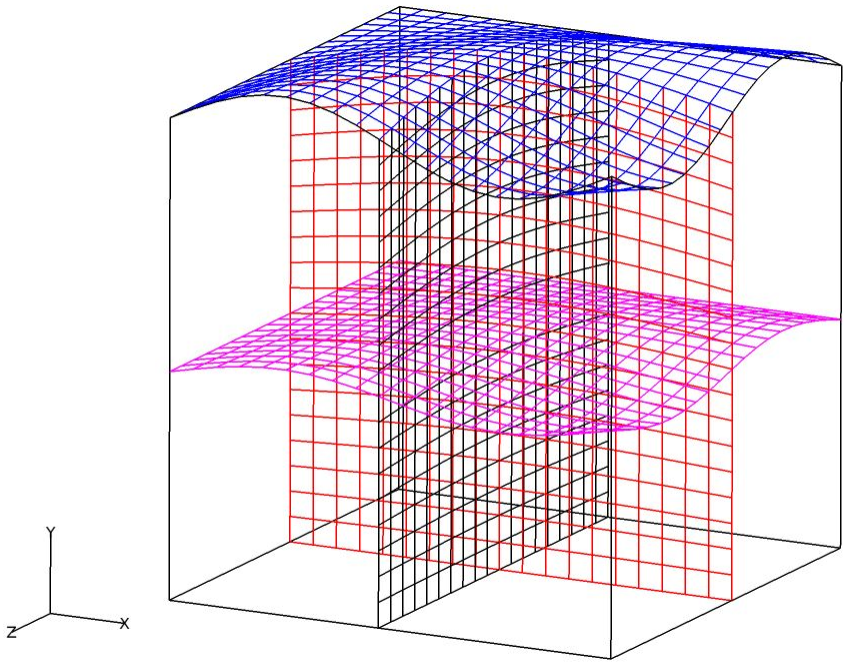


Figure 3.9: Deformed cube.

All boundary conditions applied to a computational block in MBFLO3 are specified in the connectivity file. For a fluid-structure interaction simulation, there are two flags which identify whether and how a block will proceed through the FSI routines. One flag identifies fluid-solid deformable boundaries, the other identifies fluid-fluid deformable boundaries. The displacement of fluid nodes at a fluid-solid deformable boundary is fixed and must be the same as the solid to maintain a point-matched grid. However, the fluid node displacement of edges and faces of fluid blocks normal to a deformable fluid boundary do need to be determined by the displacement distribution technique. The difficulty in implementing the displacement distribution technique is any given deformable fluid-fluid boundary may have an edge which is not allowed to deform via displacement distribution because its nodes belong

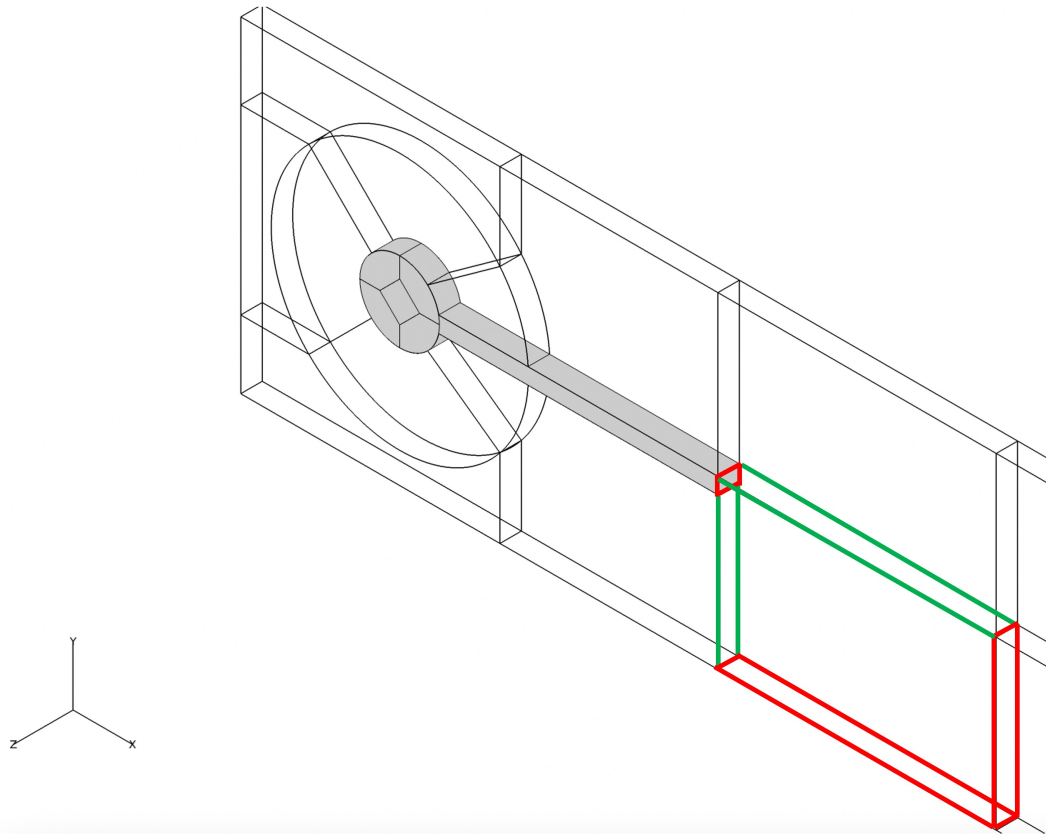


Figure 3.10: Cylinder-flag partial grid outline.

to a fluid-solid boundary or share nodes with a non-deformable fluid block. Figures 3.10 and 3.11 is included to aid in further discussion.

Figure 3.10 shows a partial grid outline with green and red edges denoting deformable and non-deformable block edges respectfully. Figure 3.11 is a close up of the block immediately downstream of the deformable flag with edge numbering included in black and face numbering included in orange. The input information from the connectivity file indicates that face 1 and 4 are fixed while faces 2, 3, 5 and 6 are deformable. Face 3 has a sub-face which has a deformable fluid-solid boundary so no displacement distribution may happen across this sub-face. To determine which edges and sub-edges qualify for displacement distribution a

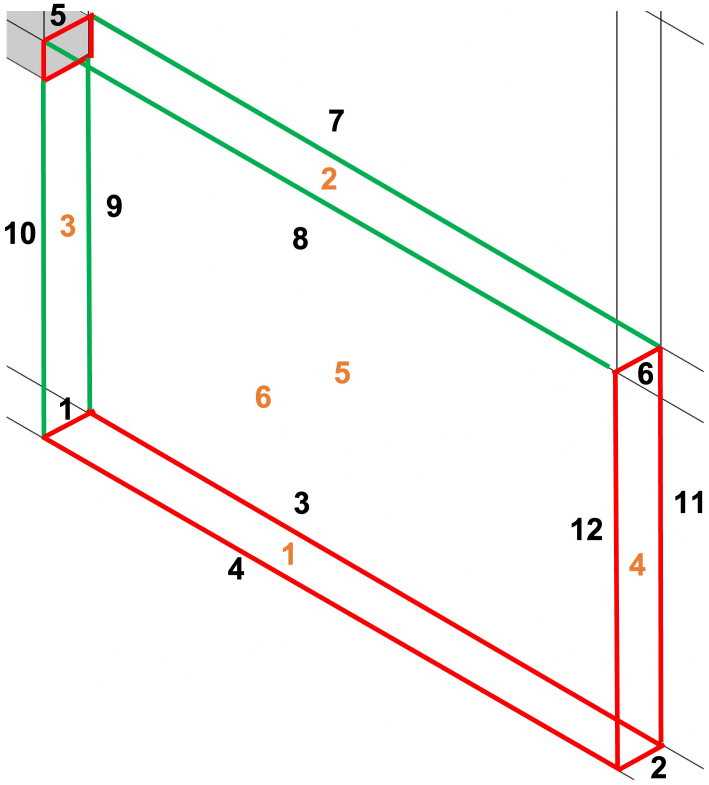


Figure 3.11: Example of deformable faces and edges of a computational block used in the Cylinder-flag test case.

deformable fluid sub-face is chosen. Then, the boundary condition information of each edge and sub-edge which shares a sub-face with the chosen fluid sub-face is checked. If that sub-face is also deformable, the edge or sub-edge will qualify for displacement distribution. This cumbersome process is completed in the pre-processing stage of MBFLO3.

Smoothing Fluid Grid with Poisson Smoothing

Poisson smoothing is conducted to preserve mesh orthogonality, grid spacing, and prevent cells from collapsing. A finite difference based Poisson smoothing technique as proposed by Thompson et. al. [62], Steger and Sorenson [59] and implemented by Davis et. al. [12] is used here. Through this method, each deformable fluid computational block's (x, y) coordinates are mapped to computational space (ξ, η), transformed into referential coordinates, and the following set of Poisson equations are solved, equations 3.54 and 3.55. In MBFLO3, this method smooths (x, y, z) coordinates of computational block by considering only two dimensions at a time. P and Q are forcing functions to maintain user-specified grid clustering near boundaries.

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J(Px_{\xi} + Qx_{\eta}) \tag{3.54}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J(Py_{\xi} + Qy_{\eta}) \tag{3.55}$$

where,

$$\alpha = x_{\eta}^2 + y_{\eta}^2$$

$$\beta = x_{\eta}x_{\xi} + y_{\eta}y_{\xi}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2$$

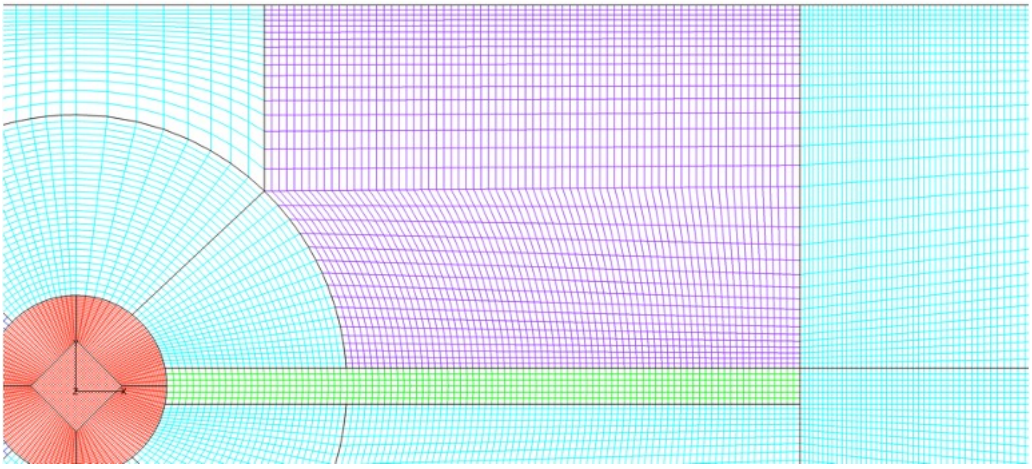


Figure 3.12: Cylinder-flag grid 1 close up.

Since each fluid block is smoothed and allowed to deform independently, it is possible for fluid nodal coordinates at inter-block boundaries to vary significantly. This outcome is undesirable because it reduces grid orthogonality and may reduce solution quality. To correct this issue, the coordinates of two nodes normal to the inter-block boundary are message passed to the neighboring fluid block and incorporated as "ghost nodes". The augmented computational block coordinates are then smoothed with the Poisson smoother. Smoothing across block boundaries can be repeated a user specified number of times to ensure grid continuity at inter-block boundaries. For the current study, smoothing across inter-block boundaries twice per global time step was found to be sufficient. The Poisson smoothing process is demonstrated on the cylinder-flag grid in figures 3.12, 3.13 and 3.14.

3.1.7 Multi-disciplinary Overlaid Grid Treatments

One of the features of this overall procedure is the use of overlaid-grids for the film-cooling holes that allow flow to be transported between the airfoil internal cooling plenums and the mainstream airfoil passage flow. In addition, heat is transferred between the fluid and solid

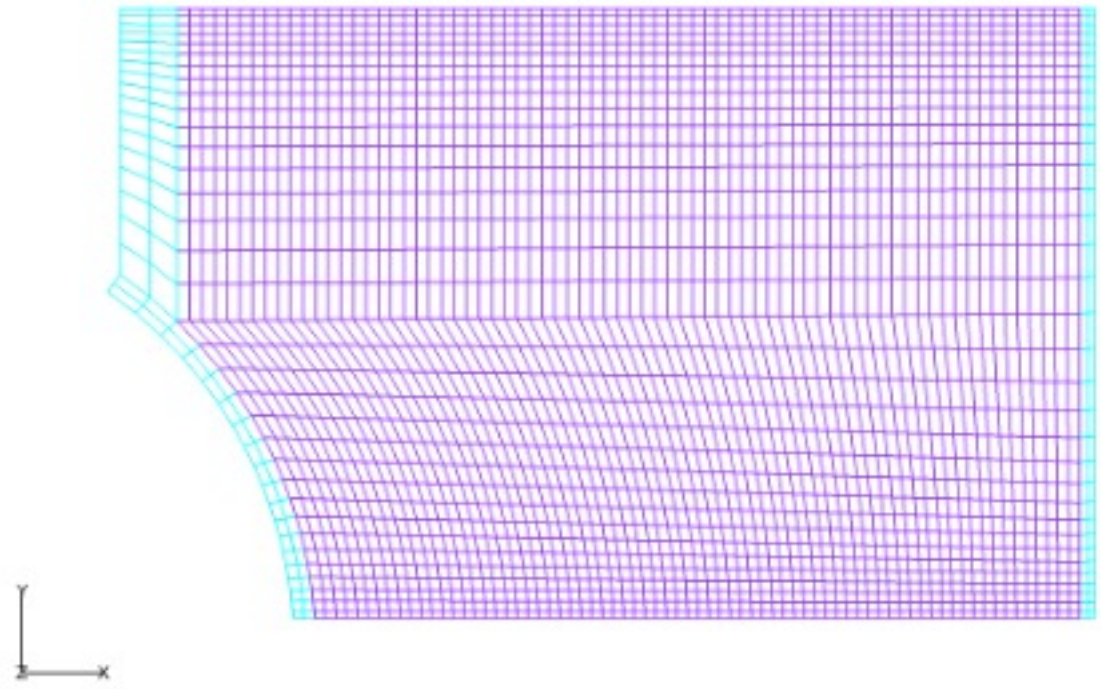


Figure 3.13: Unsmsoothed augmented computational block.

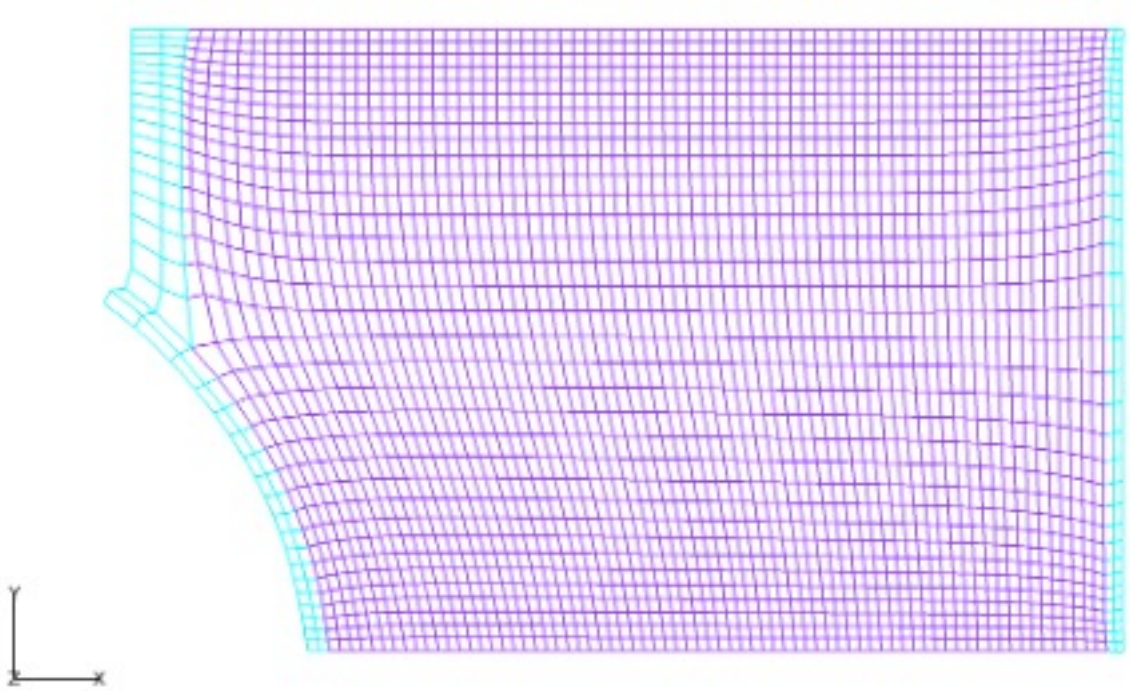


Figure 3.14: Smoothed augmented computational block.

domains. Unique multi-disciplinary overlaid-grid treatments are used to allow information to be transferred to and from the overlaid-grid blocks. An overlaid-grid block (solid or fluid) is a block that does not have a neighboring block or physical boundary condition on at least one of its faces. Overlaid-grid blocks may be FLUID or SOLID.

Two sets of overlaid-grid routines are used in the overlaid-grid treatments. The first set of routines determine if a point within any block is overlaid into another underlying block (with constraints), and if so, determine which cell of that underlying block the overlaid grid point is within. Once this is achieved, the tri-linear interpolation coefficients used to interpolate quantities for the overlaid-grid point may be determined and stored. A multi-step approach is used to determine the location of a point within a cell in computational space (i.e. the interpolation coefficients).

This first set of routines are called in the preprocessing step of the overall MBFLO3 solution procedure and are not required to be called during the time step iteration (unless overlaid-grid blocks move transiently, which is not the case in for the current research effort). This first step is the most time-consuming of the overlaid-grid procedures due to searching for the cell where the overlaid-grid point is within. Steps are taken to reduce searching time through the utilization of bounding-box information. Bounding box information consists of the coordinates of all block and sub-block corners. In the current procedure, every block is subdivided into Nsub sub-blocks by dividing each grid dimension by Ncells. The choice in the value of Ncells is a trade off between efficiency and accuracy. A large value of Ncells can result in sub-blocks being too large and overlaid-grid points being missed during the search due to sub-block curvature. A small value of Ncells can result in an excessive number of sub-blocks leading to higher search computational time and storage. In the current investigation, Ncells was chosen to be 4. The coordinates of the sub-block corners are stored and broadcast to all processors along with the block type (FLUID or SOLID) of the parent block. The search for a cell where an overlaid-grid point lies within begins by finding the sub-block, and thus the primary block, where the point lies within the maximum and minimum of the sub-block corners. This limits the next search step to Ncells 3 of the sub-block. The next step finds the closest point within the sub-block to the overlaid-grid point to reduce the third search step to 8 particular cells of the sub-block. During the third and final step of the search, a coordinate tri-linear shape function and it's first derivative with respect to the three primary directions is used to march from the closest point found in the second step to the overlaid grid point location within one of the 8 cells. For computational grids with fine grid spacing, under-relaxation of this marching technique is useful. Once a unique cell is found where the overlaid-grid point lies within, the block number and grid indices (i, j,

and k) of the overlaid-grid point and the cell that it lies within are added to a linked-list of points.

The definition of the characteristics of an overlaid-grid point shall now be discussed. In the MBFLO3 procedure, two levels of ghost nodes are used at all block boundaries for transfer of inter-block information and numerical smoothing. These point-matched inter-block boundary points and ghost node points do not qualify as an overlaid-grid point. Secondly, to avoid hysteresis effects during the interpolation of overlaid-grid point values, overlaid-grid blocks/points cannot lie within any other overlaid-grid block. In the current version of MBFLO3, overlaid-grid blocks must be embedded into other blocks that are not determined to be overlaid themselves. This constraint is used to avoid the necessity of copying the variables of all blocks into a work array each time-step prior to interpolation and then using the work array (that is unaltered) during for interpolation to avoid any hysteresis. Copying entire-block variables into work arrays is a computationally expensive operation and should be avoided when possible. Thirdly, overlaid-grid blocks are not allowed to be overlaid onto themselves. This constraint is actually a subset of the third constraint but is used explicitly for computational efficiency.

Three overlaid-grid routines of the first set are used to calculate the interpolation coefficients for the eventual transfer of information to and from the overlaid-grid blocks during the second set of overlaid-grid routines. The first of these routines (overlay) finds the cell of a FLUID or SOLID block in which the overlaid-grid point lies within for overlaid-grid block boundary and ghost node points. The second of these routines (overlays) finds the cell of a SOLID block in which an internal overlaid-grid FLUID point lies within. The third of these routines (overlayf) finds the cell of a FLUID block in which an internal overlaid-grid FLUID point lies within. The purpose of these three routines of the first set will be described next in

a discussion regarding the second set of routines that use the interpolation coefficients to efficiently interpolate information for the overlaid-grid points.

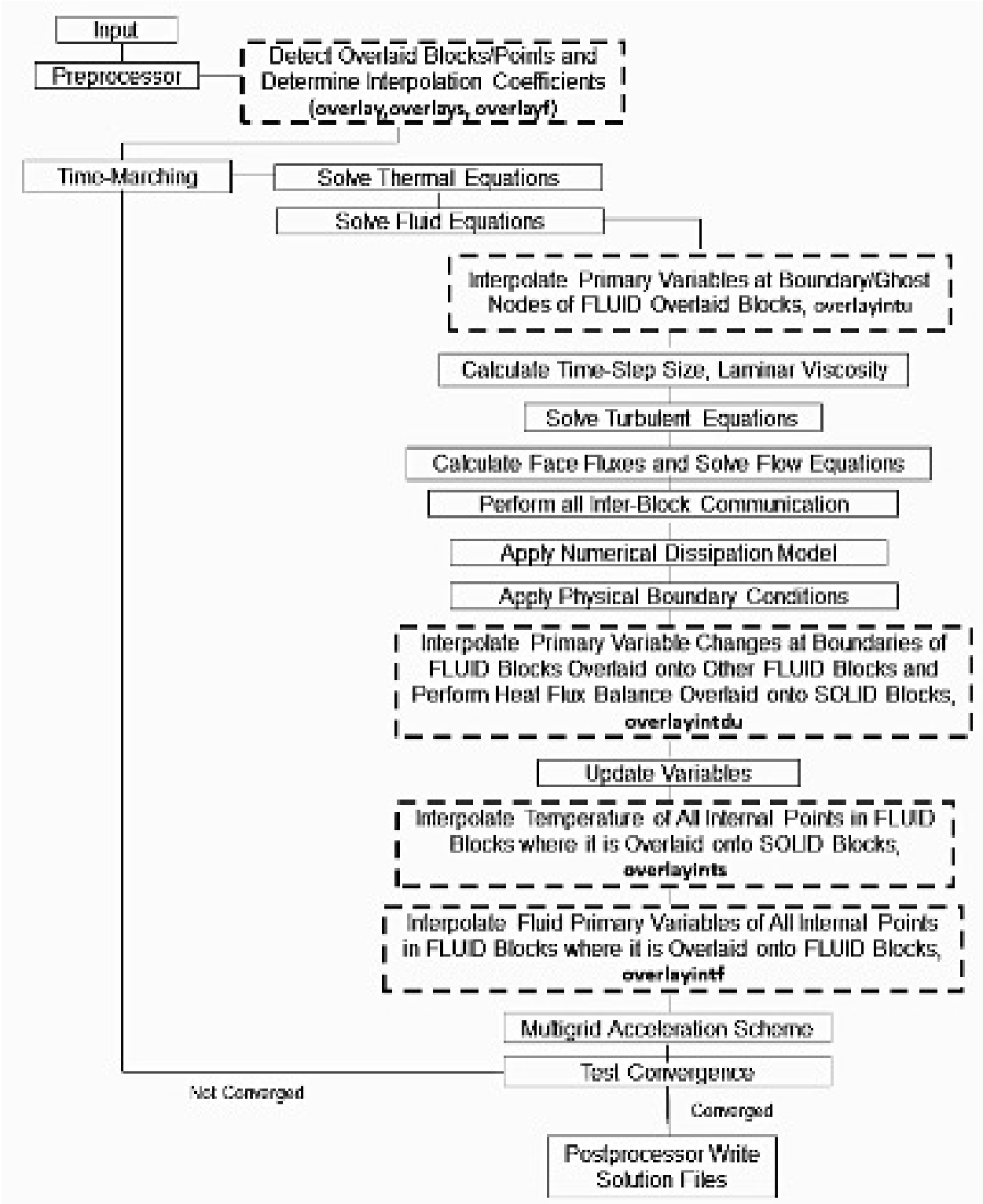


Figure 3.15: Flowchart of MBFLO3 overlaid grid routines.

The second set of overlaid-grid routines is responsible for interpolating the overlaid-grid points defined in the first set of routines from the underlying blocks during each time-step. An outline of the routines used in the MBFLO3 time-marching solver, including this second set of interpolating routines, is given in figure 3.15. There are four overlaid-grid interpolation routines shown in dashed lines. The first routine, overlayintu, shown near the top of the time-marching solver loop uses the overlay routine data structure to interpolate properties at the boundaries/ghost nodes of any FLUID overlaid-grid block. If the underlying block is also FLUID, then all primary variables are interpolated to the boundary and ghost nodes of the overlaid-grid block. If the underlying block is SOLID, then only the temperature of the SOLID block is interpolated. The second routine of the interpolation set is the overlayintdut routine which is performed just prior to updating all primary variables. In the overlayindut routine, the primary variables are interpolated from the underlying overlaid-grid blocks again at the overlaid-grid boundaries and ghost nodes determined from the overlay routine and the time-rate changes in the primary variables are computed in preparation for updating. Again, if the underlying block is also FLUID, then the time-rate changes of all primary variables are computed. If the underlying block is SOLID, then a heat flux balance between the overlaid-grid FLUID block and the underlying SOLID block is performed. The third routine of the interpolation set, overlayints, interpolates the temperature from overlaid-grid FLUID blocks to underlying SOLID blocks. As a result, the temperature of the SOLID block where an overlaid-grid FLUID block exists, is replaced by the FLUID temperature. This treatment corresponds to a temperature feedback to the SOLID block. A similar feedback of fluid primary variables to underlying FLUID blocks is performed by the fourth routine of the interpolation set. In this routine, overlayintf, all primary variables of the overlaid-grid FLUID block to any underlying FLUID block to make the fluid primary variables consistent

between the overlaid-grid block and underlying block.

3.2 Parallel Computing

MBFLO3 is an in-house monolithic, fully coupled, finite volume based, fluid-thermal-structure interaction code. It is written almost entirely in FORTRAN90 and has massively parallel capabilities using Message Passing Interface (MPI), OpenMP, and OpenACC, and is specialized for turbomachinery. MBFLO3 was originally created by Bozinoski and Davis [3] [2]. Various improvements have been made by many contributors including Bates [1], Gottlieb [9] and Lee [42].

3.2.1 Multi-Block Computational Grid Generation

The computational grids for all cases discussed in the results section are generated using case specific utilities, except for the film cooled vane case which uses an advanced in-house automated grid generator named Stage3 [11]. Each computational grid created separates the fluid and solid domains into hexahedral, point-matched computational blocks to improve the speed of computation. Computational blocks can be assigned to any number of processors provided that the number of processors is less than or equal to the number of blocks. If adjacent computational blocks are on different processors the primary variable information is communicated using Message Passing Interface (MPI). Many more details on the generation of each grid will be discussed in chapter 4.

