NUMERICAL SIMULATIONS OF HYPERSONIC AEROTHERMOELASTIC PHENOMENA

A Dissertation

by

ROBERT LEE BROWN II

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Paul G. A. Cizmas
Committee Members,	John D. Whitcomb
	Junuthula N. Reddy
	Thomas A. Strganac
Head of Department,	Rodney D. W. Bowersox

August 2016

Major Subject: Aerospace Engineering

ABSTRACT

This dissertation examines the following subjects important to hypersonic aerothermoelastic flows: aerodynamic heating on a double-wedge airfoil, active cooling on a double-wedge airfoil, and aerothermoelastic panel flutter with deformable supports. To facilitate this examination, an aerothermoelastic solver was created by coupling solvers for the structural elasticity and thermal-diffusion equations to a Reynolds-averaged Navier–Stokes (RANS) solver. Validation studies were performed on the aerodynamic, aerothermal, and aerothermoelastic configurations of the solver, which were then used to investigate the aforementioned subjects.

The aerodynamic solver was validated for hypersonic heating simulations by comparison to the Fay-Riddell equation for the peak heat flux on a circular cylinder and by comparison to compressible boundary layer flow for heat flux into a flat plate. A novel investigation was then performed of the heat flux on a double-wedge airfoil, looking at variations of heat flux due to angle of attack, wall temperature, Mach number, and altitude.

The aerothermal solver was validated using experimental and computational data from the hypersonic heating of a spherical protuberance on a flat plate. The solver was then used in a novel study to analyze the effect of active cooling on the steady-state skin temperature of a double-wedge airfoil in hypersonic flow. Active cooling using a piecewise continuous cooling distribution resulted in sufficient temperature reduction, but also results in significant chordwise temperature gradients.

The aerothermoelastic solver was validated using computational data from the

analysis of panel flutter. The solver was then used to examine the effects of deformable structural supports on the flutter dynamics of the panel for the first time. Deformable supports along existing simply-supported boundaries were shown to be ineffective at improving resistance to flutter or buckling. Deformable supports within a panel were shown to increase the flutter resistance threefold and buckling resistance almost fivefold.

To my wife.

ACKNOWLEDGEMENTS

Many thanks to Dr. Cizmas for his years of education and guidance. Thank you also to my committee, whose suggestions resulted in a better end product.

This work has been supported by the Air Force Office of Scientific Research Multidisciplinary University Research Initiative (Grant No. FA9550-09-1-0686). The authors acknowledge the Texas A&M Supercomputing Facility (http://sc.tamu.edu/) for providing computing resources useful in conducting the research reported in this paper.

NOMENCLATURE

Greek

α_k	Coefficients of the Runge-Kutta time-stepping algorithm
α_{ij}	Component of the thermal expansion tensor $\boldsymbol{\alpha}$
$\boldsymbol{\alpha}$	Thermal expansion tensor
γ	Ratio of specific heats, c_p/c_v
δT	Test function for temperature
δu	Test function for displacement
$\Delta(\cdot)$	Change in $(\cdot)$
$\partial\Omega$	Surface of cell
$\partial\Omega_{IJ}$	Surface between the dual cells associated with nodes I and J
$\boldsymbol{\epsilon}$	Strain tensor
ϵ_{ij}	Component of the strain tensor $\boldsymbol{\epsilon}$
κ	Thermal conductivity
λ	Non-dimensional pressure parameter
μ	Dynamic viscosity
μ_e	Air viscosity at boundary layer edge

μ_w	Air viscosity at blunt-body wall
ν	Poisson's ratio
ρ	Fluid density
ρ_e	Air density at boundary layer edge
ρ_m	Plate material density
ρ_w	Air density at blunt-body wall
$\boldsymbol{\sigma}$	Stress tensor
σ_{ij}	Component of the stress tensor $\boldsymbol{\sigma}$
τ_w	Wall shear stress
τ_{ij}	Component of the shear stress tensor $\boldsymbol{\tau}$
ϕ	Radial basis function
Ξ	Generalized force
	Linear polynomial, $\psi(\vec{x}) = b_0 + b_1 x + b_2 y + b_3 z$
^{N}	Shape function associated with node N
ω	Frequency (rad/s)
$\Omega(t)$	Volume of a given cell in the grid
Ω_I	Volume of the dual cell associated with node I