3.2.2 Multi-Block Data Structure

MBFLO3 has been developed to support massively parallel processing. The computational grid is read by the in-house solver and load balanced among available processors. If there are more computational blocks in the grid than available processors, then multiple computational blocks will exist on some processors and a load balancing of the processors is conducted. The load balancing approach aims to load each processor with computational blocks (with associated weights) to achieve a uniform weight distribution across all processors. A computational block's weight is determined by the number of points in the block, number of points along the faces of the block, and the number of equations solved within the block.

Consider the ideal case, where each processor has only one computational block. Figure 3.16 shows an example domain decomposition of a simple circular domain, where block numbers are in red and *NESW* refers to the cardinal directions North, East, South and West. Notice that each block has a North, East, South and West face and each face can be discritized into any number of sub-faces, provided each sub-face contains at least one node. This is best demonstrated in figure 3.16 northern boundary of block 1, where two northern sub-faces are defined and communicate with separate computational blocks. It also becomes clear that the node situated at each corner of block 1 belongs to three blocks. Changes in primary variables at the corner node will depend on the surrounding nodes of each computational block. Ideally, each computational block will have its own assigned processor so it will be necessary to communicate quantities of interest between all three blocks using Message Passing Interface (MPI). The type of data sent between adjacent computational blocks depends on the boundary condition types assigned to a block sub-face.

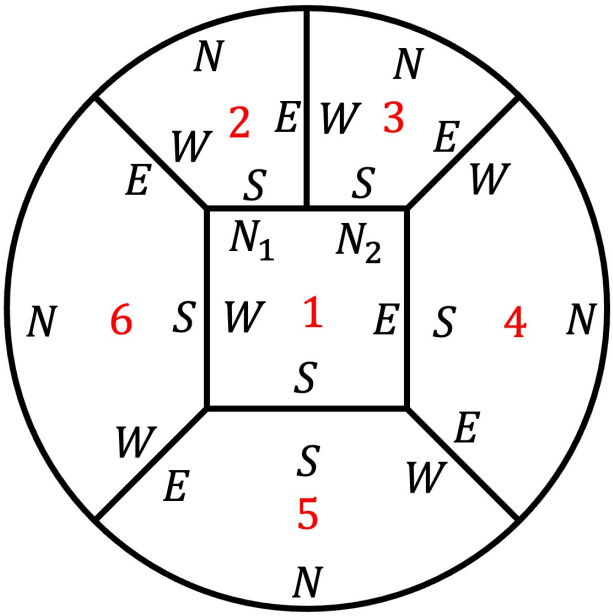


Figure 3.16: Multi-block domain decomposition.

The communication between computational blocks is defined in the connectivity file, designated as the connectivity file. This file is created manually for all cases except for the film cooled vane case because it is an output of the automated grid generator, Stage3 [11]. The connectivity file is fundamentally important to any computation as MBFLO3 has few logic checks to ensure communication between computational blocks happens only at neighboring boundaries.

3.2.3 Overlaid Grids

Overlaid grids occur when two or more grids occupy the same physical space, and are particularly useful in multi-block simulations. Overlaid grids are implemented in MBFLO3 as ghost nodes for adjacent point-matched computational blocks to communicate arrays across inter-block boundaries, for sliding grids in turbomachinery, and as Chimera over-set/embedded grids used for the modeling of film cooling holes.

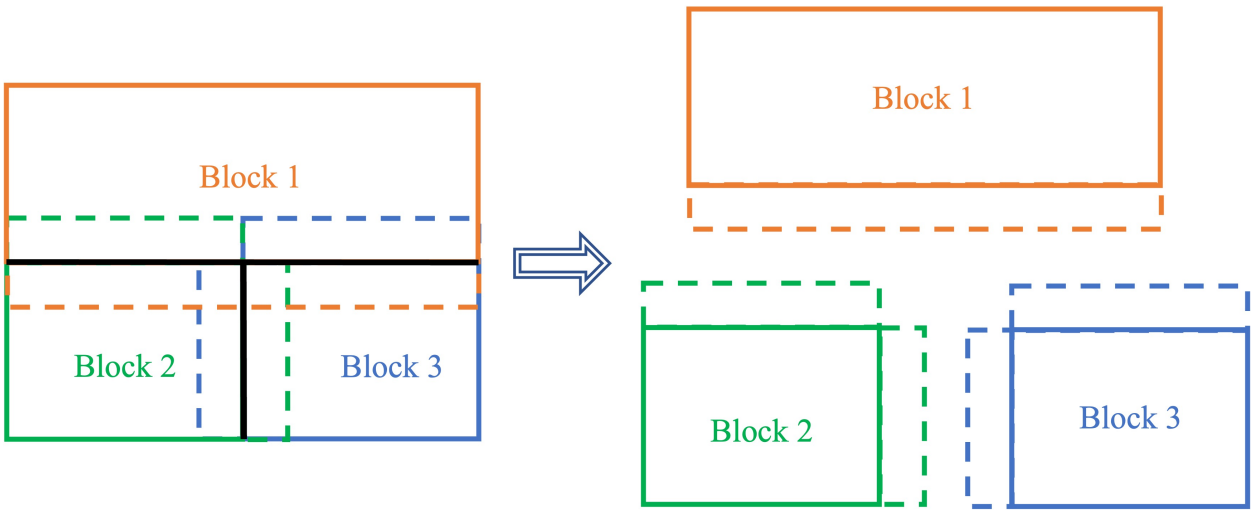


Figure 3.17: Three computational block illustration.

Ghost nodes are used to store and pass information across adjacent point-matched inter-block boundaries. The physical boundary of a computational block is extended into neighboring blocks by adding the first two interior nodes of the neighboring block to the arrays of the current block. This process is demonstrated using a simple three block case shown in figure 3.17, where the dashed boundaries indicate the ghost node boundaries of each block. Figure 3.18 demonstrates the communication overlap of each block. The shared nodes are used to exchange boundary condition information and conserved quantities across inter-block boundaries. All shared nodes will communicate with one another using MPI.

Most turbomachinery computational grids for unsteady simulations consist of rotating (blade) and non-rotating (stator/vane) grids. This complicates the analysis of turbomachinery because the interface between the rotating and non-rotating grids, known as the inter-blade-row boundary, must allow the passage of information upstream and downstream of the boundary. At the inter-blade row boundary the upstream and downstream computational blocks will each create two ghost nodes via linear extrapolation in the axial direction into the neighboro-

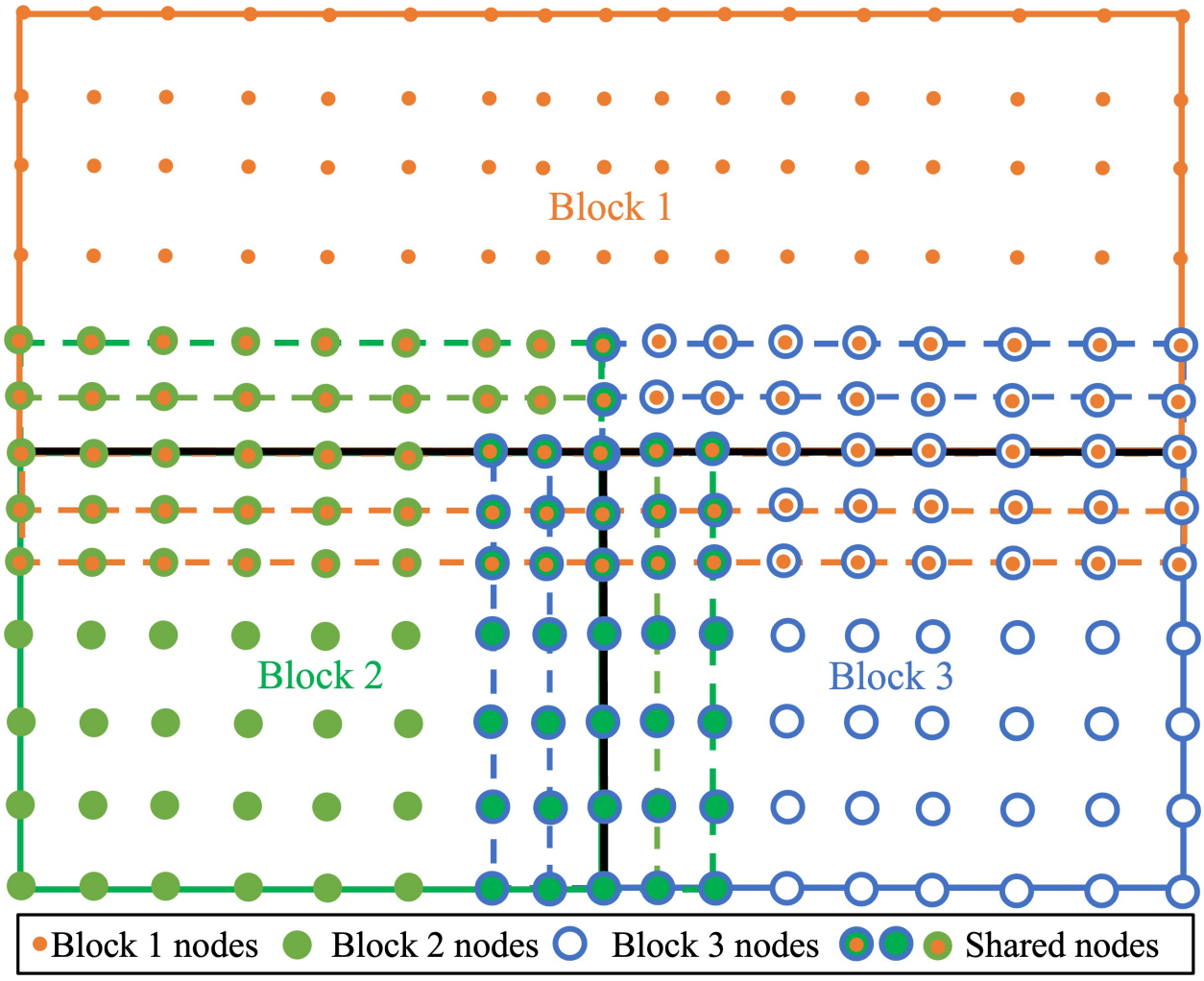


Figure 3.18: Three computational block illustration with nodes shown.

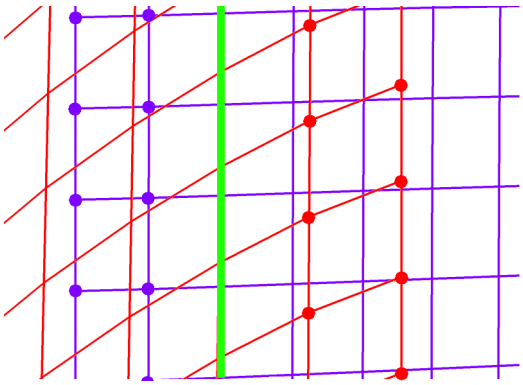


Figure 3.19: Overlaid grid points between blade row boundaries, adapted from Lee [42]

ing block. This process is illustrated in figure 3.19, which is a side view of an inter-blade-row boundary. The red points represent overlaid points that belong to the upstream non-rotating grid, and the purple points represent the overlaid points that belong to the rotating grid. Many more details regarding the implementation of sliding grid can be found in Lee [42].

Overlaid grids are also used to model film cooling holes/tubes in stator and rotor blades of turbines. The cooling tubes, like all other overlaid blocks in MBFLO3, will generate two additional grid points radially and axially along the tube. As a result, cooling tube blocks are overlaid onto both fluid and solid blocks. The cooling-tube blocks are a unique application of overlaid grids, because the primary variables and their time rate changes at all points along the edges are interpolated from the underlying fluid and solid blocks as boundary conditions. Additionally, film cooling blocks usually have a greater grid density than the underlying blocks. As such, the flow solutions inside these blocks are considered to be superior to the solutions of the underlying blocks which contain the cooling-tube blocks. So, the film cooling blocks can be considered as embedded blocks. Many more details regarding the implementation of overset grids to model film cooling holes will be discussed in 3.1.7.

3.2.4 Parallel Processing of Blocks

In MBFLO3, computational blocks are distributed to load-balance the processors based on "weights" that are a function of each block's total number of points, the block number of points on the faces, and the number of equations to be solved in the block (five flow equations, two flow turbulence equations, three structural momentum equations, and one solid heat conduction equation). The FLUID and SOLID computational blocks generated by STAGE3 are stored in a specific order for the purpose of simplifying post-processing. In MBFLO3 after the blocks are distributed to the processors, FLUID and/or SOLID blocks can exist on any given processor. Restart files are generated periodically during the iteration process and at the end of each simulation run. The restart files are generated such that blocks are listed in the original block order as the grid and connectivity files to once again simplify post-processing.

Distributed parallel processing occurs using non-blocking, immediate sends and receives in the Message Passing Interface (MPI) software package. Data at points that lie along a sub-face (or ghost nodes) of a given face of a block exchange information with its neighbor blocks appropriate sub-face/face. All communications between processors occur simultaneously and require a unique communication TAG. The unique tags are dependent on the 1) block number, 2) neighbor block number, 3) face number, 4) neighbor face number, and 5) number of points on the face. Messages do not occur in any particular fixed specific order making TAG creation somewhat difficult. In MBFLO3, generation of the TAG for each message is constructed using combinatorial mathematics that states that a combination of 5 positive, non-zero numbers, A-E, will be unique if $A + B * N + C * M + D * O + E * P$ is unique. For this to happen, $N = max(A)$, $M = max(A) + max(A) * max(B)$, $O = max(A) + max(A) * max(B) + (max(A) + max(A) * max(B)) * max(C)$ and so on. With this approach,

unique TAGs can be generated but unfortunately may be larger than the maximum TAG value allowed by the computer's MPI library. This is remedied by recognizing that the list of TAGs can be renumbered by sorting the TAG list from minimum to maximum and creating a modified TAG list that ranges from 1 to the number of messages. This creation of the TAG list occurs once for each simulation run during the pre-processing step in MBFLO3.

Chapter 4

Results

This chapter will discuss the simulations used to verify and validate portions of the newly developed algorithms. The capabilities added to MBFLO3 can be categorized into four major development projects: the creation of a finite volume based structural solver, the creation of a monolithic FSI method, enhancing the capabilities and order of accuracy of the conjugate heat transfer solver, and the creation of overlaid/over-set grid techniques for the modeling of film cooling holes. The verification of each development effort will be discussed at length in the following sections. The validation of the conjugate heat transfer solver improvements and overlaid/over-set grid techniques is also discussed.

High performance computing clusters, including the University of California, Davis HPC1 Intel cluster and various government clusters including Centennial SGI cluster at the Army Research Laboratory, Mustang SGI cluster at the Air Force Research Laboratory, and Onyx Cray cluster at the Army Engineering Research and Development Center were used during the development, verification and validation of MBFLO3.

4.1 Structural Solver Verification

Two test cases have been chosen to demonstrate the capability of the proposed structural solver. The first is a flexible cantilever beam subject to a gravitational load. This three-dimensional test case is inspired by a two-dimensional test case proposed by Turek and Hron[63], also simulated by Zorn and Davis[72]. The purpose of the first test case is to verify the implementation of the three-dimensional structural solver against literature and theoretical results. The second test case is a highly flexible square cantilever plate subject to a gravitational load. The purpose of this test case is to verify the stability and accuracy of the structural solver with a test case that has arbitrarily large strains. This test case is verified with commercially available software.

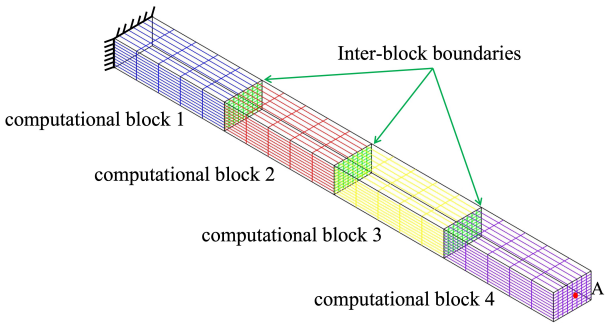
4.1.1 Cantilever Beam

The simulation parameters for a cantilever beam vibrating due to a gravitational load can be found in table 4.1. The computational grid used for this study is shown in figure 4.1a. The left side of the domain is fixed, while all other faces have a reflective boundary condition applied where time-rate changes in nodal acceleration are doubled to prevent span wise variations in nodal displacement and velocity. This boundary condition application enables the comparison of the three-dimensional solver with the two-dimensional results[72],[63]. Zorn and Davis[72] solved the two-dimensional structural momentum equations using a control-volume approach that is described in this work. Turek and Hron[63] employed the finite element method to solve the two-dimensional structural momentum equations.

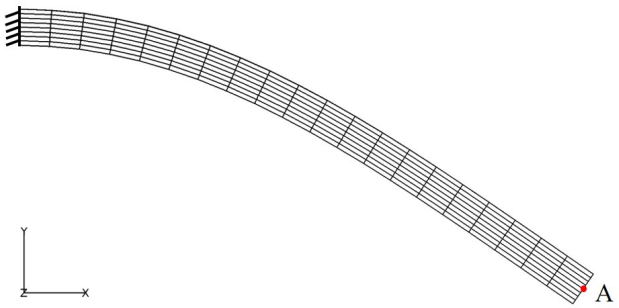
The displacement history of point A in figure 4.1a is tracked throughout the simulation. Uniform, multi-block, hexahedral, structured girds are generated using a simple utility de-

Table 4.1: Simulation parameters for cantilever beam case.

Physical Property	Value
Density (kg/m^3)	1000
Young's Modulus(MPa)	1.4
Poisson Ratio	0.4
Gravitational Acceleration (m/s^2)	2.0
Beam Length (m)	0.35
Beam Height (m)	0.02
Beam Width (m)	0.03



(a) Computational grid for the beam case; point A is used to track the displacement and velocity of the beam throughout the simulation.



(b) Beam grid at maximum deflection.

veloped for this effort. The beam grid shown in figure 4.1a has 1512 grid points and has been decomposed into 4 computational blocks to allow for parallel computing using multiple CPUs. The displacement time histories of the current study, using a time step of $5 \times 10^{-6}(s)$, is compared with the time histories reported by Zorn and Davis[72] and Turek and Hron[72] in figure 4.2. The maximum X and Y deformation of point A is $0.0381m$ and $0.142m$ respectively. Zorn and Davis[72] reported maximum displacement of $0.022m$ and $0.120m$, while Turek and Hron[63] reported $0.029m$ and $0.130m$. The discrepancy between maximum reported displacements by Zorn and Davis[72], Turek and Hron[63] and the current study is caused by the three-dimensional effects of the strain equations.

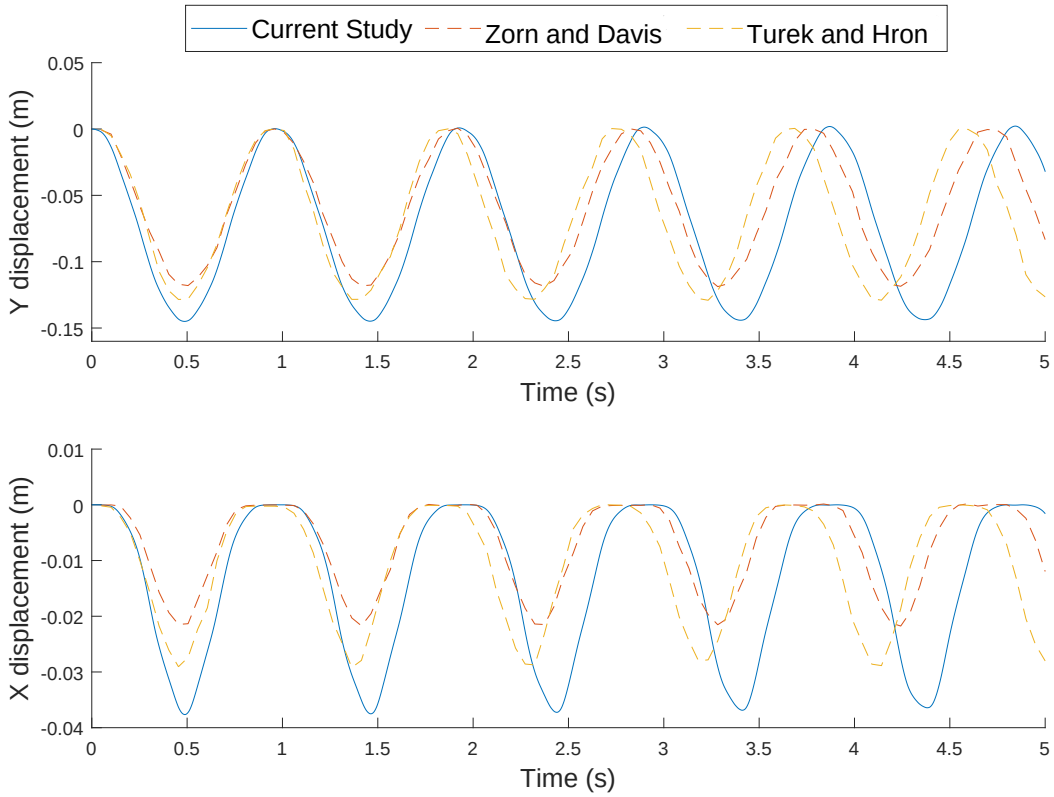


Figure 4.2: 3D Cantilever beam case displacement time histories compared with 2D studies [63],[72].

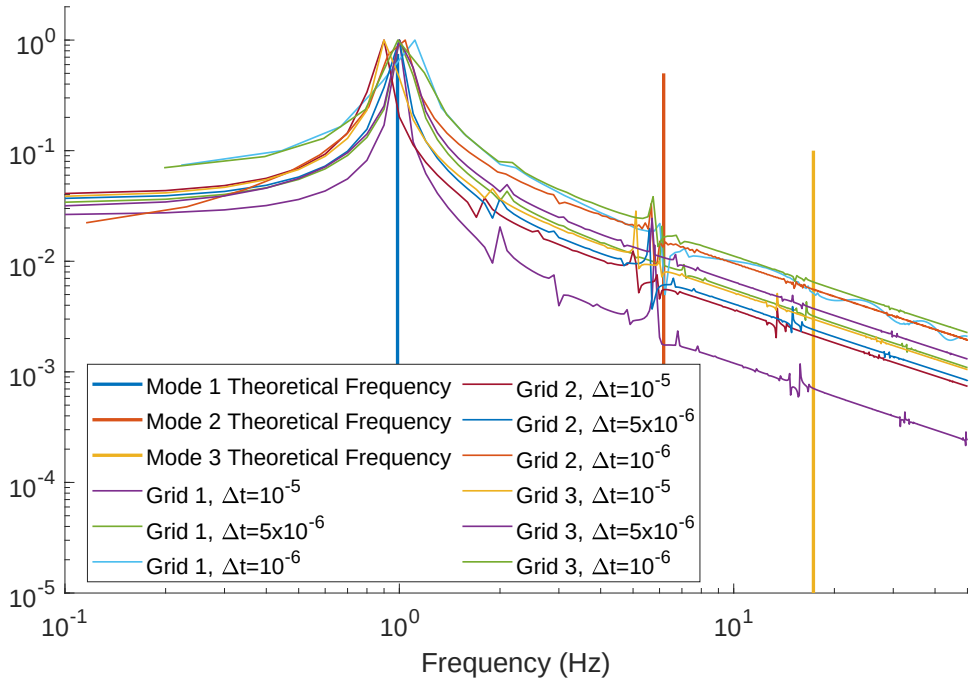


Figure 4.3: Discrete Fourier Transform (DFT) of beam tip Y displacement data.

For further solution verification, displacement time history of point A is compared with the theoretical natural frequency of a cantilever beam subject to a body force, described in equation 4.1[57]; where f_n is the natural frequency of the current mode, K_n is the vibration constant where n refers to the mode of vibration ($K_1 = 3.52$, $K_2 = 22.0$, $K_3 = 61.7$), I is the area moment of inertia and w is the force per unit length applied to the beam. The first three theoretical modal frequencies are 0.988Hz, 6.173Hz, and 17.31Hz. Equation 4.1 is useful to compare the two-dimensional and three-dimensional simulations because the theoretical beam frequency is independent of the depth of the beam, as long as the beam aspect ratio is 8 or greater[57]. The modal frequencies of the current simulation are determined by conducting a Discrete Fourier Transform (DFT) on the Y displacement history of the tracked point. The results of this operation are compared with theory and results reported by Zorn and Davis

Table 4.2: Comparison of theoretical modal beam frequencies and those predicted by current study, Zorn and Davis[72] and Turek and Hron[63].

	Theory	Current Study		Zorn and Davis[72]		Turek and Hron [63]	
Mode	Frequency (Hz)	Frequency (Hz)	% Error	Frequency (Hz)	% Error	Frequency (Hz)	% Error
1	0.989	1.00	1.25	1.05	6.31	1.10	11.37
2	6.17	6.10	1.18	No data		No data	
3	17.31	16.50	4.69	No data		No data	

[72] and Turek and Hron[63] are summarized in table 4.2. The method for solving the structural equations presented in this work predict the first three modes of vibration more accurately than the structural solvers presented Zorn and Davis[72] and Turek and Hron[72].

$$f_n = \frac{K_n}{2\pi}\sqrt{\frac{EIg}{wl^4}} \tag{4.1}$$

A grid and temporal convergence study has been conducted to establish solution independence. Three grids have been created for this study; grid 1 has 1512 grid points-shown in figure 4.1a- grid 2 and 3 have 2376 and 2772 grid points respectively. Each grid has been run with three global time steps, $10^{-5}(s)$, $5 \times 10^{-6}(s)$, and $10^{-6}(s)$. A DFT has been conducted on the Y displacement history data on all of the aforementioned cases and the results are shown in figure 4.3. A grid and time step is considered to be accurate if the calculated modal frequency for the first mode is within $\pm 5\%$ of the theoretical value. All grids meet this criteria for time steps $5 \times 10^{-6}(s)$, and $10^{-6}(s)$. However, only grid 1 meets this criteria for the $10^{-5}(s)$ time step size.

Results of the grid and temporal convergence study are also used to verify that the presented solution method is at least second order accurate. The order of accuracy of a numerical scheme[7] can be determined though the use of equation 4.2; where E is the percent error, C

Table 4.3: Global time step $5 \times 10^{-6}(s)$ simulation results; used to determine the presented method's order of accuracy.

Grid	Cell Volume (m^3)	1^{st} Mode Freq. (Hz)	% error
1	1.906×10^{-7}	0.9995	1.197
2	1.158×10^{-7}	0.998	1.045
3	9.880×10^{-8}	0.997	0.944

is some constant, and P is the formal spatial order of accuracy of a numerical scheme, and Δx is the cell volume. The constant C is only important in determining the absolute level of errors in the order of accuracy. By taking the natural logarithm of equation 4.2, the formal order of accuracy can be determined from a series of solutions with increasingly refined grids using equation 4.3, where P is the slope of $ln(E)$ vs. $ln(\Delta x)$.

$$E = C\Delta x^P \tag{4.2}$$

$$ln(E) = Pln(\Delta x) + ln(C) \tag{4.3}$$

The first mode frequencies predicted by each grid with time step $5 \times 10^{-6}(s)$ can be used to determine the computational error. The order of accuracy of this method is found by plotting $ln(E)$ vs. $ln(\Delta x)$ using the data from table 4.3 and calculating the line of best fit through the data points. With the limited number of grid refinements used in the current investigation, the spatial order of accuracy of this method is found to be 2.816.

To demonstrate benefits of developing a massively parallel structural solver algorithm, a computational speed test has been conducted. The four block computational grid shown figure 4.1a is simulated for 1,000 iterations with four blocks on one processor (serial), and with one block per processor (parallel). The serial case and parallel case had average run

times of 352(s) and 160(s) respectively. Parallel efficiency of the algorithm can be calculated using $\eta = \frac{t_{serial}}{n * t_{parallel}}$; where n is the number of processors. The resulting parallel efficiency for this case is 55%.

To demonstrate the simulation speed increase using the dual time stepping method presented in equation 3.15, consider grid 1 shown in figure 4.1a. The minimum time step for this grid is $8.05 \times 10^{-8}(s)$, using equation 3.17, while an accurate solution has been obtained for this gird with a global time step of $10^{-5}(s)$ with 100 inner iterations. In this instance, the dual time stepping method can deliver speed increases around 24%.

Comparison Solid Smoothing Techniques

Several solid domain smoothing techniques have been developed for this work all with the intention of minimize the formation of hourglass modes. The implementation of these solid domain smoothing methods are discussed in subsection 3.1.3. This section discussed the development of four smoothing techniques: hourglass stiffening, hourglass smoothing, 2^{nd} difference smoothing, and 4^{th} difference smoothing. Hourglass stiffening is applied to all nodes in the solid domain, while the 2^{nd} and 4^{th} smoothing is applied only to solid surfaces.

To verify the implementation of the newly developed 2^{nd} and 4^{th} difference smoothing algorithms, the beam case grid 1 is run with a time step of $10^{-5}(s)$ under a gravitational load of $2m/s^2$. This grid and time step previously demonstrated grid and time step independence as discussed in 4.1.1 and [23]. This simulation is run for $4(s)$ of physical time with an hourglass coefficient, Q_{hg}, of 0.05. The displacement history of the tip of the beam is tracked throughout the simulation. Displacement histories for select cases considered are shown in figure 4.4.

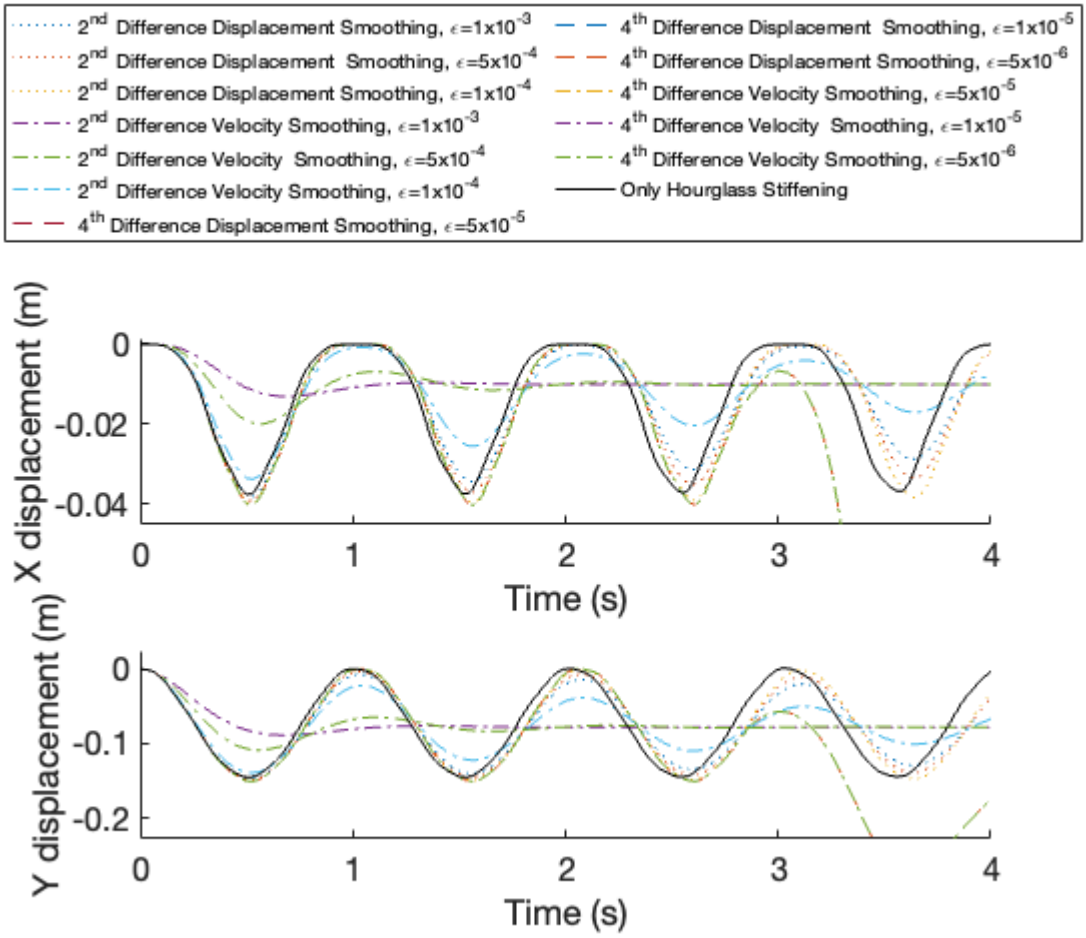


Figure 4.4: Displacement history of beam tip with various smoothing techniques applied.

The modal frequencies of the second difference displacement and velocity based smoothing are determined by conducting a Discrete Fourier Transform (DFT) on the Y displacement

history of the tracked point. The first three modal frequencies are again compared with theoretical frequencies calculated using equation 4.1 in figure 4.5.

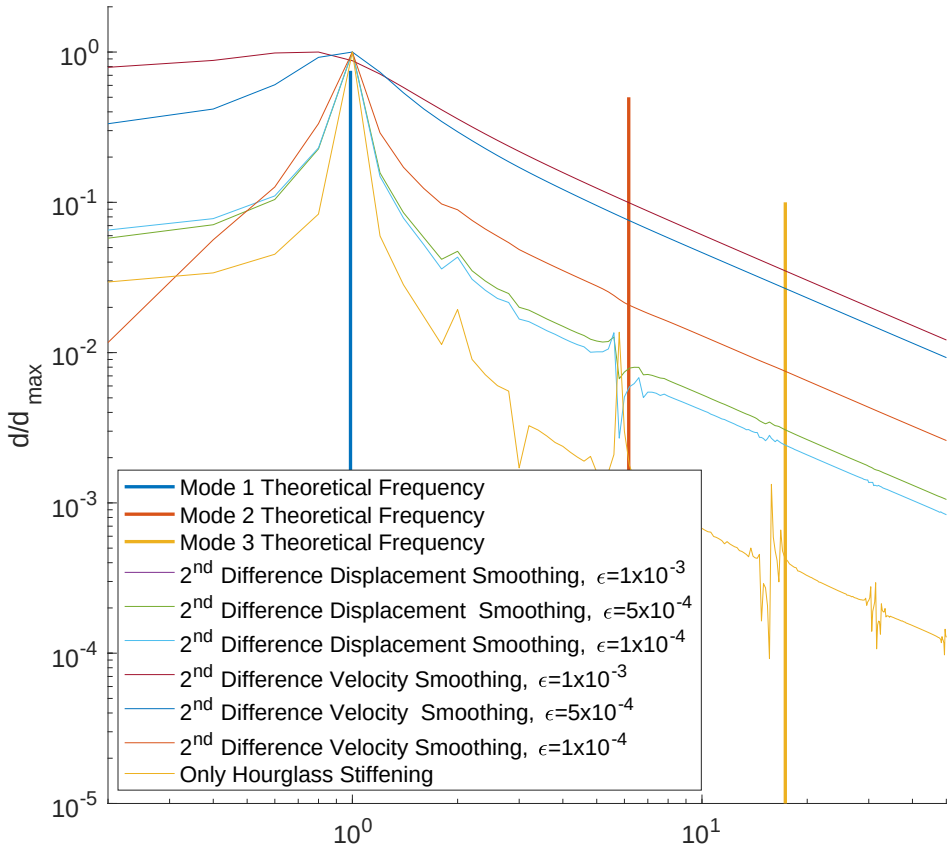


Figure 4.5: Discrete Fourier Transform of displacement time history for 2nd difference cases.

The results of this simulation suggest that 2^{nd} difference smoothing may be used in conjunction with hourglass stiffening without impacting the stability of the simulation. 2^{nd} difference displacement-based smoothing is less dissipative than the velocity-based smoothing and may be a more suitable method. Additionally, the time frequency of vibration and displacement magnitudes were not highly impacted by the 2^{nd} difference displacement smoothing. 4^{th} difference displacement and velocity based smoothing both diverge with the numerous tested values of ϵ, and therefore are not considered viable methods of smoothing for this application.

4.1.2 Cantilever Plate

To further demonstrate the capabilities of this structural solver, a three-dimensional highly flexible plate case is considered. The goal of this case is to demonstrate the versatility of the structural solver, by capturing multiple modes of a highly flexible plate in free vibration. Simulation parameters are identical to those used by the beam case, listed in table 4.1, except the plate width is $0.35m$, and the plate thickness is $0.01m$. The plate is subject to a $2m/s^2$ gravitational downward vertical acceleration. A fixed boundary condition is applied to the left side of the domain and all other faces are free. The displacement of three points- A, B and C in figure 4.6- on the plate tip are tracked throughout the simulation. Uniform, multi-block, hexahedral, structured girds are generated using a simple utility developed for this effort. The plate grid 1 has 3969 grid points, decomposed into a maximum of seven computational blocks. The plate experiences a maximum x and y displacement of $0.297m$ and $0.303m$ respectively. Figure 4.7 shows the plate grid 1 at maximum deflection.

A time history plot for the plate grid with a global time step of $2.5 \times 10^{-6}(s)$ is shown in figure 4.8 **a)** and **d)**. Figures 4.8 **b)**, **c)**, **e)**, and **f)** show detail views of the maxima and minima of displacement history for the first deformation cycle to demonstrate how the displacement varies across the span of the plate.

To verify the plate simulations, a modal model of the plate was created in the commercial code ANSYS DiscoveryAIM. To establish solution independence, two meshes were created using eight node hexahedral elements. The first mesh had 450 nodes, the second mesh had 1152 nodes. A fixed boundary condition was applied to the left face of the plate, all other faces are free. The first three modes calculated with ANSYS for each mesh are $0.526Hz$, $1.230Hz$, $3.179Hz$ and $0.525Hz$, $1.224Hz$, $3.15Hz$ respectively. The modal frequencies

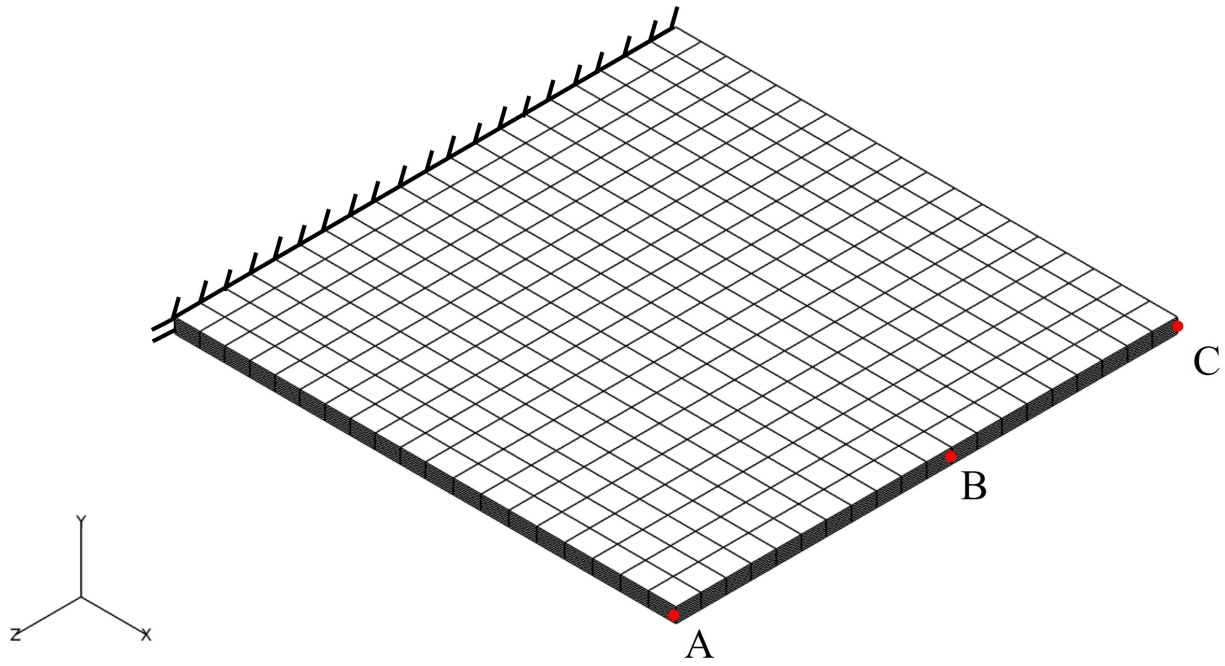


Figure 4.6: Plate computational grid, 3969 grid points. Points A, B, and C are tracked throughout the simulation.

calculated by ANSYS on the second mesh have been used to verify the frequency response of the plate calculated by the current method. The frequency response of the MBFLO3 plate simulation is determined by conducting a DFT on the displacement time history of point A. Table 4.4 compares the results of the plate simulation with 3969 grid points and a time step of $2.5 \times 10^{-6}(s)$ with the 1152 node ANSYS mesh. There is very good agreement between ANSYS and the in-house code presented for the first two modes. The third mode predicted using the current method is $1.71Hz$ and does not agree with ANSYS predictions. Sampling additional nodes may provide a better agreement with ANSYS third mode prediction.

To establish solution independence, a grid and temporal convergence study was conducted on the plate. The first plate grid was run at an additional global time step of $10^{-6}(s)$, and a second grid with 8624 points was created and run at both time steps. A DFT was then

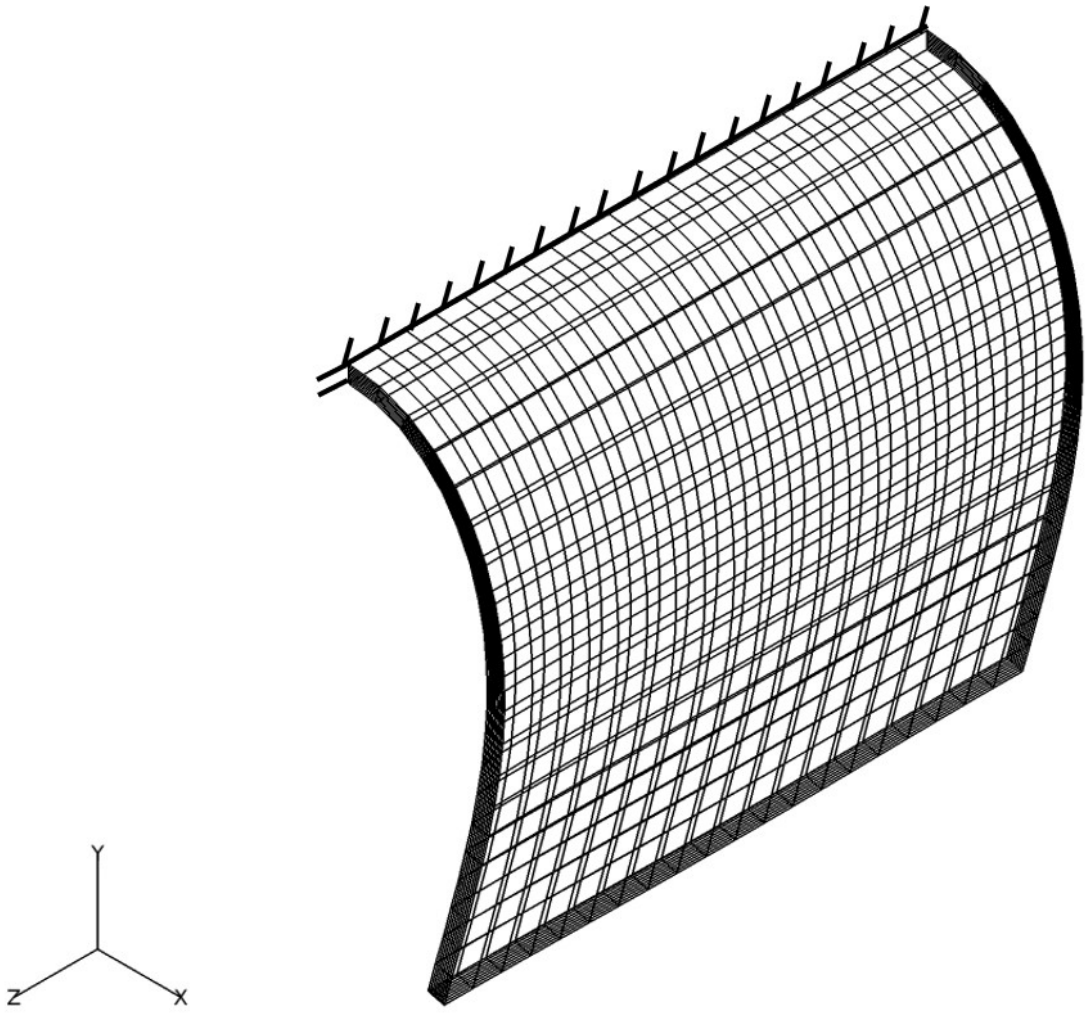


Figure 4.7: Plate grid 1 at maximum deflection.

conducted on the displacement history of point A for each grid. All simulations were run for 7(s) of physical time, which allows the plate to complete three and a half deformation cycles. The results of this grid study are shown in figure 4.9. The first mode frequencies predicted by both meshes and time steps are within $\pm 5\%$ of each other, and demonstrate good agreement with ANSYS frequency predictions satisfying the grid and temporal convergence requirements.

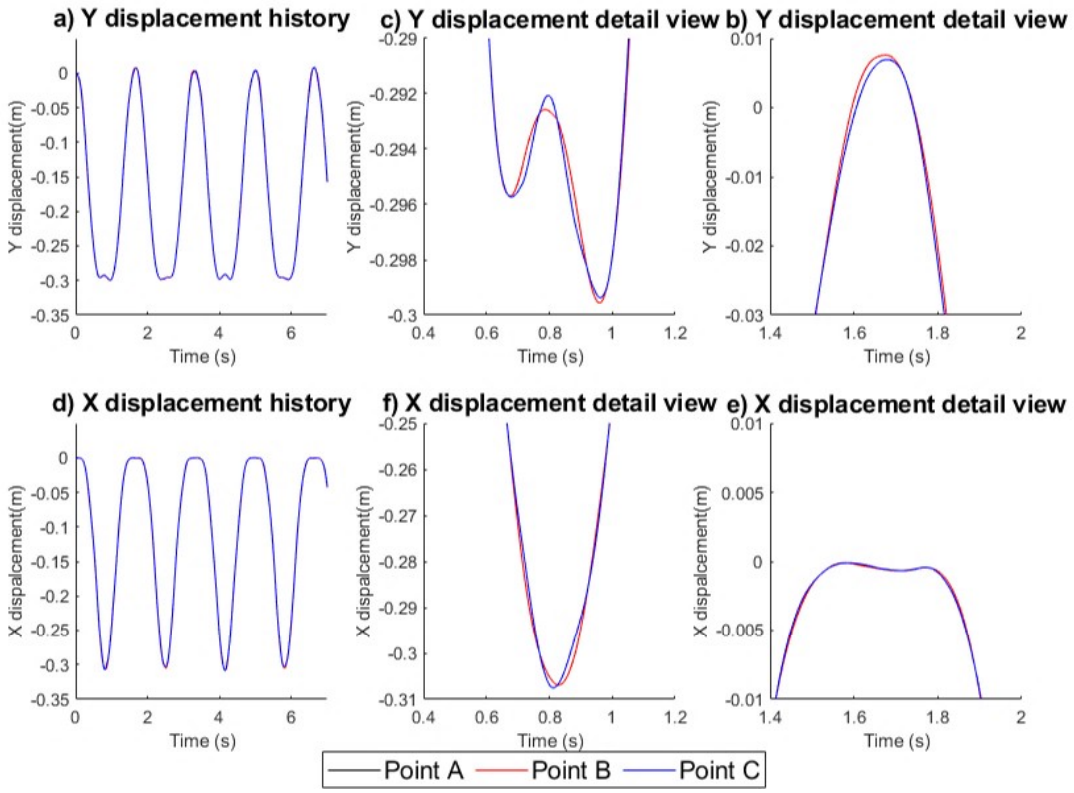


Figure 4.8: Plate simulation displacement history of point A, B, and C for grid 1 $\Delta t = 2.5 \times 10^{-6}$ (s).

To demonstrate the benefits of massively parallel computing, a computational speed test for the cantilever plate case has also been conducted. Plate grid 1, shown in figure 4.6, is decomposed into seven equally sized computational blocks. Then, a simulation is conducted for 1,000 iterations serially (with seven blocks on one processor) and in parallel (with one block per processor). The serial case and parallel case had average run times of 1,222(s) and 221(s) respectively. The resulting parallel efficiency is 79%. This parallel efficiency value is 24% larger than that reported in the beam case because the computational size (overall number of grid points as well as the ratio of block grid points to block face grid points) of the plate case lends itself to parallelism. The number of points per processor for the parallel

Table 4.4: Comparison plate modal frequencies predicted by current study and ANSYS-DiscoveryAIM.

	ANSYS-DiscoveryAIM	Current Study	
Mode	Frequency (Hz)	Frequency (Hz)	% difference
1	0.525	0.571	8.1
2	1.224	1.286	4.8

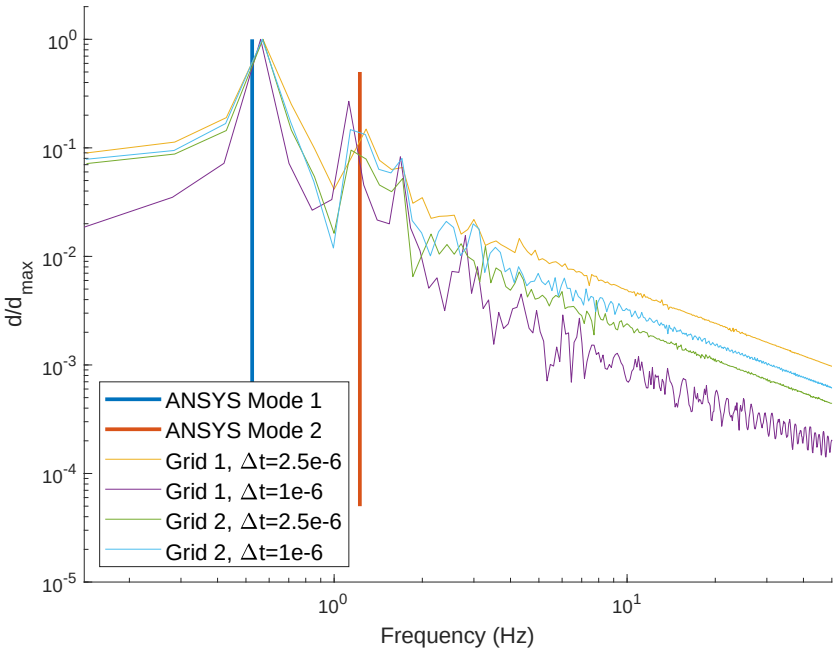


Figure 4.9: Discrete Fourier Transform (DFT) of plate Y displacement time histories of point A.

beam case is 432, while the plate case has 567 points per processor.

4.2 Fluid Structure Interaction

A common fluid structure interaction test case is considered to verify the monolithic fluid-structure interaction code developed for this effort. The Cylinder-Flag test case proposed

by Turek and Hron [63] is a common test case used to verify FSI solvers [72], [27].

4.2.1 Cylinder Flag

The cylinder flag test case has been to used verify the implementation of the fluid-structure interaction solver because it is a benchmark test case used for this purpose. Turek and Hron [63] used a monolithic finite element based approach with an incompressible flow solution procedure. Zorn and Davis [72] used Turek and Hron's [63] cylinder-flag geometry to verify a two-dimensional, monolithic, fully coupled, fluid-structure interaction solver technique with a compressible finite-volume flow solver. The structural response of the deformable flag in this work is compared with structural results presented by Turek and Hron's [63] and Zorn and Davis' [72] predictions to verify the newly developed three-dimensional FSI method.

4.2.2 Grid generation

The computational grids used for the cylinder-flag test case are generated using a utility created for this effort. The basic dimensions of the domains are set by Turek and Hron [63], shown in figure 4.10. The computational domain has a longer total domain length at $3.5m$, instead of $2.5m$ suggested by Turek and Hron [63]. The longer domain was found to be necessary to reduce the effect of pressure wave propagation originating from the outlet by Zorn and Davis [72] who used a similar computational method to the present work. Two different multi-block point matched computational grids with 26 computational blocks are used in this analysis to determine the effects of grid density. Grid 1 has 179,000 grid points and grid 2 has 103,000 grid points. Every-other grid point of grid 1 is shown in figure 4.11.

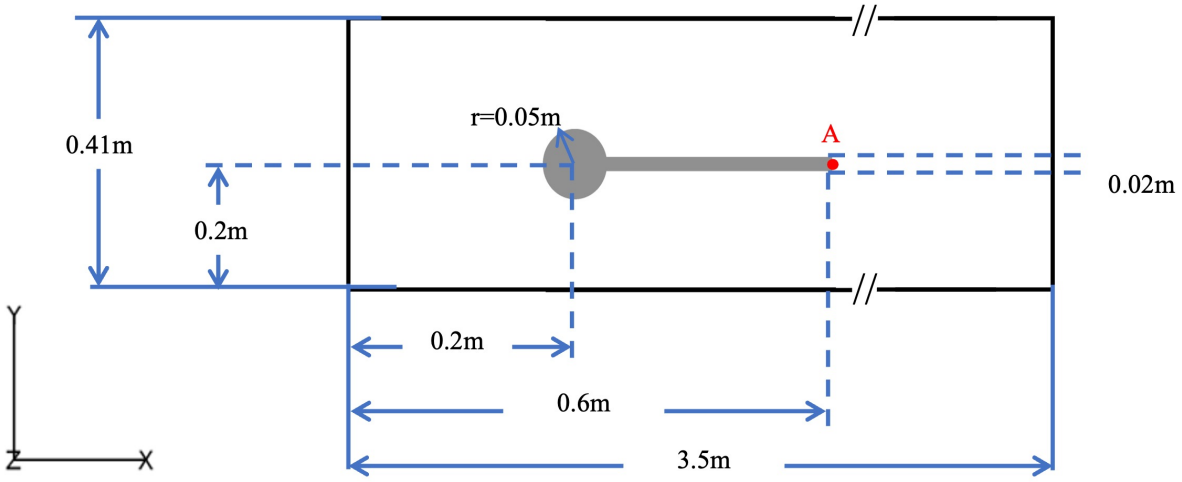


Figure 4.10: Cylinder-flag computational domain dimensions

Initial Conditions and Boundary Conditions

A slighty modified version of the parabolic velocity profile is prescribed at the left channel inflow, subject to equation 4.4, per Turek and Hron [63], where H is the total domain height y is the vertical coordinate, and $\overline{M}$ is the average inlet Mach number. MBFLO3 is a compressible flow solver, unlike the finite element incompressible flow solver used by Turek and Hron [63], therefore the velocity profile is input as a function of Mach number. An average inlet Mach number of 0.1 is chosen to minimize the effects of compressibility. The inlet total temperature is chosen to match the Reynolds number and Strohaul number used in Turek and Hron [63]. The inlet static temperature is calculated by solving equations 4.5, 4.6 and 4.7 simultaneously.

$$M(y) = 1.5\overline{M}\frac{y(H-y)}{(\frac{H}{y})^2} \tag{4.4}$$

$$St = \frac{D\omega}{U} \tag{4.5}$$

$$Re = \frac{\rho DU}{\mu} \tag{4.6}$$

$$U = M\sqrt{\gamma RT} \tag{4.7}$$

To maintain dynamic similarity with Turek and Hron [63], the ratio of fluid density to solid density must be 10.0, which presents a minor difficulty for the compressible flow solver. A fluid density of $1kg/m^3$ is chosen resulting in static and total temperatures that are near absolute zero.

Reynolds Number	100
Strohaul Number	0.2
Vortex shedding frequency, ω	1.9 Hz
Average inlet Mach number,$\overline{M}$	0.1
Average inlet velocity, $\overline{U}$	1.0 m/s
Average inlet total temperature, $\overline{T_t}$	0.250K
Average inlet total pressure, $\overline{P_t}$	71.7 Pa
Fluid density, ρ_f	1 kg/m^3
Solid density, ρ_s	10 kg/m^3
Non-dimensional Young's Modulus, $\frac{E}{\rho_f U^2}$	1400
Poisson's Ratio, ν	0.4

Table 4.5: Cylinder-flag simulation initial values

No slip boundary conditions are specified at the top and bottom surfaces of the channel and all solid surfaces. Front and back faces of the domain are inviscid. The inlet boundary condition holds specified Mach number, pressure and temperature profiles and the outlet uses linearized flow theory. All fluid computational blocks which have a sub-face that contacts the deformable flag will pass pressure and shear stress as loading conditions every inner iteration per equation 3.44.

A color coded cylinder flag grid is shown in figure 4.11 to demonstrate which computational blocks are allowed to deform. Cyan blocks are deformable fluid blocks, the green block is the deformable flag, and all other blocks are stationary. Fluid blocks which have no adjacent

boundary to the flag are allowed to deform to enhance the Poisson smoothing method's the ability to maintain inter-block grid orthogonality.

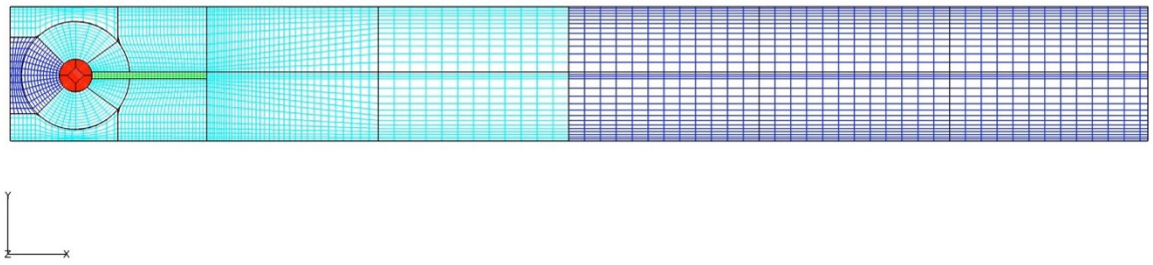


Figure 4.11: Color coded cylinder-flag computational grid, every other point shown; deformable fluid blocks are cyan, fixed fluid blocks are dark blue, rigid cylinder blocks are red, deformable solid flag is green.

Results

The cylinder flag fluid only simulation was run for 1.75 seconds of physical time to allow for a vortex street behind the cylinder to be established. Then the flag was deformed with an initial x and y tip displacement of $-0.008m$ and $-0.06m$. The fluid simulation was then run for 2 seconds of physical time to to allow the fluid flow time to respond to the deflected flag, then FSI simulations began. FSI simulations start with 2^{nd} difference velocity based smoothing in conjunction with hourglass smoothing for the first cycle to prevent solid domain hourglassing. 2^{nd} difference smoothing is initially necessary since there is no time history of nodal velocities or displacements at the start of an FSI simulation with a pre-deformed grid. The lack of time history data causes spurious errors in the dual time-stepping method. After the first cycle, the 2^{nd} difference smoothing parameter, ϵ, is gradually reduced from its initial value of 0.005 to 0.0. Hourglass smoothing is continuously applied with a smoothing parameter Q_{hg} set to 0.15.

The displacement and velocity of the flag tip, at point A shown in figure 4.10 is tracked throughout the simulation. Displacement time histories of x and y displacement are plotted in figure 4.12 and the grid is shown in figure 4.11 with a time step of $0.0001(s)$. The current study's time history is also compared with the tip y displacement history reported by Turek and Hron [63] and Zorn and Davis [72]. The displacement history predicted in the current study demonstrates fair agreement with the literature.

Figure 4.13 and 4.14 shows the Mach number contours at the maximum positive displacement and maximum negative displacement of the flag respectively. In these figures, the capability of the fluid-structure interaction solver to simulate large deformations while maintaining numerical stability is demonstrated.

To gain insight into how effective the implemented mesh deformation techniques are consider figure 4.15, which is a closeup of the deformed grid from figure 4.13. Despite the extreme displacement, the mesh quality remains acceptable. The mesh quality could be further improved, at the expense of computational time, if additional Poisson smoothing iterations were used.

A grid and temporal convergence study has been conducted to establish solution independence. Grid 2 has run with three global time-steps, 5×10^{-4}, 1.25×10^{-4}, and 1×10^{-4} and grid 1 ran with two global time steps 5×10^{-4}, 1.25×10^{-4}. All grids and time steps are run for a minimum of 10 deformation cycles. A Discrete Fourier Transform (DFT) is performed on the displacement history data for each grid and step, the result is shown in figure 4.16. All time steps and grids shown in figure 4.16 are self similar, with first and second mode frequency of $1.73Hz$, and $10.5Hz$. The results also demonstrate good agreement with frequencies predicted by Turek and Hron [63], and Zorn and Davis [72], who predicted a first

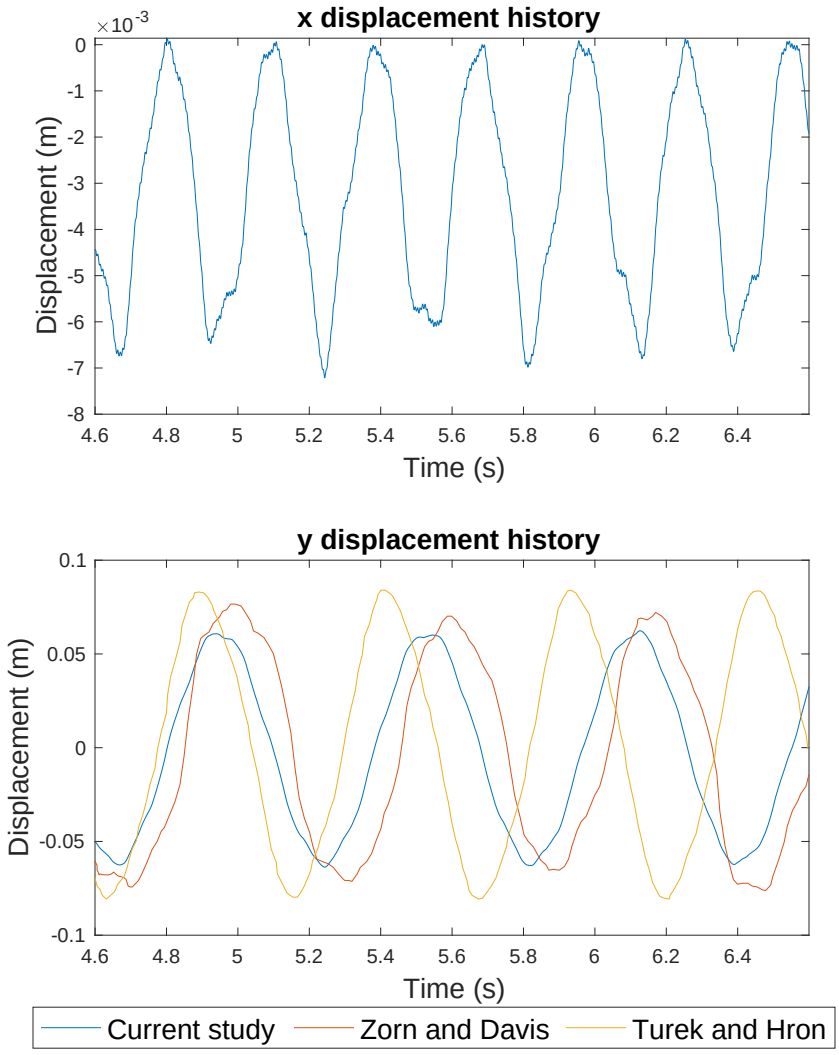


Figure 4.12: Cylinder-flag displacement history comparison with Zorn and Davis [72] and Turek and Hron [63]

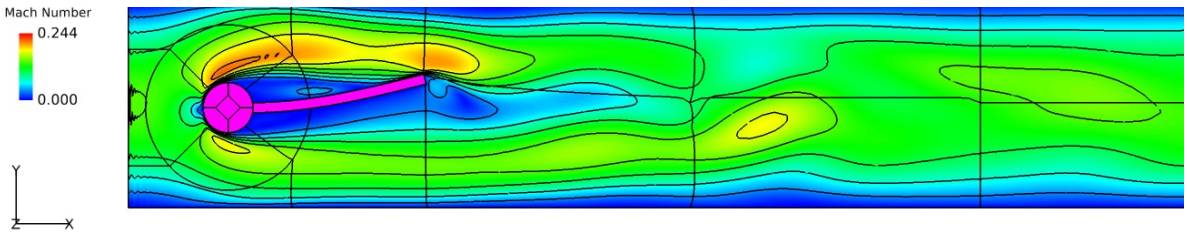


Figure 4.13: Cylinder-flag Mach number contours at maximum displacement.

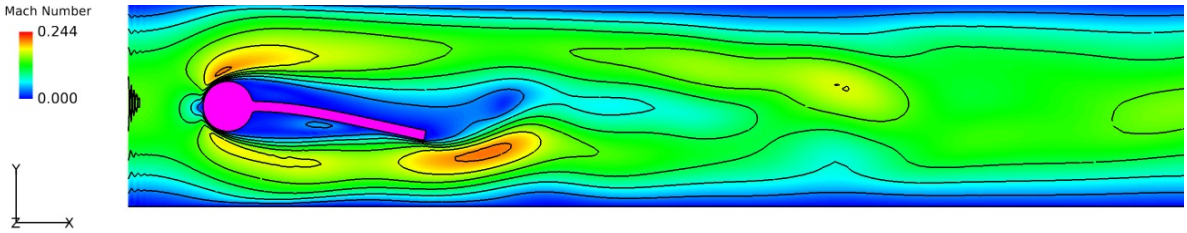


Figure 4.14: Cylinder-flag Mach number contours at minimum displacement.

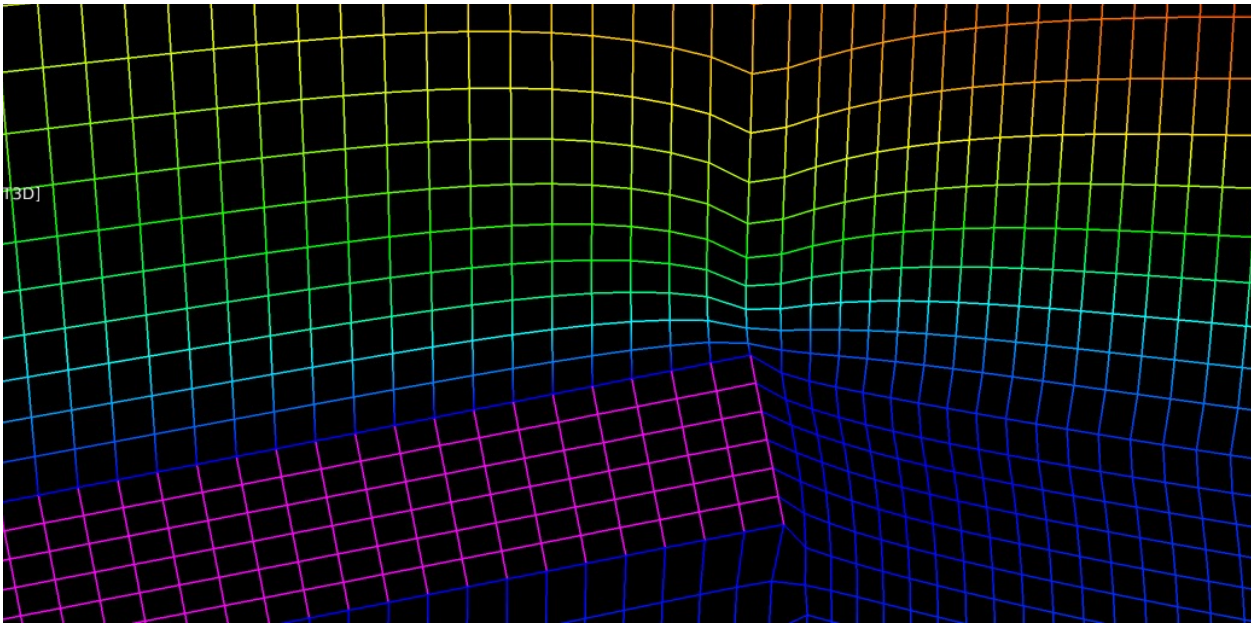


Figure 4.15: Cylinder-flag grid close up at maximum positive displacement.

mode frequency of $2.0Hz$ and $1.7Hz$ respectively.

The FSI method presented in this work demonstrates self similarity with the grid and temporal convergence study. The displacement history frequency is verified with numerical results presented by Turek and Hron [63] and Zorn and Davis [72].

4.3 Heat Transfer Boundary Condition Verification

Two major modifications were made to the heat transfer solver to support this effort, increasing the order of accuracy and enabling solid block-solid block heat transfer. Two simple test cases, turbulent flow over a hot flat plate and a simple two block heat conduction case are considered to verify the implementation of these improvements.

4.3.1 Turbulent Flat Plate

The second order accurate conjugate heat transfer scheme has been added to MBFLO3, and the first order accurate scheme is still maintained. A simple turbulent flow over a hot flat plate case, shown in figure 4.17, has been used to verify the implementation of the new scheme.

The computational grid consists of six blocks, four of which are fluid blocks, shown in blue, and two solid blocks, shown in gray, in figure 4.18. The grid has $436,000$ grid points and was generated using a utility developed specifically for this effort.

The fluid has linearized inflow and outflow boundary conditions, with inviscid walls everywhere except for the solid boundaries, and an initial temperature of 530^oR. The solid has a thermal conductivity of $480\frac{W}{mK}$ material density of $8000\frac{kg}{m^3}$, specific heat of $480\frac{J}{kgK}$, and a

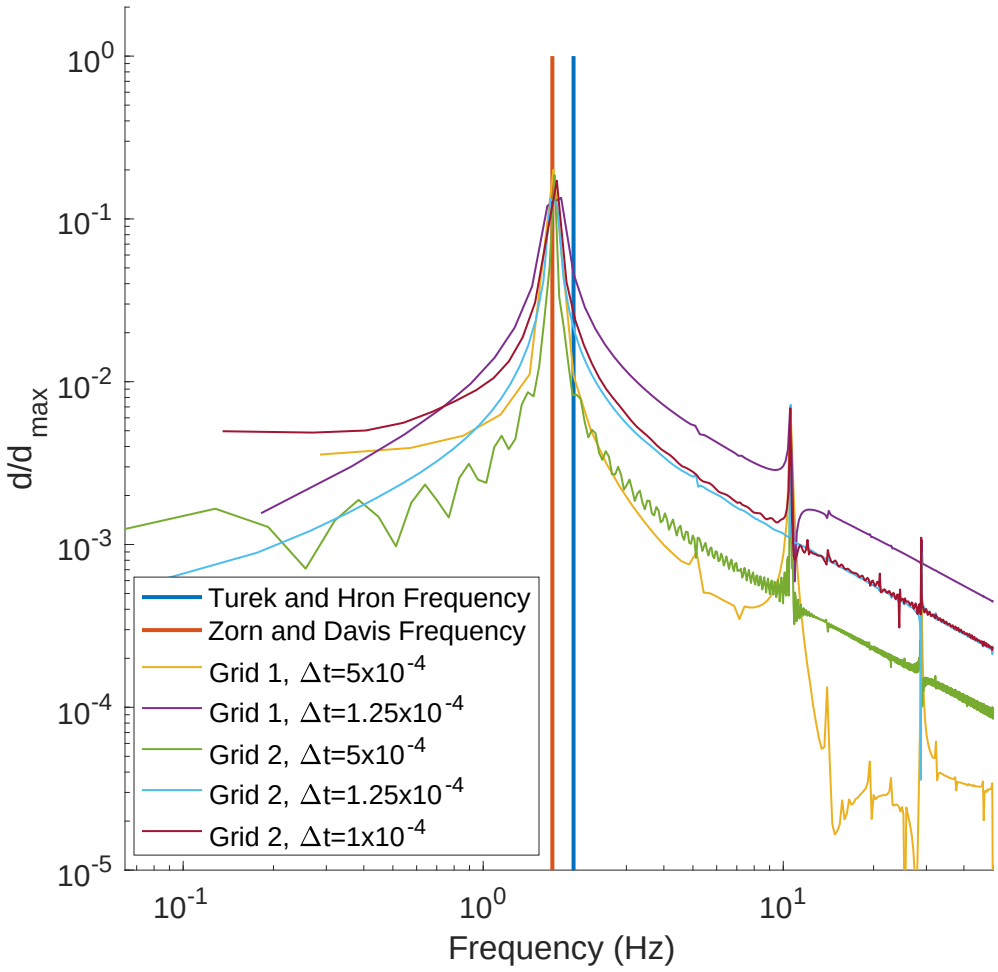


Figure 4.16: DFT of cylinder-flag y displacement history.

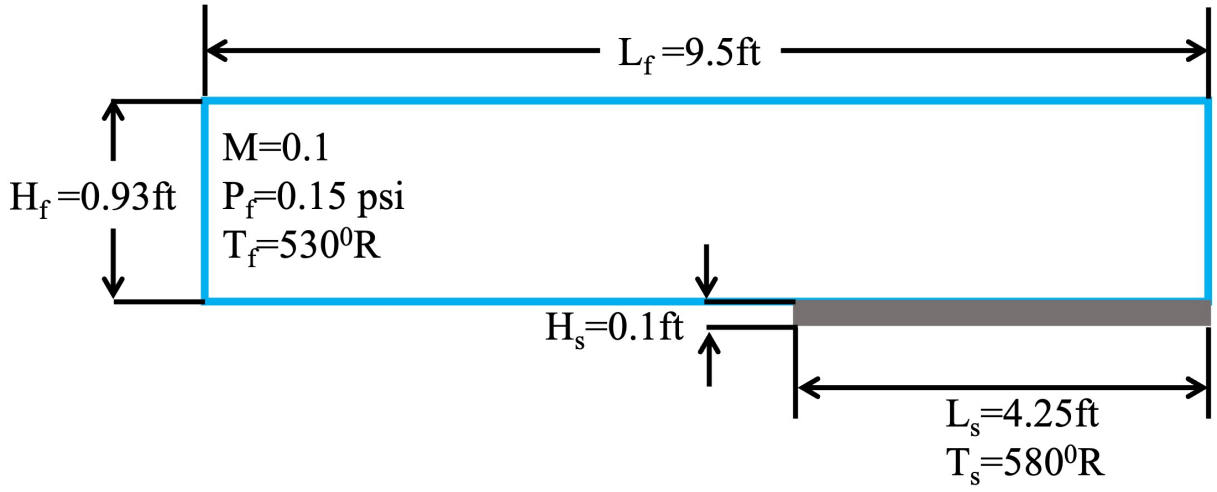


Figure 4.17: Turbulent flat plate case domain and initial values.

constant temperature of 580^oR applied to all faces which are not in contact with the fluid. A conjugate boundary condition is applied at the fluid-solid interface. The mesh near the flat plate is highly refined, and has a $y+$ value of less than 2.0 along the solid boundaries.

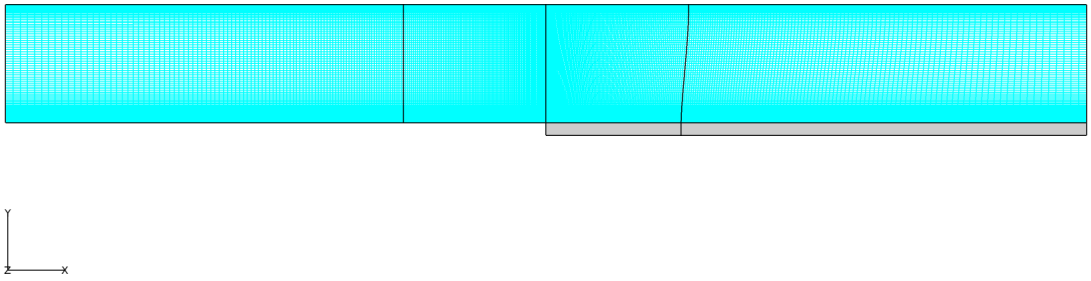


Figure 4.18: Turbulent flat plate grid used for conjugate heat transfer second order improvements

A steady simulation of the turbulent flat plate was run with both the first order and second order accurate conjugate heat transfer schemes. The first order accurate scheme has been previously verified and validated by Lee [42] and Bates [1]. Simulation results from the first order scheme were to verify the implementation of the second order accurate scheme. A comparison of the Temperature profiles near the leading edge of the flat plate, $x = 4.2505ft$, at $0.5s$ physical time, and $85s$ physical time are shown in figure 4.19. The temperature profiles at $0.5s$ physical time are identical, which is expected with this simple case. As the solution reaches convergence at $80s$, the temperature profiles differ in the fluid and solid. The steep temperature gradient near $y = 0.1ft$ in figure 4.19 is the fluid-solid interface, and is expected; only the boundary layer region of the fluid is expected to have a significant temperature gradient. Figure 4.20 shows the temperature profile in the boundary layer. The temperature profiles at $0.5s$ are identical which indicates that the second order accurate scheme has been implemented correctly. At $85s$ the temperature profiles differ by about 4^oR, which may be significant depending on the application.

4.3.2 Solid-Solid Heat Transfer

The addition of solid-solid heat transfer capabilities are verified using a simple two block case. The heat conduction improvements are necessary to capture the heat transfer between the thermal barrier coating and the metal turbine blade for the turbine vane case. This case is discussed in detail in section 4.5. The addition of conduction heat transfer required changes to message passing routines to enable solid-solid heat transfer.

The implementation of the solid-solid heat conduction case is verified using a two block solid configuration, shown in figure 4.21. The computational domain is $1ft$ tall by $1ft$ deep, and $2ft$ wide. The grid has two blocks and 2000 nodes. The left and right solid blocks have an

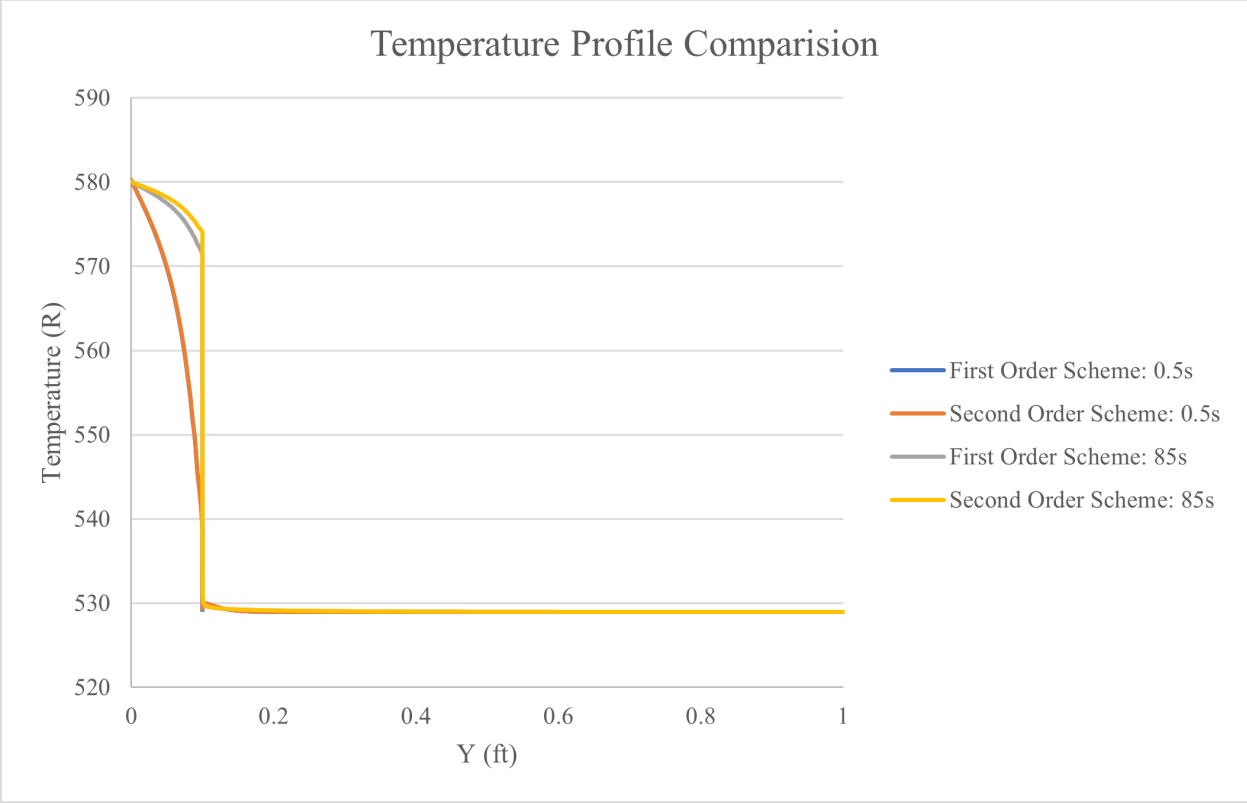


Figure 4.19: Temperature profile comparison near the leading edge of the plate.

initial temperature of 580^oR and 530^oR respectively. The leftmost and rightmost boundaries are held at 580^oR and 530^oR respectively. The solid blocks have a thermal conductivity of $480\frac{W}{mK}$ material density of $8000\frac{kg}{m^3}$, and specific heat of $480\frac{J}{kgK}$.

The steady simulation is allowed to run until the temperature field converges. The result shown in figure 4.21 matches expectations of a linear temperature distribution through the two blocks, $q = -\frac{kA}{\Delta x}(T_2 - T_1)$ [29].

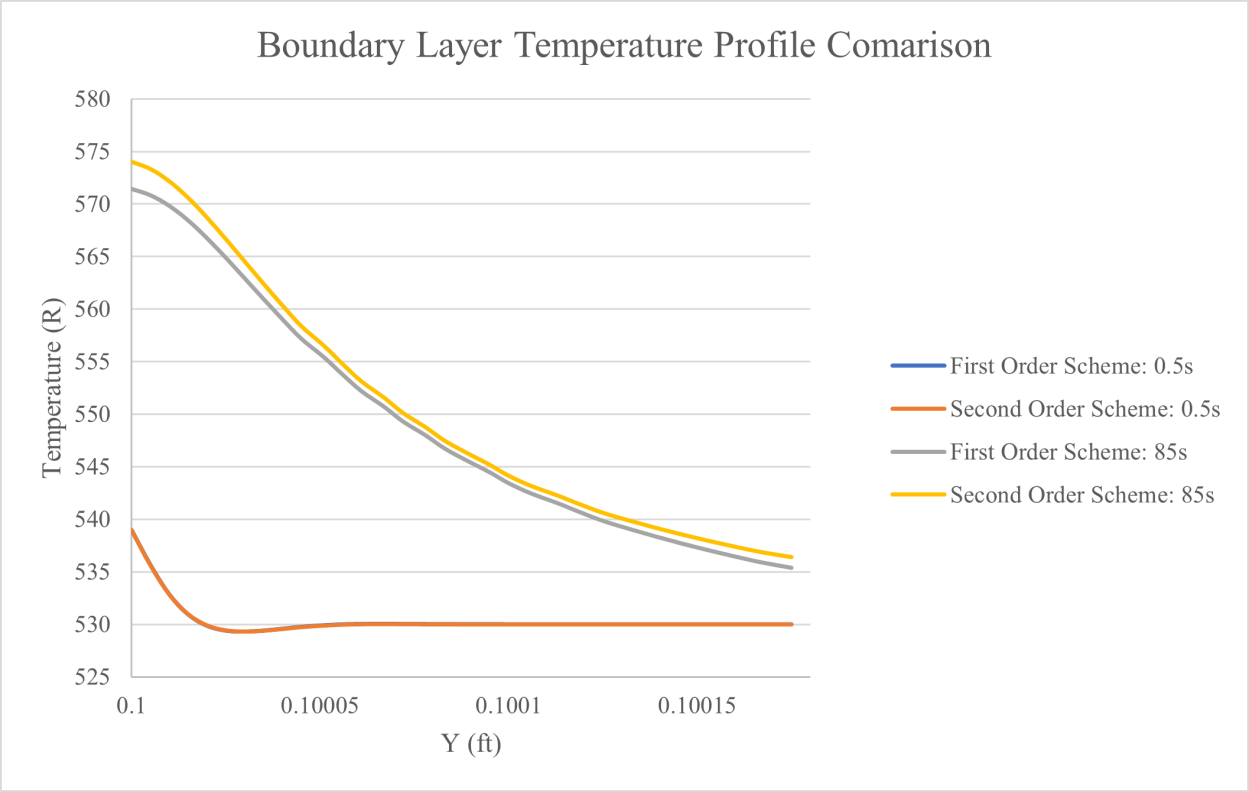


Figure 4.20: Temperature profile comparison in the boundary layer near the leading edge.

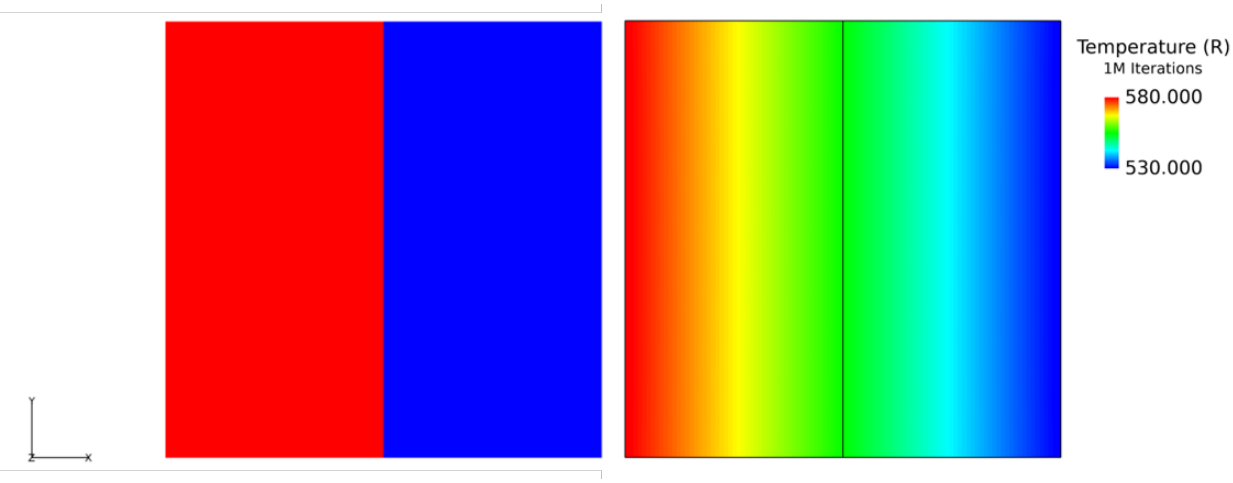


Figure 4.21: two block conduction case, a) initial temperature distribution b) final temperature distribution

4.4 Overlaid Grid Algorithm Verification

Verification of the previously described MBFLO3 overlaid-grid routines, overlayp, overlayintup, overlayintdup, overlaysp, overlayinttp, overlayfp, and overlayintfp used for cooling/blowing holes has been performed using a simple four-block conjugate converging-diverging channel. The configuration and grid, shown in figure 4.22, has been used to verify conjugate heat transfer with overlaid grids. The domain consists of three point-matched cubes (blue and green blocks are fluid with black solid block) have unit lengths and one overlaid grid (red, used to pass fluid from the upstream fluid domain (blue) to the downstream fluid domain (green) through the solid (black) that is two units in length. The grid has $6,000$ grid points. The grid was generated by a utility developed specifically for this test case.

For this case, the inlet Mach number is 0.1, the inlet Reynolds number per unit length is $10,000\ 1/ft$, and the inlet total temperature is 530^oR. The solid block (shown in black) has a fixed temperature of 580^oR on its boundaries. The overlaid-grid (shown in red) essentially "drills" a hole through the solid block. An overlaid-grid conjugate boundary condition, that is second order accurate similar to the standard matched-point conjugate treatment, is applied at the boundary of the overlaid grid where it is embedded in the solid block. Figure 4.23 shows the resulting temperature contours of the solid and fluid regions. Streamwise cutting planes are shown as well as a spanwise cutting plane for the overlaid-grid block. The spanwise cutting plane shows the heat propagating into the overlaid fluid block from the solid block. The streamwise cutting planes show the heat propagating into the upstream and downstream fluid blocks that are point-matched with the solid block. Figure 4.24 shows the Mach number contours in the fluid blocks. A no-slip boundary condition is used on the walls of all fluid blocks including the overlaid-grid block. The Mach number contours show that the flow accelerates through the overlaid-grid block as it migrates downstream which is

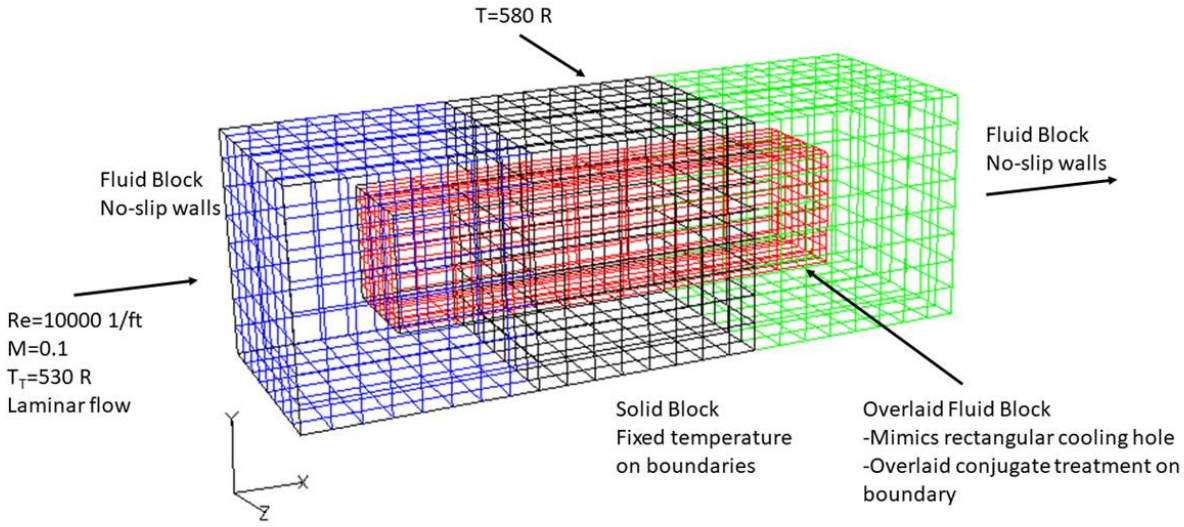


Figure 4.22: Four block conjugate test case grid (blue, green, and red fluid blocks with black solid block).

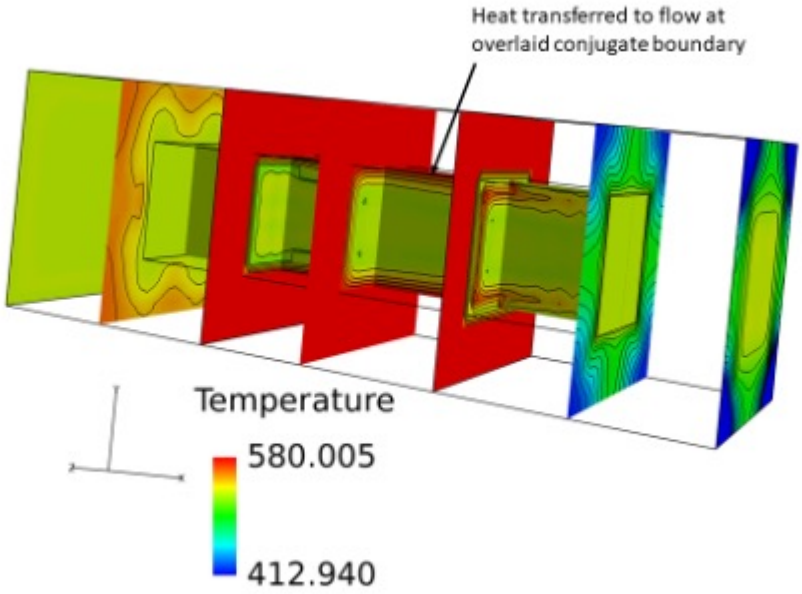


Figure 4.23: Temperature contours in four-block conjugate test case. Heat propagates into the upstream, downstream, and overlaid-grid blocks

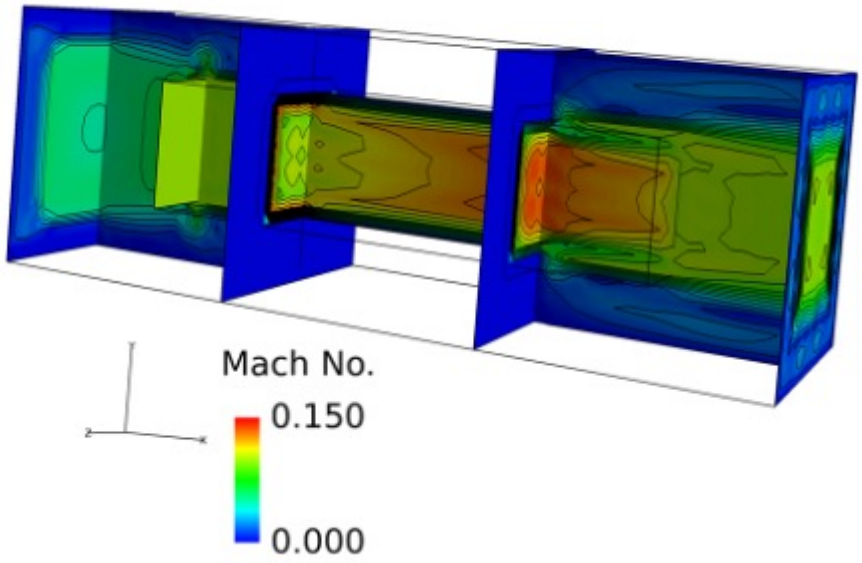


Figure 4.24: Flow Mach number contours in 4-block conjugate test case. Contours show how flow accelerates in cooling hole (overlaid-grid) block.

expected due to the boundary layer blockage and the heat that is added to the flow.

4.5 High Pressure Film Cooled Turbine Vane with Film Cooling Holes and Thermal Barrier Coating

The overlaid grid algorithm and improvements made to the conjugate heat transfer solver have been validated using experimental data by Johnson et. al [35],[36], [37] and Ooten [54]. Comparisons between the experimental data and other predictions can be found in [54],[53],[52],[51]. This test case demonstrates the capability to model extremely complex geometries and flow fields.

4.5.1 Grid Generation

Grid generation for conjugate simulations of film-cooled turbines can be extremely complicated due to the cooling holes. Typical modern high-pressure turbine airfoils have hundreds of holes distributed over the airfoil and endwall surfaces to maintain metal temperature below critical values. For standard convection cooling schemes, each cooling hole cuts through an airfoil wall to allow cooling flow to pass from an airfoil internal plenum to the mainstream flow. The intersection between the cooling hole and the airfoil internal plenum or mainstream is extremely difficult region for grid generation with point-matched computational grids and can result in an abundance of "stripped" clustered-grid regions in the domain that lead to numerical instability or degradation of convergence rate. Some researchers have instead used unstructured grids of different cell shapes to avoid topological issues related to matching grids at the cooling-hole/plenum or cooling-hole/mainstream intersection regions. Although this approach can provide grids that blend smoothly between the cooling-holes and plenums or mainstream, the techniques used to generate the grids can often require significant human-hours of labor and force the use of unstructured-grid conjugate solution procedures that can be computationally expensive. An alternate approach to film-cooled airfoil grid generation is the utilization of overlaid-grids. As previously mentioned, this approach is utilized in the current research. The use of overlaid-grids for this application removes the point-matching and topological issues at the interface between the cooling-holes and the plenum or mainstream. However, an additional cost related to communication/interaction between the overlaid-grids in these regions is added to the solution procedure.

The computational grid used for the current investigation is generated using the STAGE3 [11] multi-block, structured-grid generation procedure. In this approach, the computational grid is generated directly from the design-intent information tree. In some grid generation

Feature	Sub-Block	Type	Number	I-dim	J-dim	K-dim
Mainstream:	Airfoil O-grids	Fluid	4	161	49	441
	Leading H-grid	Fluid	1	129	17	441
	Trailing H-grid	Fluid	1	97	17	441
	Mid-passage H-grids	Fluid	8	129 (2), 153 (2), 157 (2), 97 (2)	65	441
Cooling Plenums:	O-grids	Fluid	2	357, 309	17	441
	H-grids	Fluid	2	133, 138	47, 18	441
Airfoil Cooling Holes:	O-grids	Fluid	566	33	33	13
	H-grids	Fluid	566	33	9	9
Airfoil Thermal Barrier Coating (Kapton)	O-grid	Solid	1	641	5	441
Cooling Plenum Walls:	O-grid	Solid	2	357, 209	9	441
Trailing Edge Wall	H-grid	Solid	1	35	18	441

Table 4.6: Grid block decomposition

processes, the design-intent information is input to a computer-automated-design (CAD) package to generate splined surfaces with the desired features. Unfortunately, once the splined surfaces are created, the topology of the geometrical entities is often lost making grid generation on those surfaces and the volumes within quite difficult. Surface grid cells (usually unstructured) can be generated on those surfaces more easily but the volume grids within the surfaces is difficult. In the STAGE3 grid generation system [11], the design-intent information along with the known topology of each feature is transformed into a computational grid directly. When grid blocks are not easily connected with point-matched boundaries, overlaid-grids are used. Grid blocks are algebraically generated and smoothed using a Poisson equation to make the grid normal to boundaries when possible. The STAGE3 system also automatically generates the block connectivity information used in the parallel processing of blocks on multiple computer processors.

Figure 4.25 shows the computational grid for a film-cooled axial turbine vane. The computational grid for this particular example consists of 1154 blocks with 68.9 million grid points.

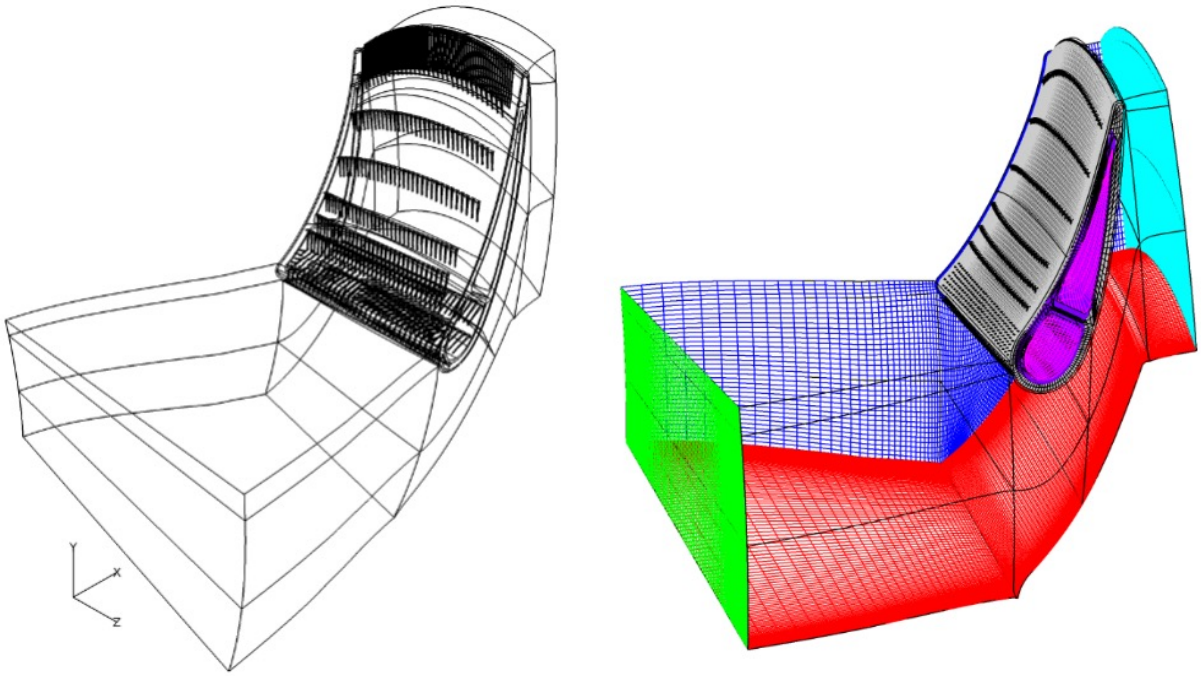


Figure 4.25: Vane case computational grid outline(left) and computational grid every 4^{th} point shown (right).

A fine grid is required in this particular example due to the large number of cooling holes that blanket the airfoil surface, especially on the vane pressure surface. Stream-wise clustering of the airfoil surface grid is performed in STAGE3 in the vicinity of the film-cooling hole rows. The computational grid shown in figure 4.25 was constructed automatically in 49 minutes on a single core of the Air Force Research Laboratory (AFRL) Mustang computer. The description of the computational grid blocks is given in table 4.6.

Advances in Model/Grid Generation

1. Cooling Plenum Walls

Walls of the internal cooling plenum are constructed based on prescribed STAGE3 inputs of plenum wall thickness and location of inter-plenum walls as well as the airfoil surface. Inter-plenum walls are constructed based on prescribed inter-plenum wall thickness and the axial locations where the inter-plenum wall intersects the upper and lower airfoil surfaces. Inter-plenum outer wall boundaries are then determined as a straight line between the upper and lower airfoil intersection points. Once the inter-plenum outer wall boundary coordinates are determined, each plenum then has a closed outer wall boundary. The inner plenum wall could be determined by projecting a normal from the outer plenum wall boundary if the wall thickness is large enough. However, problems arise in regions where the airfoil thickness becomes small such as near the trailing edge where the inner plenum wall boundary could overlap. This can always be avoided by sacrificing the use of boundary normals for the plenum wall grid. Instead, the inner plenum wall coordinates can be determined by first finding the centroid of each plenum and then projecting a line between the centroid and each point on the plenum outer wall boundary. This step guarantees non-overlapping grids. Next, the plenum inner wall boundary coordinates are initially determined as the location

that is some percentage distance (example 90%) along the line between the centroid and plenum outer wall boundary. The plenum inner wall boundary location is then corrected to make sure the plenum wall thickness is held. The corners of the plenum inner wall boundary between the region of the airfoil surface and the inter-plenum walls are rounded using a spline to avoid sharp corners in the wall that could create stress or thermal concentration. As an example, figure 4.26 shows two cooling plenums used for the film-cooled turbine vane.

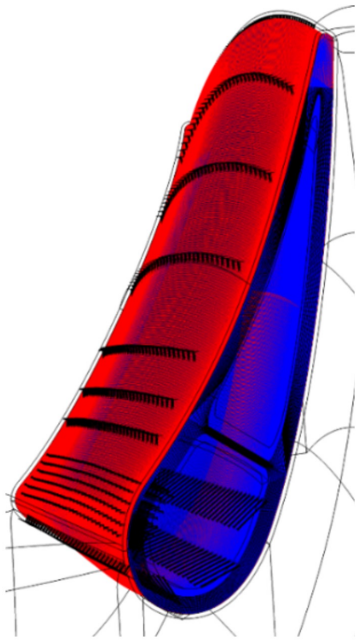
2. Thermal Barrier Coating (Kapton Layer)

Thermal barrier coatings may be included in the model with a single computational O-grid around the airfoil based on prescribed coating thickness. For the film-cooled turbine vane example shown in figures 4.25, a thin polyimide film coating, called a Kapton layer, is used to embed the heat flux gauges into the actual vane used in the experimental configuration that also acts as a thermal barrier coating. Figure 4.26 also shows the computational grid used to model this Kapton layer for the film-cooled turbine vane.

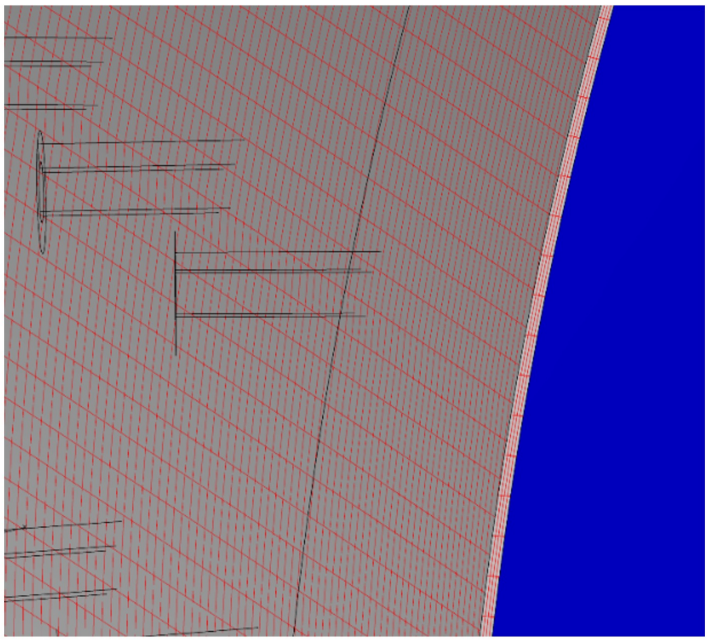
3. Cooling Holes

Cooling hole blocks consist of an O-grid block just interior to the surface of the cooling hole and an H-grid that resides inside of the O-grid throughout the length of the cooling hole. Cooling holes can be flared at their exit where the cooling flow enters the main passage flow. The cooling hole geometry STAGE3 inputs specified by the designer consist of:

- Number of cooling hole rows.
- Number of cooling holes within a row.
- The side of the airfoil (lower or upper) the cooling hole row corresponds.
- The radial location where the cooling hole row responds or a set of five quadratic curve



Cooling Plenums



Thermal Barrier Coating Grid (Kapton layer-shown in red)

Figure 4.26: Cooling Plenum model/grid and thermal barier coating (Kapton) grid

coefficients that describe the radial locations of the cooling holes within a row.

- The diameter of the holes within a row.
- The absolute or relative angles to the airfoil surface of the hole. Angles are in the x, θ frame and the θ r frame.
- The flare angle of cooling holes within a row, if desired.

Cooling holes can be flared at their exit in order to improve the film-cooling effectiveness. For the cooling holes that are flared, the outer boundary of the cooling hole is transformed from a circle at a location just inside of the plenum wall to some desired geometry (e.g. square, etc.) at the cooling hole exit. The cooling hole O-grid is then transfinite interpolated between the circular O-grid inner radial boundary to the cooling hole outer radial boundary.

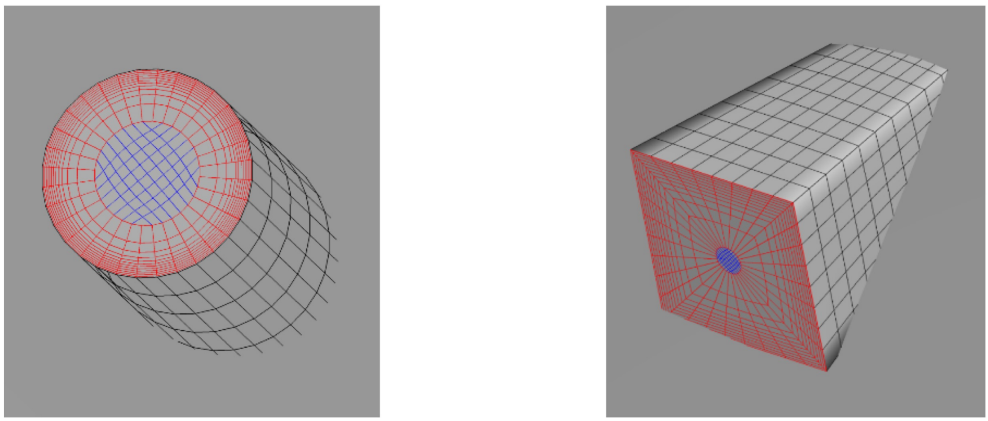


Figure 4.27: Standard and flared overlaid cooling surface grids

4. Airfoil/Endwall Fillets

Near-circular fillets of specified radius in STAGE3 can be applied at the intersection of the airfoil and endwalls that are used in manufactured blade-rows. A circular shape is fitted in the corner between the airfoil and endwall at each streamwise grid location. The fillet does not affect the thickness of any thermal barrier coating that may exist but does affect the local thickness of any plenum wall with the inner plenum wall boundary not affected. The end-wall fillets on the vane blade is shown in figure 4.28

4.5.2 Results and Validation

To validate MBFLO3 overlaid-grid capability and validate the overall multi-blade-row, conjugate simulation procedure, a simulation of a high-pressure, film-cooled, transonic turbine vane tested experimentally at the Air Force Research Laboratory is used. Transonic turbine rig tests were conducted in the Turbine Research Facility at Wright-Patterson Air Force Base (Dayton, Ohio) [54]. Details of the experimental data are given by Johnson et. al [35],[36], [37] and Ooten [54]. Comparisons between the experimental data and other predic-

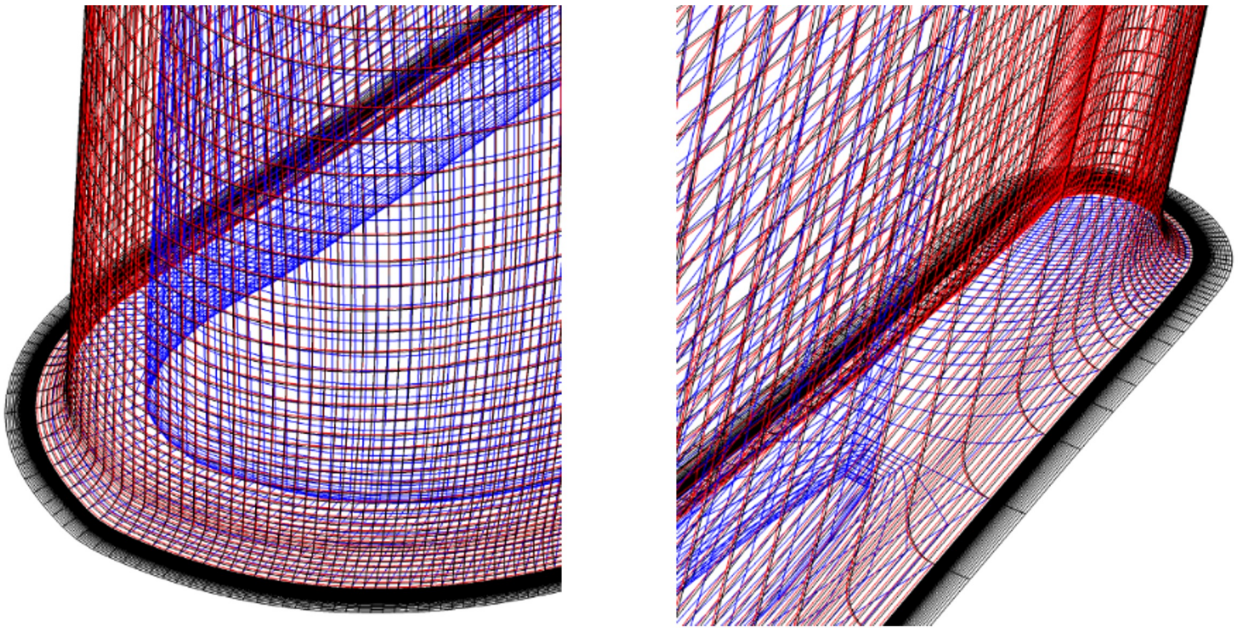


Figure 4.28: Standard and flared overlaid cooling surface grids

tions can be found in [54],[53],[52],[51]. The Turbine Research Facility, TRF shown in figure 4.29, is a full-scale transient facility designed to observe a single-spool turbomachine at flow conditions consistent with turbine environments and extract time-resolved pressures, surface temperature, and heat flux data.

For the vane-alone experiments, the airfoil count in the experimental rig consisted of 23 first vanes, shown in figure 4.29, around the wheel. The surface temperature and heat flux data were measured using surface mounted (embedded in the Kapton layer) thermocouples and double-sided thin-film heat-flux gauges, respectively. A radially-centered rake of nine pressure gauges collects the total pressures up and downstream of the turbine vane. These rakes swept 120 degrees of the annulus throughout the duration of the experiment and measured the total pressure at the inlet of the first vane and the total pressure at the exit of the second vane. The pressure transducers used are Kulite transducers, model LQ-062, and

they have uncertainty of measurements to be 0.05 percent of the full scale output, resulting in 344.7 Pa on the first vane. The pressure data was sampled at a rate of 800 kHz. The first vane in the transonic experimental turbine rig had airfoil surface and endwall film-cooling. A total of 16 first vanes out of the 23 around the wheel had film-cooling with most of the cooling holes located on the pressure surface of each airfoil, fewer on the suction surface of each airfoil, and even fewer on the first vane hub and casing endwalls.

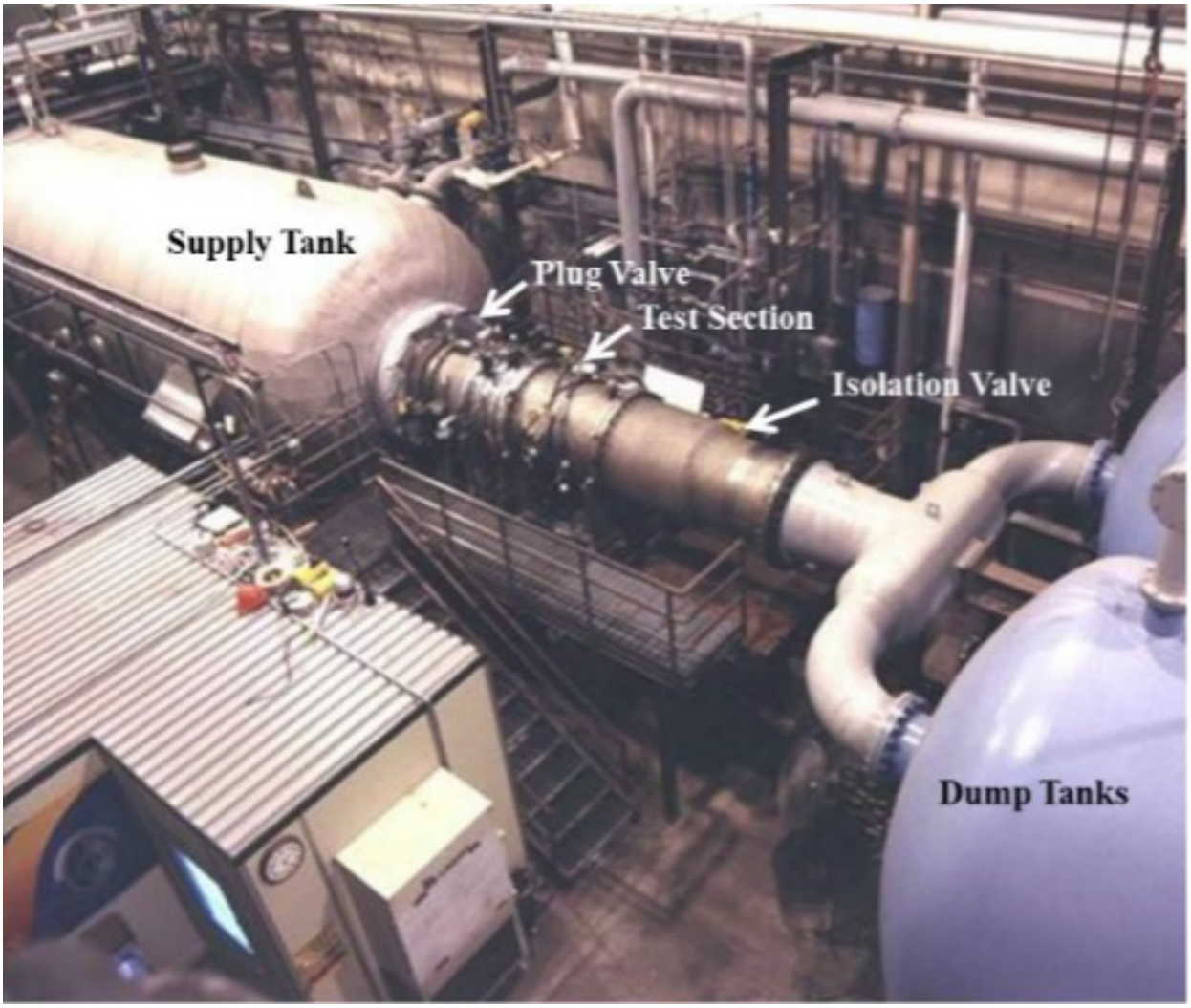


Figure 4.29: Experimental TRF wind tunnel.

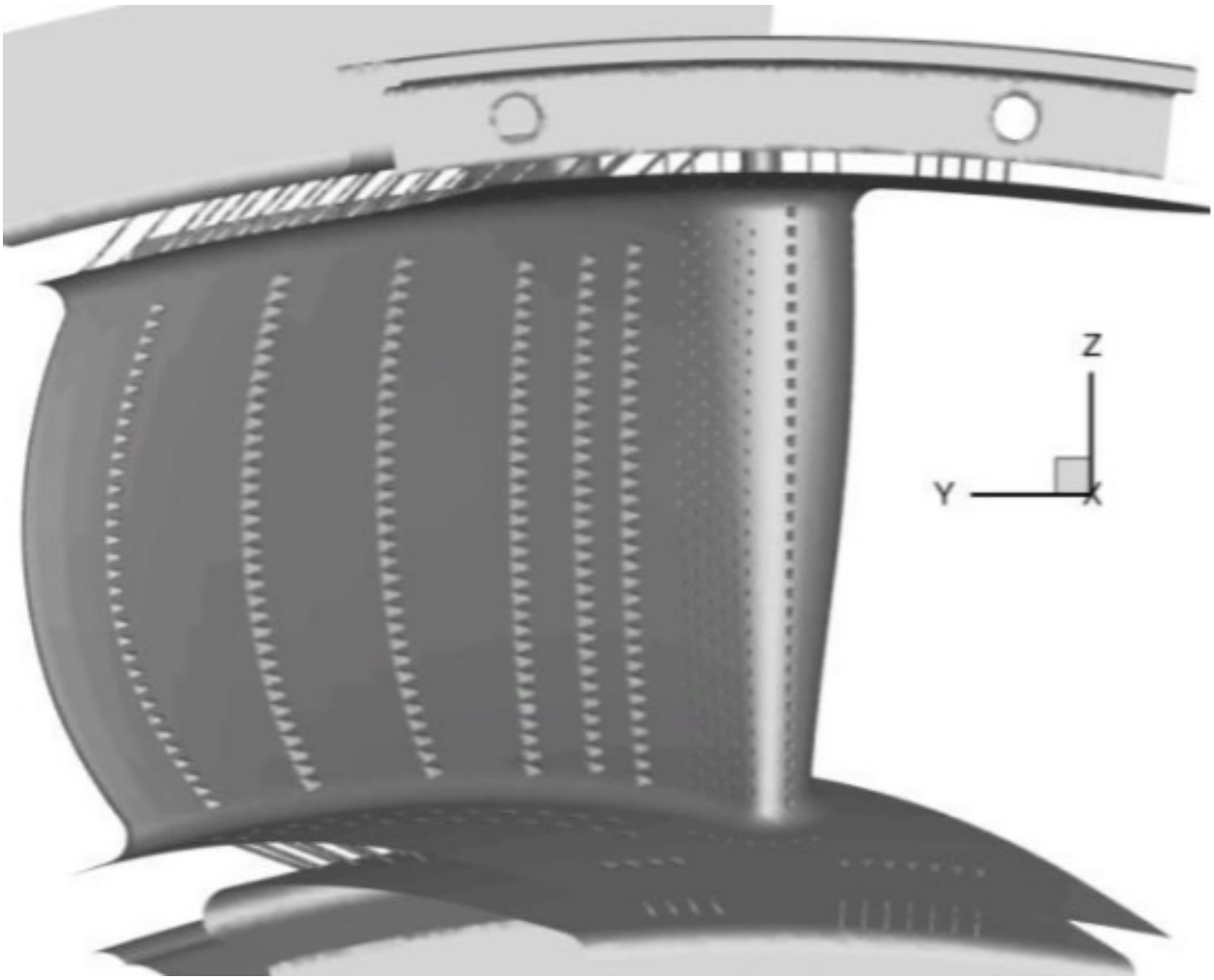


Figure 4.30: High Pressure Vane geometry.

Steady Turbine Vane Simulation

A steady simulation of the vane-alone configuration with grid dimensions previously described in Table 4.6 and computational grid shown in figures 4.25-4.28 has been executed on various high-performance computer clusters. The film-cooling holes on the airfoil surface were modeled using the standard and flared overlaid-grids, as previously described, with design-intent prescribed geometry. The film-cooling holes on the rig endwalls were not

Inlet Pressure	451,678.3 Pa (65.51 psi)
Inlet Total Pressure	453,677.8 Pa (65.8 psi)
Average Exit Pressure	266,897.7 Pa (38.71 psi)
Upstream Cooling Plenum Inlet Total Pressure [[35],[36], [37]	460,779.4 Pa (66.83 psi)
Upstream Cooling Plenum Inlet Total Temperature [35],[36], [37]	243 K (437.4 R)
Downstream Cooling Plenum Inlet Total Pressure[35],[36], [37]	472,086.9 Pa (68.47 psi)
Downstream Cooling Plenum Inlet Total Temperature [35],[36], [37]	242 K (435.6 R)
Upstream Cooling Plenum Inlet Total Pressure [52]	472,290.9 Pa (68.5 psi)
Upstream Cooling Plenum Inlet Total Temperature [52]	241.9 K (435.4 R)
Downstream Cooling Plenum Inlet Total Pressure [52]	460,914.5 Pa (66.85 psi)
Downstream Cooling Plenum Inlet Total Temperature [52]	243.6 K (438.6 R)
Real Gas Constant for Air	286 J/(kg K) (1716.32 ft lbf/(slug R)
Airfoil Solid Thermal Conductivity	15 N/(s K) (1.8735 lbf/(s R))
Airfoil Solid Specific Heat	480 $m^2/(s^2$ K) (2870.55 $ft^2/(s^2$ R))
Airfoil Solid density	7996.54 kg/m^3 (15.52 slugs/ft3)
TBC (Kapton) Solid Thermal Conductivity	0.12 N/(s K) (0.015 lbf/(s R))
TBC (Kapton) Solid Specific Heat	1090 $m^2/(s^2$ K) (6518.54 $ft^2/(s^2$ R))
TBC (Kapton) Solid density	1419.4 kg/m^3 (2.7548 slugs/ft^3)

Table 4.7: Aerodynamic and material conditions for the Vane simulation.

modelled in the current simulation. The inlets to the airfoil internal cooling plenums were located at the hub for the upstream plenum and at the casing for the downstream plenum where the flow was held normal to the boundary. The airfoil plenum H-grid lower and outer radial boundaries were used to hold the cooling plenum inlet conditions. The aerodynamic and material conditions used in the simulation are listed in Table 24.7. The original cooling plenum inlet conditions given by Ooten[54]and Johnson et. al[35],[36], [37] are given along with those determined by Ni et. al [52] based on a root cause analysis of differences between their predictions and the experimental data. Simulations were executed for both cooling plenum inlet condition sets.

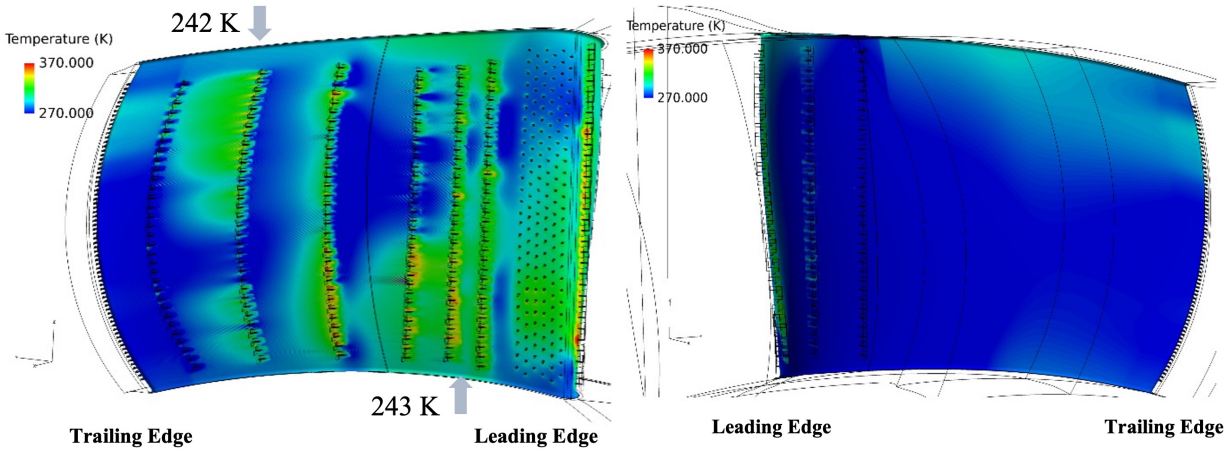
High performance computing clusters, including the University of California, Davis HPC1 Intel cluster and various government clusters including Centennial SGI cluster at the Army Research Laboratory, Mustang SGI cluster at the Air Force Research Laboratory, and Onyx Cray cluster at the Army Engineering Research and Development Center were used during

the development and validation of the current capability in MBFLO3. The number of compute nodes used in the example vane-alone simulation varied between 4 (SGI) and 6 (Cray or Intel) with the number of 132 cores on both systems. The validation simulation was executed until the maximum residual in the flow and temperature dropped three orders in magnitude and the airfoil surface temperature was no longer changing with iteration, which took under 10^5 time-steps. On the Onyx Cray cluster, each time-step took 1.28 seconds. Solution times are expected to be reduced in the future with improvements in the multiple-grid scheme and the use of multiple iterations in the conjugate solver for each time-step in the fluid solver.

1. Temperature and Heat Flux Results:

Pressure- and suction-side temperature contours are shown in figures 4.31 and 4.32 for both sets of cooling plenum inlet conditions. These contours show how the film-cooling flow is exiting the holes and affecting the surface temperature. The inlet total temperature to the mainstream and the cooling plenums are given in Table 4.7. The cooling plenum inlet conditions of Johnson et. al [35],[36], [37] result in a higher surface temperature on the upstream portion of the vane due to the higher inlet total temperature and lower inlet total pressure of the upstream cooling plenum.

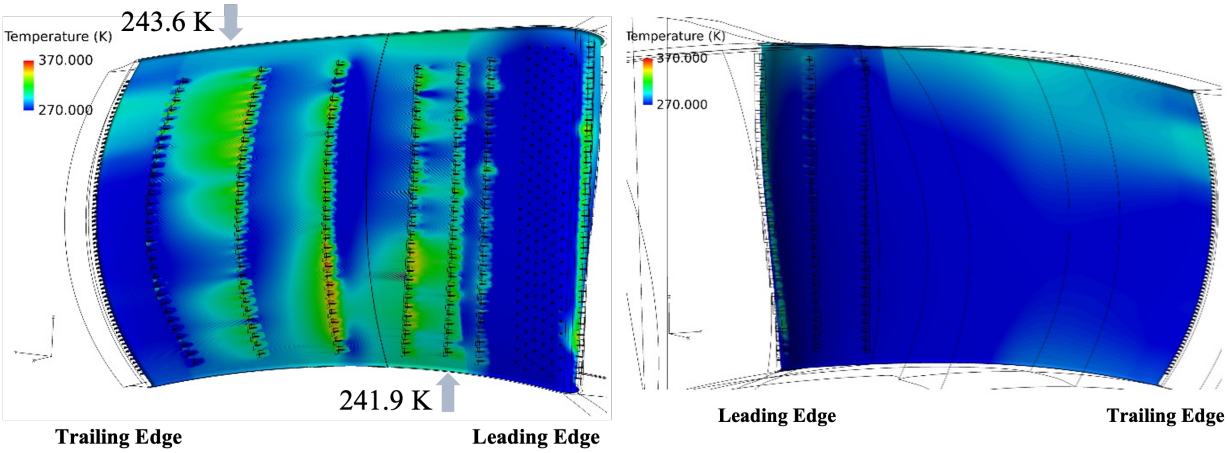
Based on total mass-flow parameter, both conditions result in a lower cooling mass-flow rate exiting the upstream cooling holes with the Johnson et. al [35],[36], [37] conditions compared to that with the Ni et. al [52] conditions. The surface temperature on the downstream portion of the vane is nearly the same for the two sets of cooling plenum inlet conditions. The downstream cooling plenum inlet total temperature is nearly the same in the two sets of conditions but the inlet total pressure of the Johnson et. al [35],[36], [37] conditions is 2% higher than that of the Ni et. al [52] conditions which results in a slightly higher mass-flow exiting the downstream cooling holes. The slightly lower surface temperature resulting from



(a) Pressure side

(b) Suction side

Figure 4.31: Vane surface temperature contours using cooling plenum inlet conditions from Johnson et. al. [35],[36], [37]



(a) Pressure side

(b) Suction side

Figure 4.32: Vane surface temperature contours using cooling plenum inlet conditions from Ni et. al. [52]

the slightly higher mass-flow in the downstream cooling holes of the Johnson et. al [35],[36], [37] conditions compared to that with the Ni et. al [52] conditions is offset.

Figures 4.33 and 4.34 shows pitchwise temperature contours at various spanwise locations through both the fluid and solid domains. The figures show the film-cooling flow exiting the holes and how much the cooling flows penetrate the mainstream flow. The cooling flows lie along the airfoil surfaces as desired and do not separate the flow along the airfoil surfaces. The figures also show the temperature distribution through the solids and the effect of the fluid cooling flows. As described for figures 4.31-4.32, the pressure side temperature is higher for the Johnson et. al [35],[36], [37] inlet cooling plenum conditions than that predicted using the Ni et. al [52] cooling plenum inlet conditions.

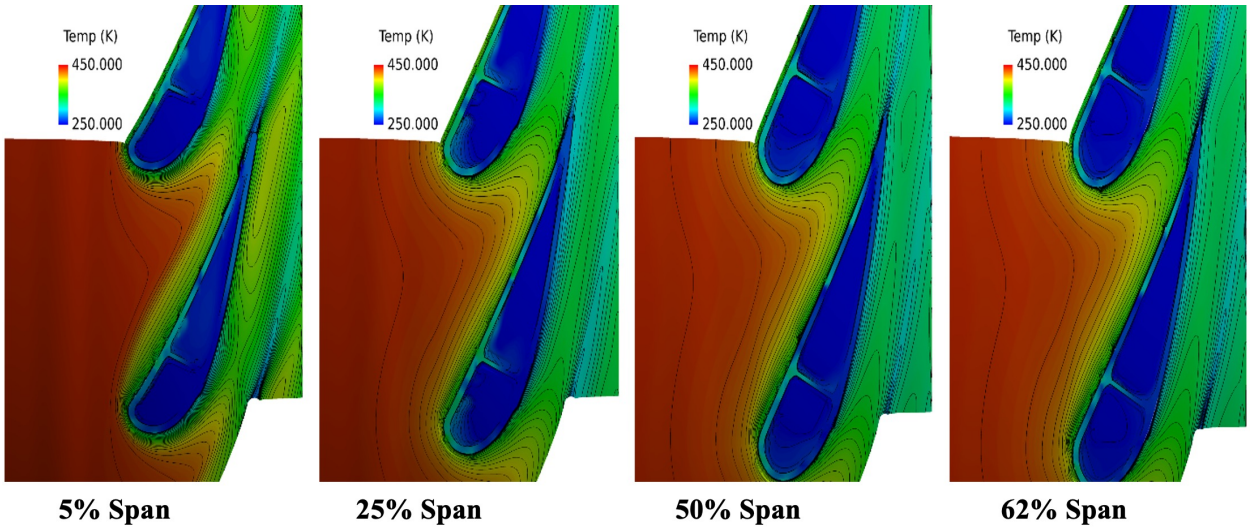


Figure 4.33: Pitchwise cooling plenum temperature contours, using cooling plenum inlet conditions from Johnson et. al [35],[36], [37]

Figures 4.35 and 4.36 shows temperature distributions at 5%, 25%, 50% and 62% span. The results are in fair agreement with the experiment. At 5% span, the temperature is high on the suction side ($X/Bx > 0$ where Bx is the axial chord) near the trailing edge which is likely

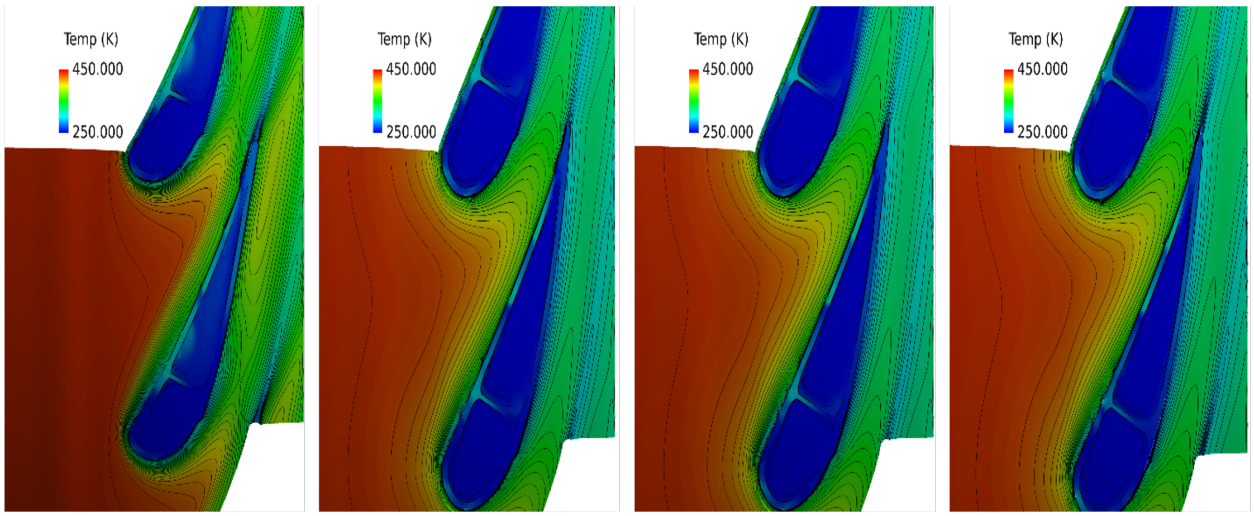


Figure 4.34: Pitchwise cooling plenum temperature contours, using cooling plenum inlet conditions from Ni et. al [52].

due to the lack of modeling the endwall film-cooling flow. As shown in figures 4.31-4.32, the temperatures in the front half of the vane for the Johnson et. al [35],[36], [37] conditions are higher than those with the Ni et. al [52] conditions. The temperatures in the aft half of the vane are similar between the two sets of conditions due to the flow and heat transfer physics explained for figures 4.31-4.32.

The heat flux distributions at the 5%, 25%, 50% and 62% span-wise locations are shown in figures 4.37 and 4.38. All of the heat flux distributions are in reasonably good agreement with the experimental data for both sets of conditions.

2. Pressure Results:

Figures 4.39-4.40 shows pressure- and suction-side contours of the pressure. The pressure loading on the airfoil surface is somewhat two-dimensional with only minor three-dimensional effects. The pressure loading is affected locally only slightly by the film-cooling flow. The

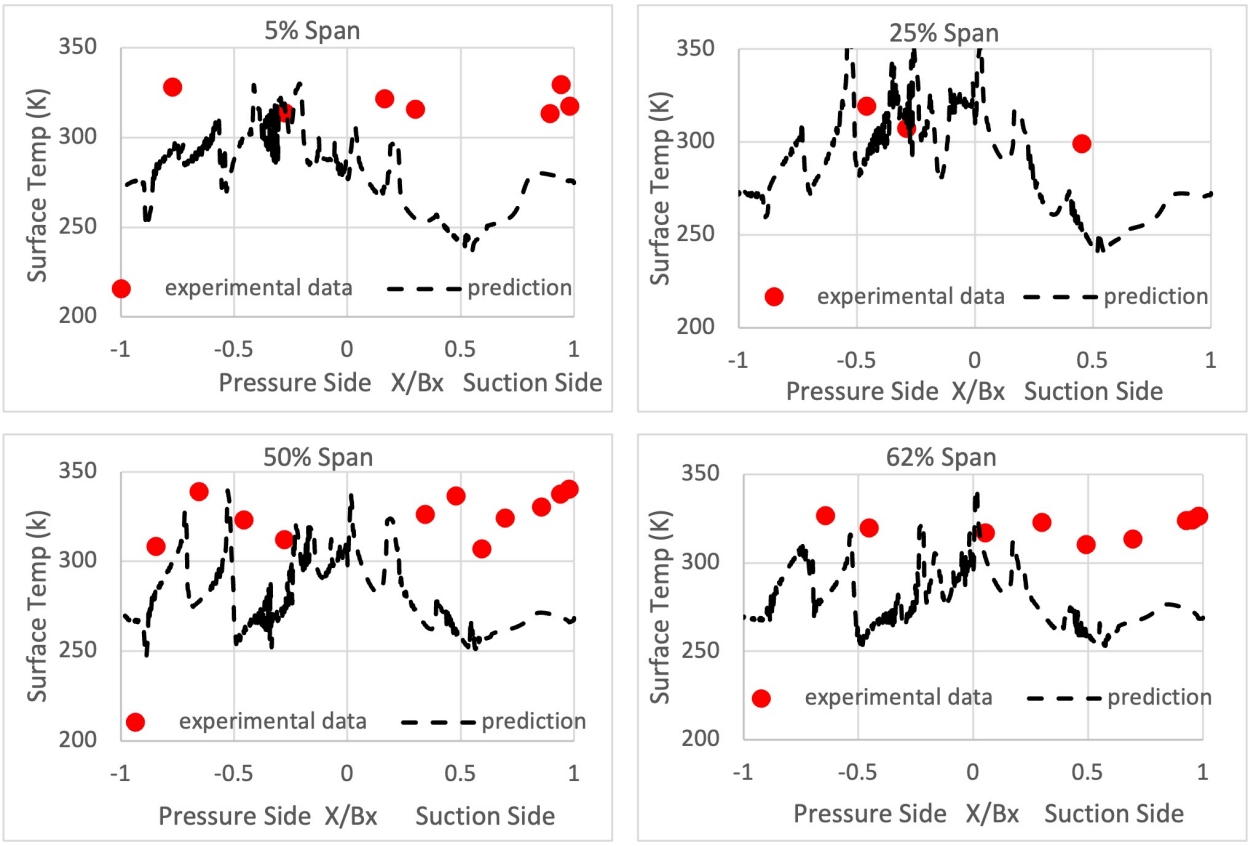


Figure 4.35: Surface temperature distributions, using cooling plenum inlet conditions from Johnson et. al [35],[36], [37].

pressure contours are nearly the same for the two sets of cooling plenum inlet conditions.

Figures 4.41-4.42 show pitch-wise contours of the static pressure at the 5%, 25%, 50%, and 62% spanwise locations. The figures show how the film-cooling flow only slightly affects the local pressure as it exits the holes. As shown in figures 4.39 and 4.40 the pressure field does not vary significantly spanwise. The pitch-wise contours of static pressure are very similar between the simulations with the two sets of cooling plenum inlet conditions.

Figures 4.43 and 4.44 show the pressure distributions at 5%, 25%, 50%, and 62% span. The loading distributions are in good agreement with the experimental data and predictions

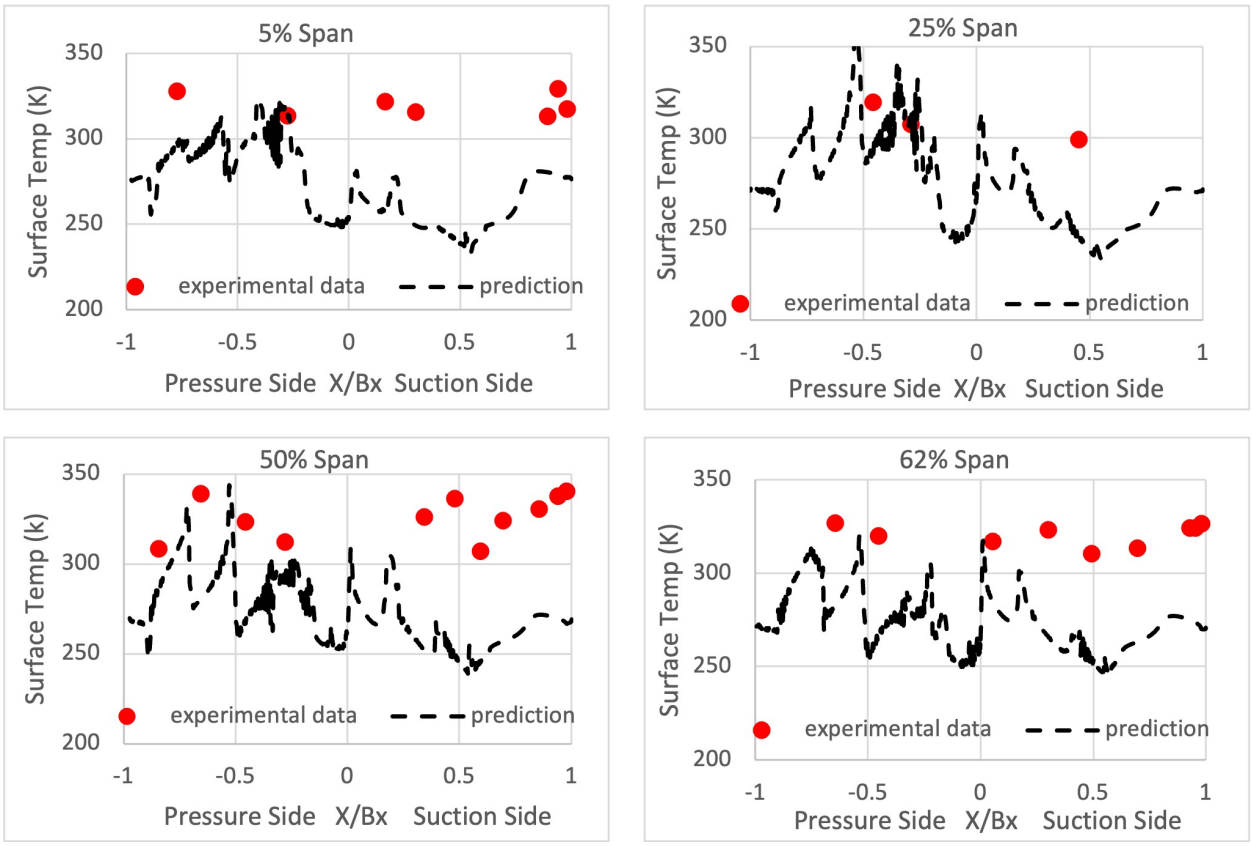


Figure 4.36: Surface temperature distributions, using cooling plenum inlet conditions from Ni et. al [52].

of Ooten [54]. The predicted loadings are essentially the same for both sets of cooling plenum inlet conditions. The predicted loadings are slightly lower than what is shown by the experimental data. Sudden jumps in the predicted pressure are shown where the cooling air exits into the mainstream flow.

3. Mach Number Results:

Figures 4.45 and 4.46 show pitch-wise pressure contours of Mach number. The Mach number at the throat of the vane passage is the greatest at the lower spanwise postions. The Mach number contours are essentially the same for both sets of cooling plenum inlet conditions.

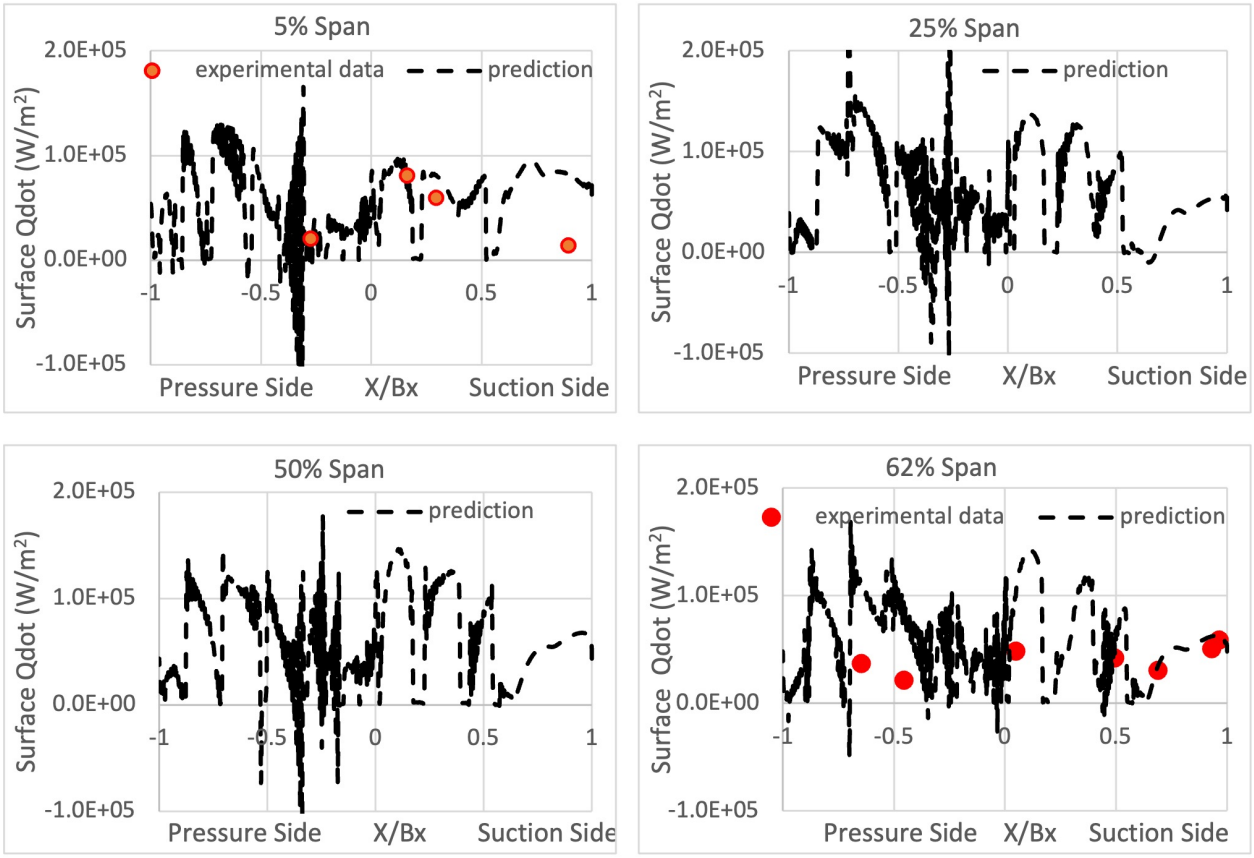


Figure 4.37: Surface heat flux distributions, using cooling plenum inlet conditions from Johnson et. al [35],[36], [37].

The Mach number inside the cooling plenums varies spanwise between 0 and 0.12 in the upstream cooling plenum and between 0 and 0.095 in the downstream cooling plenum. For flow initialization purposes, an average cooling plenum Mach number of 0.05 would be a reasonable value to use.

The validation results shown in figures 4.31-4.46 demonstrate the ability of the MBFLO3 solution procedure to model film-cooling flows in axial turbines and solve for the conjugate heat transfer through the airfoil metal and thermal barrier coating (Kapton) surfaces. The presented method of automated grid generation and conjugate heat transfer simulations with

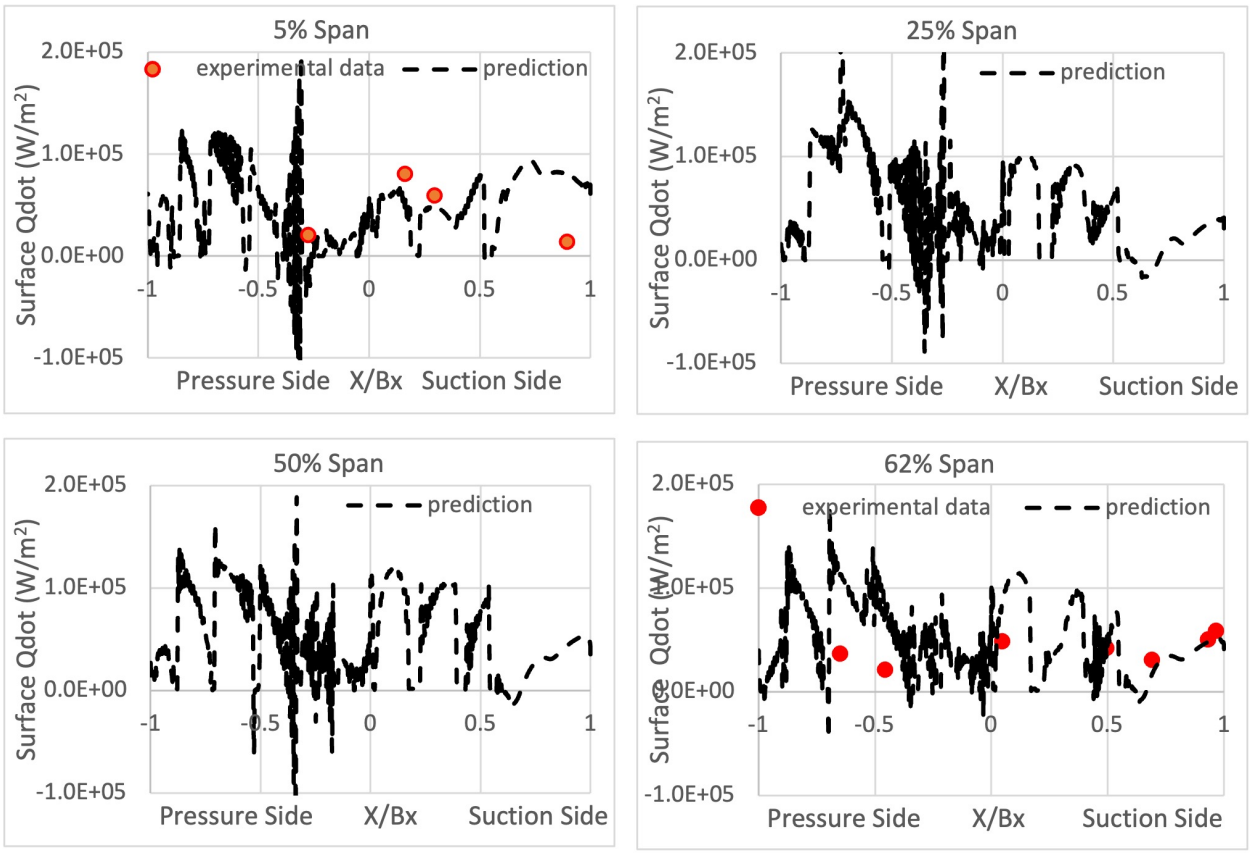
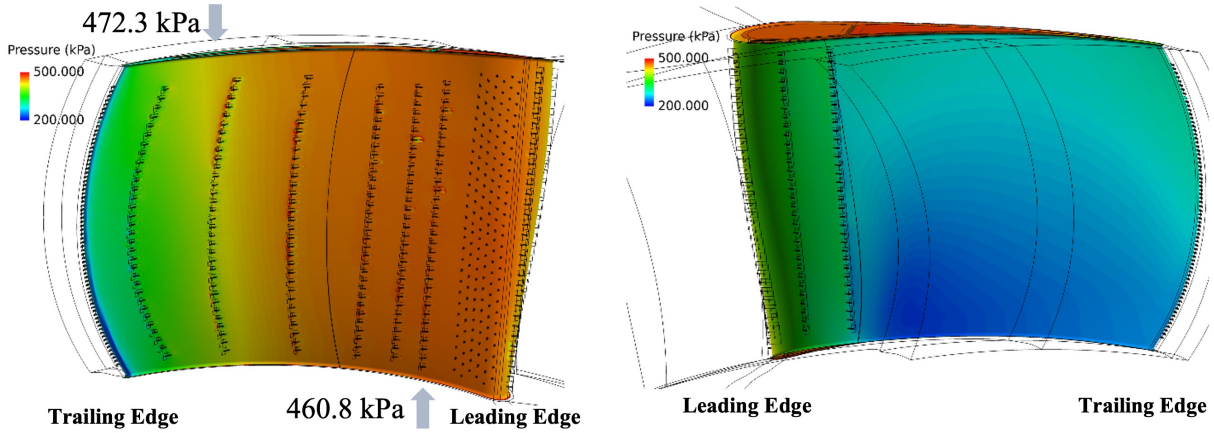


Figure 4.38: Surface heat flux distributions, using cooling plenum inlet conditions from Ni et. al [52].

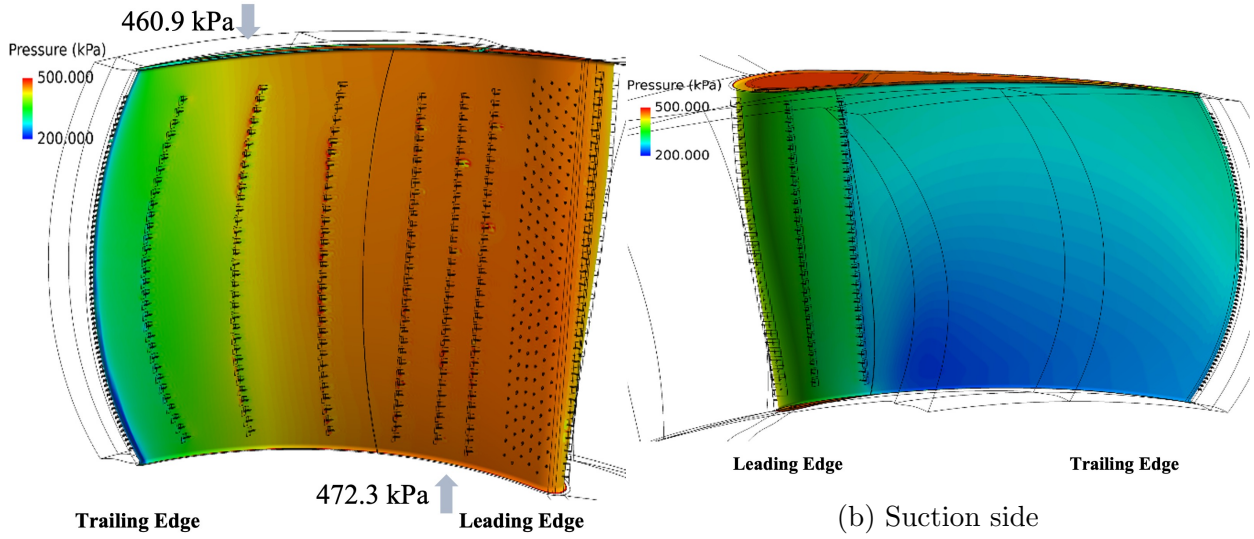
over-laid grids has been proven to be valid.



(a) Pressure side

(b) Suction side

Figure 4.39: Vane surface pressure contours using cooling plenum inlet conditions from Johnson et. al. [35],[36], [37].



(a) Pressure side

(b) Suction side

Figure 4.40: Vane surface pressure contours using cooling plenum inlet conditions from Ni et. al. [52].

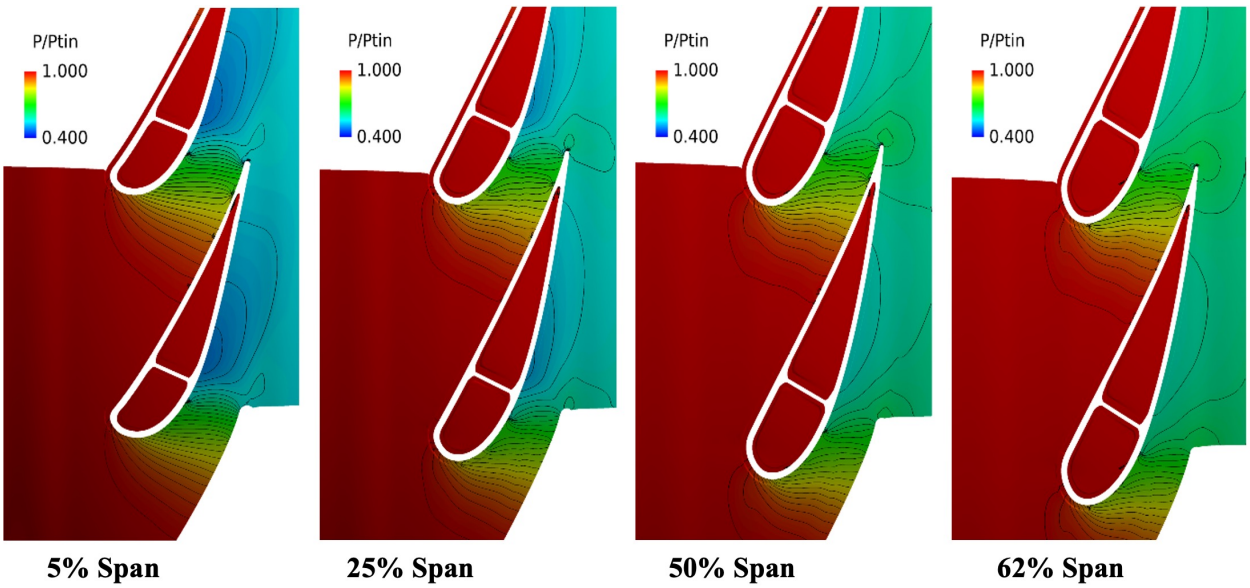


Figure 4.41: Pitchwise cooling plenum pressure contours , using cooling plenum inlet conditions from Johnson et. al [35],[36], [37].

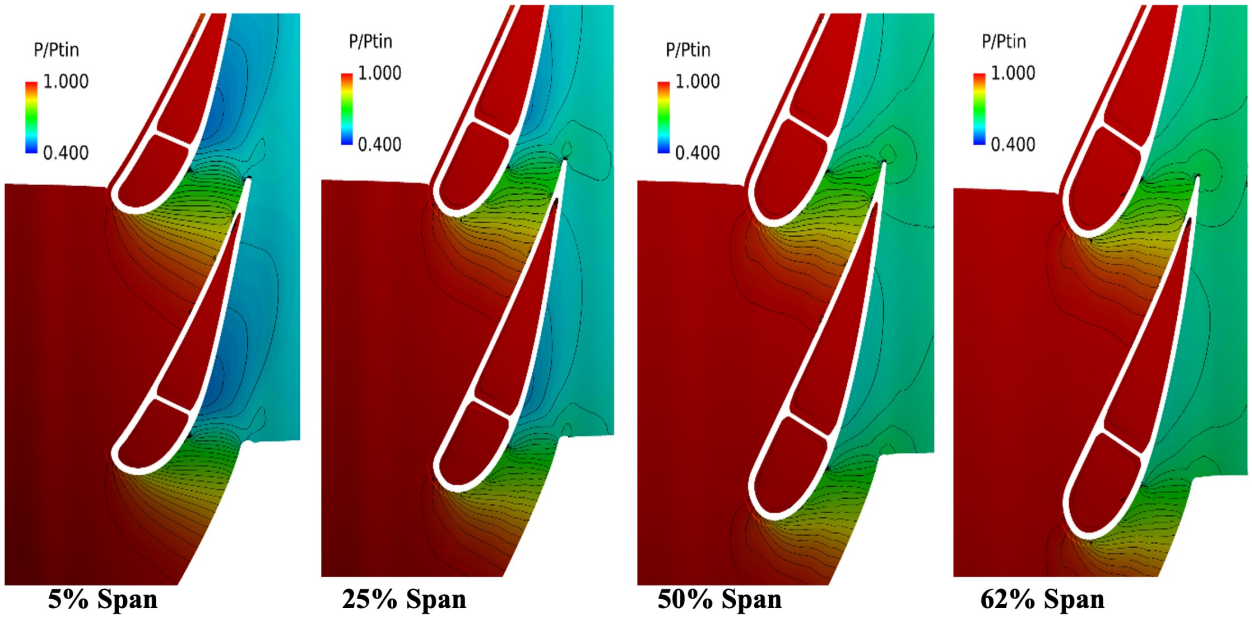


Figure 4.42: Pitchwise cooling plenum pressure contours, using cooling plenum inlet conditions from Ni et. al [52].

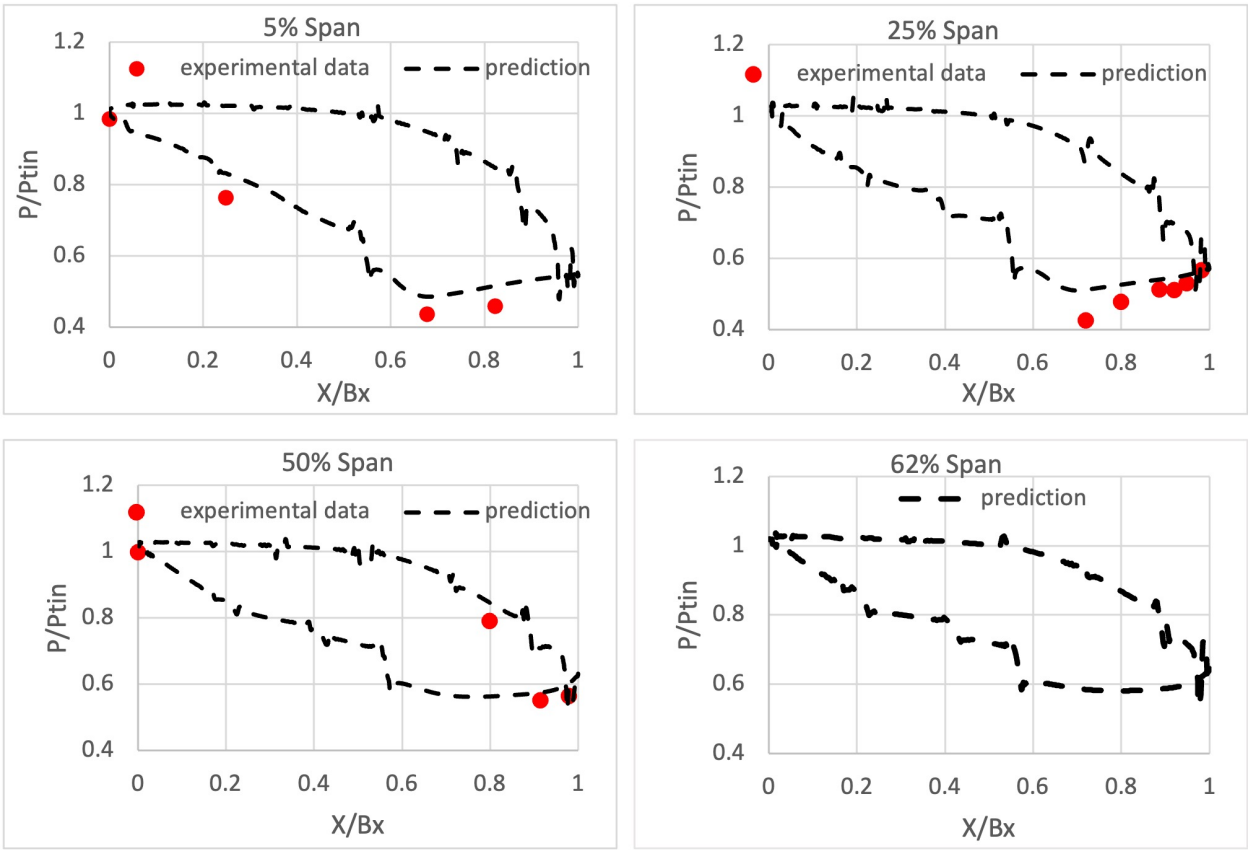


Figure 4.43: Surface pressure distributions, using cooling plenum inlet conditions from Johnson et. al [35],[36], [37].

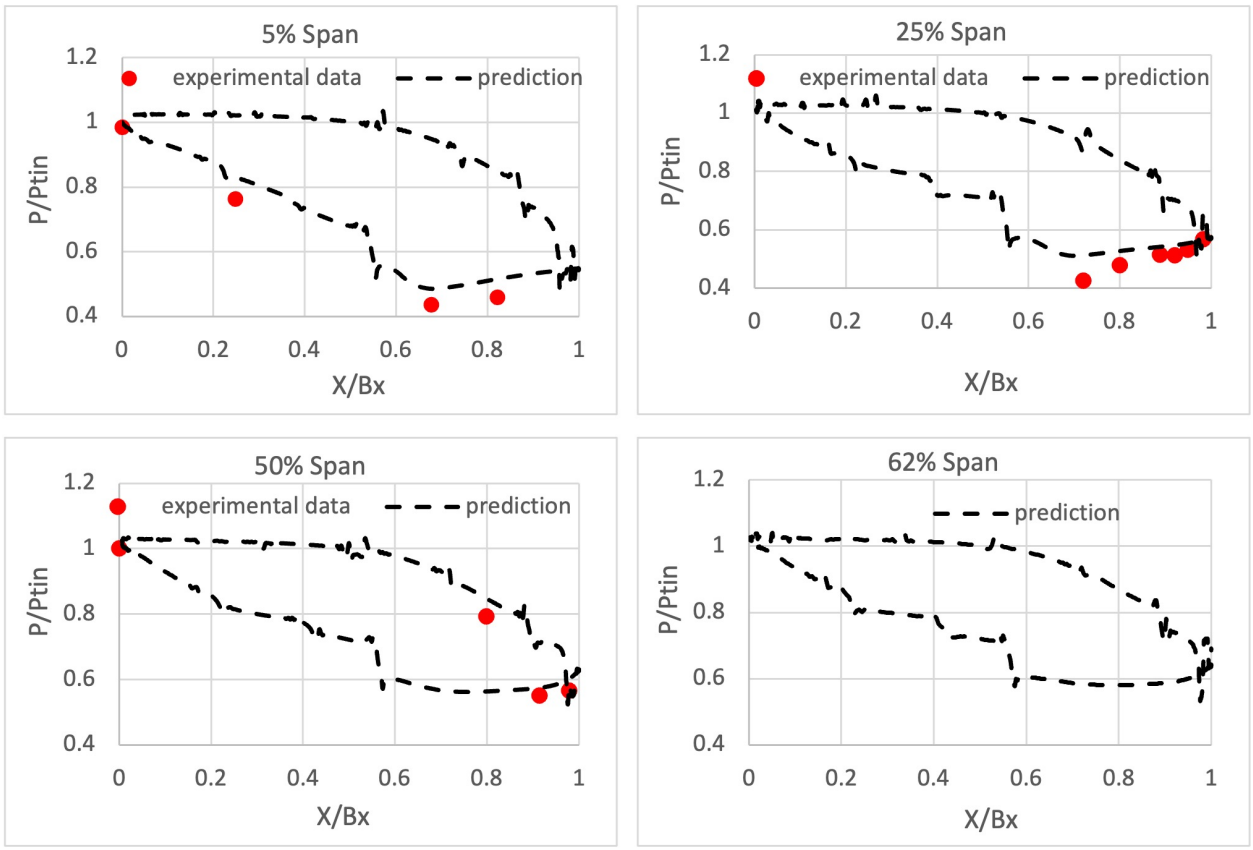


Figure 4.44: Surface pressure, using cooling plenum inlet conditions from Ni et. al [52].

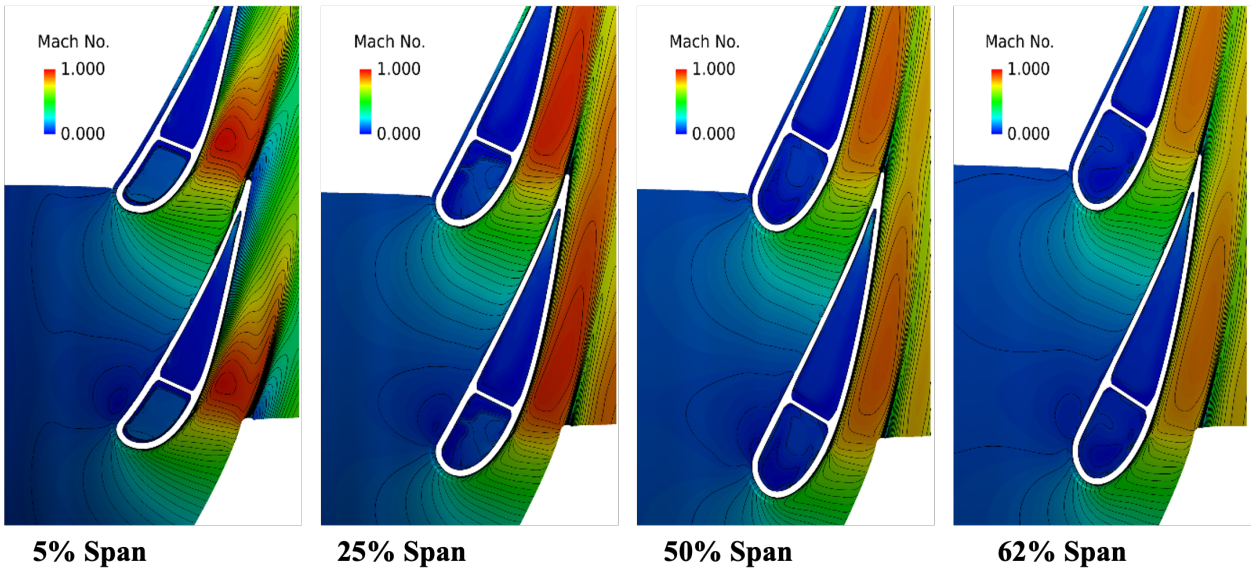


Figure 4.45: Pitchwise cooling plenum Mach number contours, using cooling plenum inlet conditions from Johnson et. al [35],[36], [37].

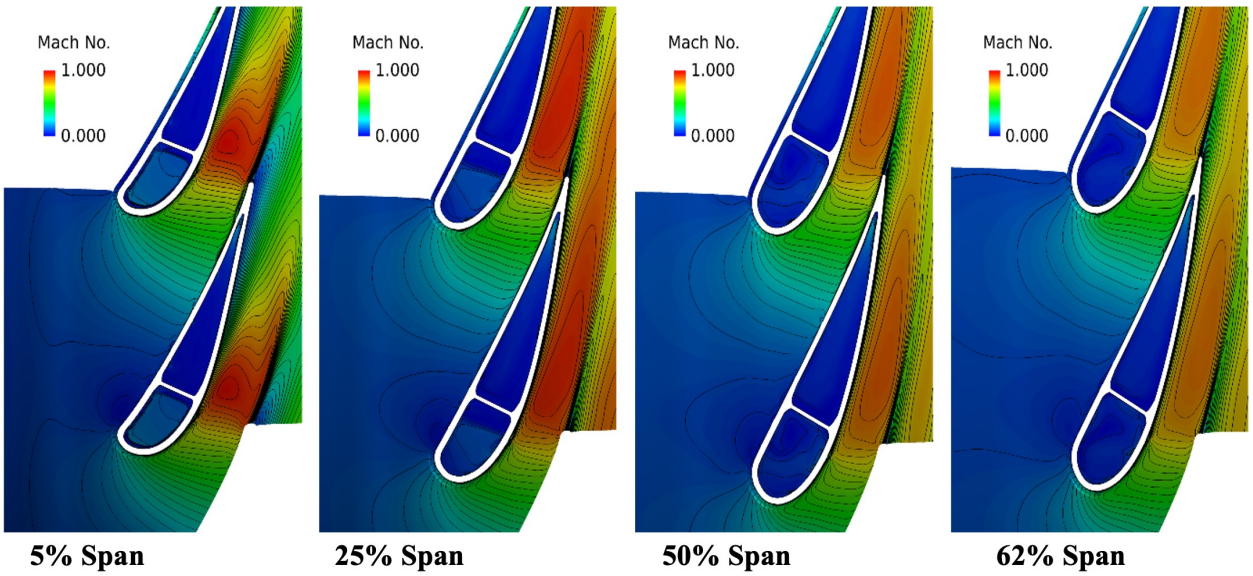


Figure 4.46: Pitchwise cooling plenum Mach number contours, using cooling plenum inlet conditions from Ni et. al [52].

Chapter 5

Summary and Recommendations

5.1 Summary

This work details the development effort of the in-house, high-fidelity, fully coupled, fluid-thermal-structure interaction finite volume code, MBFLO3. A three-dimensional finite volume based structural solver capable of simulating arbitrarily large strains has been integrated into the existing fluid-thermal interaction code to develop a monolithic, fully coupled, fluid-thermal-structure interaction technique. The structural solver presented is verified through two test cases. The first test case is a three-dimensional cantilever beam subject to a gravitational load that is verified using theory and two-dimensional simulations reported in literature. The second test case is a three-dimensional highly deformable cantilever plate subject to a gravitational load. The results of this case are verified through a comparison with the modal response calculated by commercially available software.

The fully coupled, monolithic, fluid-structure interaction method is capable of simulating

large structural displacements while maintaining mesh quality through the use of a Poisson smoothing technique. The FSI method is verified using Turek and Hron's [63] cylinder-flag configuration. Throughout the development process of coupling the fluid and structural solvers, several solid domain smoothing techniques were investigated and combined to maintain solution integrity during fluid-structure interaction simulations.

The MBFLO3 solution procedure has been extended to handle overlaid grids and the conjugate heat transfer modeling of multiple solids using spatial second-order accurate treatments. A three-dimensional, steady, conjugate simulation of a film cooled vane with thermal barrier coating is compared with experimental data from a vane-alone experiment at the Air Force Turbine Research Facility. The numerical techniques used in the conjugate solution procedure are described and the procedure is validated against the vane-alone experimental data.

5.2 Recommendations

The next steps in the development process of MBFLO3 are to validate the fluid-structure interaction solver. There are several test cases that could be used for the validation effort including Giordano [20], Yates et. al [69] and Graves [24]. Graves et. al [24] may be a particularly interesting test case because the experimental data provided details the displacement and the first modal frequency at several locations of the MAVRIC-I wing with various wing configurations across a range of Mach numbers and angle of attack.

Recall from chapter 1 the eventual goal of this research was to perform a fully coupled fluid-thermal-structure interaction simulation on a $1\frac{1}{2}$ stage turbine with thermal barrier coating and film cooling holes on the first stage. The proposed computational domain is enormous

with over 3000 computational blocks and over 150 million grid points. Three considerations must be addressed before a simulation of this magnitude can be performed. First, an understanding of how film cooling holes will be allowed to vibrate in an FSI simulation. Second, whether and how to consider the pressure and shear stress imparted by the fluid onto the solid through the film cooling hole. Third, the numerical stiffness of the fully coupled, monolithic, fluid-thermal-structure interaction method should be reduced.

During a fluid-thermal-structure interaction simulation of the $1\frac{1}{2}$ stage turbine, the structure will displace resulting in film cooling hole displacement. Film cooling holes should be allowed to displace along with the solid domain, but this motion would require re-interpolation of overlaid grid points that would prove to be memory intensive and time consuming. Instead, cells which have overlaid points (underlying cells) should displace rigidly with the solid and film cooling hole overlaid grid to avoid re-interpolation. This method should work for simulations with relatively small aeroelsatic displacements, which should suffice for most film-cooled turbomachinery configurations.

It is clear that the presence of cooling holes will somewhat act as stress concentrators on the turbine blades. Pressurized flow is being ejected from these holes which create a complex thermal distribution throughout the blade and further loads the blade near the holes because of the high cooling plenum pressure. There are pressure and shear stress loads acting on the walls of the film cooling holes, on the plenum walls (the interior of the blade), and the blade coating surface. A methodology currently exists within MBFLO3 for load transfer between fluids and solids with point matched grids, which exist at all fluid-solid interfaces except inside the film cooling holes. An efficient load transfer methodology between the overlaid grids of the film cooling holes and the solid will be needed to simulate the effects of the film cooling holes on the blades. However, the extent to which the surface pressure and

shear loads on the film cooling hole surface within the solids will effect the forced response of the turbine blades is not known, and may be minimal. This hypothesis could be tested by considering the single film cooling hole and thermal barrier coating test case presented by Jiang et. al [34]. In this simulation, a low pressure hot gas and high pressure cooler gas flow in parallel and are separated by a metal plate with thermal barrier coating and a single film cooling hole. After demonstrating good agreement in surface temperature predictions with experimental results, Jiang et. al [34] predicted large thermal stresses near the film cooling hole. A similar analysis could be conducted using MBFLO3 with and without fluid loading on the interior of the film cooling hole. The resulting simulation predictions could inform whether it is necessary to include fluid loading on the interior of film cooling holes, and serve as a validation case for the fluid-thermal-structure interaction method.

The fluid-thermal-structure interaction simulation of the $1\frac{1}{2}$ stage turbine with film cooling holes and thermal barrier coating will be computationally expensive and numerically stiff with the current treatment of the structural solver and conjugate heat transfer solvers in MBFLO3. The numerical stiffness is a result of the point-matched grids that must be used to enable the fully coupled, monolithic solution method. The point-matched grids force the solid domain to have a much finder grid than is strictly necessary to resolve the temperature and displacement fields. The increased solid domain refinement causes a smaller local time step for the solid domain. As discussed in chapter 3 section 3.1.2, the local time step of the solid domain may be orders of magnitude smaller than the local time step of the fluid domain. This requires many additional inner iterations for the structural and conjugate solvers to obtain convergence. To improve solid domain convergence and reduce numerical stiffness, a multiple grid scheme for the solid domain or other convergence acceleration method should be investigated.

Bibliography

[1] Bates, J. C. A 3D, Parallel. Finite Volume, Implementation of a Conjugate Heat Transfer and Fluid Flow Solver. Master's thesis, University of California, Davis, 2016.

[2] R. Bozinoski and R. Davis. General Three-Dimensional, Multi-Block, Parallel Turbulent Navier-Stokes Procedure. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2008. American Institute of Aeronautics and Astronautics.

[3] Bozinoski, R. *Accurate Prediction of Unsteady and Time-Averaged Pressure Loads using a Hybrid Reynolds-Averaged/Large-Eddy Simulation Technique*. Dissertation, University of California, Davis, 2009.

[4] L. Castillon, G. Billonnet, J. Riou, S. Péron, and C. Benoit. A Technological Effect Modeling on Complex Turbomachinery Applications With an Overset Grid Numerical Method. *Journal of Turbomachinery*, 136(10):101005, October 2014.

[5] Boles, M. A. Cengel, Y. A. *Thermodynamics- An Engineering Approach, 5th Edition*. McGraw-Hill, 2006.

[6] Xiangying Chen, Gecheng Zha, and Mingta Yang. Numerical Simulation of 3-D Wing Flutter with Fully Coupled Fluid-Structure Interaction Approach. In *44th AIAA*

Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2006. American Institute of Aeronautics and Astronautics.

[7] V. Chetty and R. L. Davis. Higher Order Control Volume Scheme for Solving the Euler Equations. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2006. American Institute of Aeronautics and Astronautics.

[8] Gottlieb J. D., R. L. Davis, and Clark J. P. Simulation strategy for film-cooled multistage turbine design and analysis. *AIAA Journal of Propulsion and Power*, 2013.

[9] Gottlieb J. D., R. L. Davis, and Clark J. P. Conjugate rotor-stator interaction procedure for film-cooled turbine sections. *Aircraft Engineering and Aerospace Technology Journal*, 2014.

[10] J. F. III Dannenhoffer. *Grid Adaptation for Complex Two Dimensional Transonic Flows*. Doctor of philosophy, Massachusetts Institute of Technology, August 1987.

[11] R. L. Davis and J. P. Clark. Geometry-Grid Generation for Three-Dimensional Multidisciplinary Simulations in Multistage Turbomachinery. *Journal of Propulsion and Power*, 30(6):1502–1509, November 2014.

[12] R. L. Davis, R. H. Ni, and Carter J. E. Cascade viscous flow analysis using the Navier-Stokes equations. *Journal of Propulsion and Power*, 3(5):406–414, September 1987.

[13] J. Degroote. Partitioned Simulation of Fluid-Structure Interaction: Coupling Black-Box Solvers with Quasi-Newton Techniques. *Archives of Computational Methods in Engineering*, 20(3):185–238, September 2013.

[14] W. G. Dettmer and D. Perić. A new staggered scheme for fluid-structure interaction: A New Staggered Scheme for Fluid-Structure Interaction. *International Journal for*

Numerical Methods in Engineering, 93(1):1–22, January 2013.

[15] Hall, C. A. Dixon, S. L. *Fluid Mechanics and Thermodynamics of Turbomachinery, Seventh Edition*. Elsevier, 2014.

[16] H. Doi and J. J. Alonso. Fluid/Structure Coupled Aeroelastic Computations for Transonic Flows in Turbomachinery. In *Volume 4: Turbo Expo 2002, Parts A and B*, pages 787–794, Amsterdam, The Netherlands, January 2002. ASMEDC.

[17] P. Dong and R. S. Amano. High-Pressure Gas Turbine Vane Turbulent Flows and Heat Transfer Predicted by RANS/LES/DES. In *Volume 5B: Heat Transfer*, page V05BT22A001, Charlotte, North Carolina, USA, June 2017. American Society of Mechanical Engineers.

[18] A Farve. Equationds des gaz turbulents compressibles. *J Mecan*, 4:361–390, 1965.

[19] D. P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17(5):679–706, May 1981.

[20] J. Giordano, G. Jourdan, Y. Burtschell, M. Medale, D. E. Zeitoun, and L. Houas. Shock wave impacts on deforming panel, an application of fluid-structure interaction. *Shock Waves*, 14(1-2):103–110, June 2005.

[21] C. Goormans-Francke, G. Carabin, and C. Hirsch. Mesh Generation for Conjugate Heat Transfer Analysis of a Cooled High Pressure Turbine Stage. In *Volume 4: Heat Transfer, Parts A and B*, pages 615–625, Berlin, Germany, January 2008. ASMEDC.

[22] D. A. Gottfried and S. Fleeter. Aerodynamic Damping Predictions in Turbomachines Using a Coupled Fluid-Structure Model. *Journal of Propulsion and Power*, 21(2):327–

334, March 2005.

[23] J. S. Graff, R. L. Davis, and J. P. Clark. Computational structural dynamics general solution procedure using finite volumes. *Journal of Algorithms & Computational Technology*, 16:174830262210840, January 2022.

[24] S. S. Graves, A. W. Burner, J. W. Edwards, and D. M. Schuster. Dynamic Deformation Measurements of an Aeroelastic Semispan Model. *Journal of Aircraft*, 40(5):977–984, September 2003.

[25] D. Griffini, M. Insinna, S. Salvadori, and F. Martelli. Clocking Effects of Inlet Non-Uniformities in a Fully Cooled High-Pressure Vane: A Conjugate Heat Transfer Analysis. In *Turbine Technical Conference and Exposition*, page 13. American Society of Mechanical Engineers, 2015.

[26] Z. H. Han and S. Gortz. Hierarchical kriging model for variable-fidelity surrogate modeling. *AIAA Journal*, 50(9):1885–1896, 2012.

[27] Han, D., Liu, G.R., and Abdallah, S. An Eulerian-Lagrangian-Lagrangian method for 2D fluid-structure interaction problem with a thin flexible structure immersed in fluids. *Computers & Structures*, 228:106179, February 2020.

[28] Fadl, M. He, L. Multi-scale time integration for transient conjugate heat transfer. *International Journal for Numerical Methods in Fluids*, 83:887–904, 2016.

[29] Holman, J. P. *Heat Transfer, Tenth Edition*. McGraw-Hill, 2010.

[30] G. Hou, J. Wang, and A. Layton. Numerical Methods for Fluid-Structure Interaction — A Review. *Communications in Computational Physics*, 12(2):337–377, August 2012.

[31] H. Im, X. Chen, and G. Zha. Detached Eddy Simulation of Transonic Rotor Stall Flutter Using a Fully Coupled Fluid-Structure Interaction. In *Volume 6: Structures and Dynamics, Parts A and B*, pages 1217–1230, Vancouver, British Columbia, Canada, January 2011. ASMEDC.

[32] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th Computational Fluid Dynamics Conference*, Honolulu,HI,U.S.A., June 1991. American Institute of Aeronautics and Astronautics.

[33] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. In *14th Fluid and Plasma Dynamics Conference*, Palo Alto,CA,U.S.A., June 1981. American Institute of Aeronautics and Astronautics.

[34] J. Jiang, L. Jiang, Z. Cai, W. Wang, X. Zhao, Y. Liu, and Z. Cao. Numerical stress analysis of the TBC-film cooling system under operating conditions considering the effects of thermal gradient and TGO growth. *Surface and Coatings Technology*, 357:433–444, January 2019.

[35] J. J. Johnson, J. P. Clark, R. A. Anthony, M. K. Ooten, R. H. Ni, and W. H. Humber. 3D Heat Transfer Assessment of Full-Scale Inlet Vanes With Surface-Optimized Film Cooling: Part 2 — Conjugate CFD Simulations. In *Volume 5B: Heat Transfer*, page V05BT19A025, Phoenix, Arizona, USA, June 2019. American Society of Mechanical Engineers.

[36] J. J. Johnson, P. I. King, J. P. Clark, and P. J. Koch. Exploring Conjugate CFD Heat Transfer Characteristics for a Film-Cooled Flat Plate and 3-D Turbine Inlet Vane. In

Volume 4: Heat Transfer, Parts A and B, pages 1163–1174, Copenhagen, Denmark, June 2012. American Society of Mechanical Engineers.

[37] Jamie Johnson, Paul King, John Clark, Richard Anthony, Peter Koch, M. K. Ooten, E. Kasik, and R. H. Ni. Three-Dimensional Film-Cooled Vane CFD Simulations and Preliminary Comparison to Experiments. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Orlando, Florida, January 2011. American Institute of Aeronautics and Astronautics.

[38] D. A. Johnston, Cross, C. J., and Wolff, J. M. An Architecture for Fluid/Structure Interaction Analysis of Turbomachinery Blading. *ASME Joint Propulsion Conference Exhibit*, 41:12, July 2005.

[39] Y. Kim, S. Lee, and K. Yee. Variable-Fidelity Optimization of Film-Cooling Hole Arrangements Considering Conjugate Heat Transfer. *Journal of Propulsion and Power*, 34(5):1140–1151, September 2018.

[40] B. Lad and L. He. Use of an Immersed Mesh for High Resolution Modeling of Film Cooling Flows. *Journal of Turbomachinery*, 135(1):011022, January 2013.

[41] Menter, F. R. Langtry, R. B. Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. *AIAA Journal*, 47:2894–2906, 1994.

[42] D. Y. Lee. Three-Dimensional Unsteady, parallel simulation of a multi-stage turbine with conjugate heat transfer. Master's thesis, University of California, Davis, 2015.

[43] F. Liu, J. Cai, Y. Zhu, H. M. Tsai, and A. S. F. Wong. Calculation of Wing Flutter by a Coupled Fluid-Structure Method. *Journal of Aircraft*, 38(2):334–342, March 2001.

[44] J. S. Liu, M. C. Morris, M. F. Malak, R. M. Mathison, and M. G. Dunn. Comparison of 3D Unsteady Transient Conjugate Heat Transfer Analysis on a High Pressure Cooled Turbine Stage With Experimental Data. In *Volume 5A: Heat Transfer*, page V05AT10A006, Charlotte, North Carolina, USA, June 2017. American Society of Mechanical Engineers.

[45] J.H. Liu, Y.B. Liu, and L. Liu. Film cooling modeling of a turbine vane with multiple configurations of holes. *Case Studies in Thermal Engineering*, 11:71–80, March 2018.

[46] L. Mangani, M. Cerutti, M. Maritano, and M. Spel. Conjugate Heat Transfer Analysis of NASA C3X Film Cooled Vane With an Object-Oriented CFD Code. In *Volume 4: Heat Transfer, Parts A and B*, pages 1805–1814, Glasgow, UK, October 2010. ASMEDC.

[47] Jameson, A., and Alosnso,J. J. McMullen, M. Application of non-linear frequency domain solver to the euler and navier stokes equations. In *40th AIAA Aerospace Sciences Meeting & Exhibit*. AIAA, April 2002.

[48] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32:1598–1605, 1994.

[49] S. Moffatt, W. Ning, Y. Li, R. G. Wells, and Li He. Blade Forced Response Prediction for Industrial Gas Turbines. *Journal of Propulsion and Power*, 21(4):707–714, July 2005.

[50] R. H Ni. A multiple grid scheme for solving the Euler equations. *AIAA*, 20:259–264, 1982.

[51] R. H. Ni, W. Humber, G. Fan, J. P. Clark, R. J. Anthony, and J. J. Johnson. Comparison of Predictions From Conjugate Heat Transfer Analysis of a Film-Cooled Turbine Vane

to Experimental Data. In *Volume 3B: Heat Transfer*, page V03BT11A013, San Antonio, Texas, USA, June 2013. American Society of Mechanical Engineers.

[52] R. H. Ni, W. Humber, G. Fan, D. P. Johnson, J. Downs, J. P. Clark, and P. J. Koch. Conjugate Heat Transfer Analysis for a Film-Cooled Turbine Vane. In *Volume 5: Heat Transfer, Parts A and B*, pages 423–434, Vancouver, British Columbia, Canada, January 2011. ASMEDC.

[53] M. K. Ooten, R. J. Anthony, A. T. Lethander, and J. P. Clark. Unsteady Aerodynamic Interaction in a Closely Coupled Turbine Consistent With Contrarotation. *Journal of Turbomachinery*, 138(6):061004, June 2016.

[54] Ooten, M. K. *Unsteady Aerodynamic Interaction in a Closely Coupled Turbine Consistent withContra-Rotation*. PhD thesis, University of Dayton, Dayton Ohio, August 2014.

[55] P. Patel and G. Zha. Improved Delayed Detached Eddy Simulation of AGARD Wing Flutter with Fully Coupled Fluid-Structure Interaction. In *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, January 2021. American Institute of Aeronautics and Astronautics.

[56] K. Ramakrishnan, D. Gottfried, P. Lawless, and S. Fleeter. Development and Application of an ALE Solver for Turbomachinery Aeromechanics. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2006. AIAA, American Institute of Aeronautics and Astronautics.

[57] R. J. Roark, W. C. Young, and R. G. Budynas. *Roark's formulas for stress and strain*. McGraw-Hill, New York, 7th ed edition, 2002.

[58] M. Sadeghi and F. Liu. Coupled Fluid-Structure Simulation for Turbomachinery Blade

Rows. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2005. American Institute of Aeronautics and Astronautics.

[59] J.L. Steger and R.L. Sorenson. Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *Journal of Computational Physics*, 33(3):405–410, December 1979.

[60] B. Sun. Revisiting the Reynolds-averaged Navier–Stokes equations. *Open Physics*, 19(1):853–862, 2021.

[61] P.R.F. Teixeira and A.M. Awruch. Numerical simulation of fluid–structure interaction using the finite element method. *Computers & Fluids*, 34(2):249–273, February 2005.

[62] Thompson J. F., Thames F. C., and Mastin, W. Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Numer of Arbitrary Two Dimensional Bodies. *Journal of Computational Physics*, 15(3):299–319, 1974.

[63] S. Turek and J. Hron. *Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow.* Number 53 in Lecture notes in computational science and engineering. Springer-Verlag, New York, Berlin, 2006. OCLC: ocm71798417.

[64] D. K. Walters and D. Cokljat. A three-equation eddy-viscosity model for reynolds-averaged navier-stokes simulations of transitional flow. *Journal of Fluids Engineering*, 130, 2008.

[65] M. Wang, H. Zhu, and Y. Xu. Optimization Design of Turbine Blade Cooling Structure Based on Conjugate Heat Transfer. In *Volume 7C: Heat Transfer*, page V07CT13A003,

Virtual, Online, September 2020. American Society of Mechanical Engineers.

[66] R. Wang and X. Yan. Background-Grid Based Mapping Approach to Film Cooling Meshing: Part III — Applications in a Full-Cooled Turbine Vane. In *Volume 7B: Heat Transfer*, page V07BT12A042, Virtual, Online, September 2020. American Society of Mechanical Engineers.

[67] F. M. White. *Viscous Fluid Flow*. McGraw Hill, P-24, Green Park Extension, New Delhi, 2006.

[68] D. C. Wilcox. *Turbulence modeling for CFD*. DCW Industries, Inc, La Cãnada, CA, 1993.

[69] E. C. Jr Yates. Agard standard aeroelastic configurations for dynamic response i-wing 445.6. *AGARD Report*, 765, 1985.

[70] Z. Zhang, A. Hou, W. Tuo, A. Xia, and S. Zhou. Forced Response Analysis of Compressor Blades under Unsteady Inlet Distortion. In *31st AIAA Applied Aerodynamics Conference*, San Diego, CA, June 2013. American Institute of Aeronautics and Astronautics.

[71] J. Zorn. *Development of Unified, Tightly-Coupled Fluid-Structure Interaction Simulation Method*. Doctor of Philosophy, University of California, Davis, December 2018.

[72] J. Zorn and R. L. Davis. Procedure for 2D fluid–structure interaction simulation. *Journal of Algorithms & Computational Technology*, 13:174830261986173, January 2019.