Roman

a	Plate length/width
$\mathbf{a}$	Vector of nodal accelerations
c	Specific heat capacity of solid material
c	Speed of sound
c_p	Specific heat under constant pressure
c_v	Specific heat under constant volume
C_{ijkl}	Tensor of linear elasticity constants
D	Blunt-body diameter
d_{b_i}	Displacement of boundary node i in x, y, or z
d_{IJ}	Distance between nodes I and J
E	Young's modulus
e_0	Stagnation energy
f	Frequency (1/s)
$\mathbf{F}_p$	Vector of forces due to pressure
$\mathbf{F}_v$	Vector of viscous forces
$\mathbf{F}_{\Delta T}$	Thermal load

$\mathbf{F}_{c_{IJ}}$ Approximate convective flux

$\mathbf{F}_c$ Vector of convective fluxes

$\mathbf{F}_{v_{IJ}}$ Approximate viscous flux

g_a Non-dimensional aerodynamic damping parameter

h Plate thickness

h_w Air enthalpy at blunt-body wall

h_{0_∞} Farfield stagnation enthalpy

$(JG)_s$ Stringer torsional rigidity

K Stagnation point velocity gradient at the edge of the boundary layer

K_p Panel effective stiffness (pressure load/displacement)

$\mathbf{K}$ Stiffness matrix

$\mathbb{M}$ Generalized mass

M Mach number

$\mathbf{M}$ Mass matrix

N Number of nodes

$N(I)$ Set of nodes neighboring node I

n_i Component of unit normal vector $\hat{n}$

p Pressure

p_e Pressure at boundary layer edge

Pr Prandtl number

q_{avg} Average heat flux

q_{FR} Peak blunt-body heat flux predicted by Fay–Riddell equation

q_i Component of heat conduction vector $\vec{q}$

$\mathbf{Q}$ Vector of conservative state variables

$\mathbf{Q}^*$ Approximate vector of state variables for pseudo-time stepping

$\mathbf{Q}_I$ Vector of state variables averaged over the dual cell associated with node I

$\mathbf{R}_I$ Residual at node I

$\mathbf{R}^*$ Residual using $\mathbf{Q}^*$

$\mathbf{R}^*_{unsteady}$ Residual using $\mathbf{Q}^*$, including source terms from dual-time stepping scheme

S_{IJ} Surface area of surface between the dual cells associated with nodes I and J

$\hat{t}_{IJ}$ Unit tangent vector of the edge from node I to J

T Temperature

t Time

$\mathbf{T}$ Vector of nodal temperatures

$\vec{u}$ Flow velocity

$\vec{u}$ Vector of structural displacements

$\vec{u}_g$ Grid velocity

$\vec{u}_r$ Flow velocity relative to grid face, $\vec{u}_r = \vec{u} - \vec{u}_g$

U Specified flow velocity

u_i Component of flow velocity vector $\vec{u}$

$\mathbf{u}$ Vector of nodal displacements

V_n Plate upwash velocity

$\mathbf{v}$ Vector of nodal velocities

w Plate out-of-plane deformation

$\vec{x}$ Location (x, y, and z)

$\vec{x}_{b_i}$ Location of boundary node i (x, y, and z)

y^+ Dimensionless wall distance, $\frac{y}{\mu}\sqrt{\tau_w \rho}$

y_1 Distance from surface to first out-of-plane point

Math

$(\cdot)_a$ Relating to the aerodynamic model

$(\cdot)_s$ Relating to the structural model

$(\cdot)_\infty$ Farfield conditions

$\hat{\cdot}$ Unit vector of three components, corresponding to x, y, and z directions

∇ Vector of spatial derivatives

$\partial(\cdot)$ Boundary of $\cdot$

$\vec{\cdot}$ Vector of three components, corresponding to x, y, and z directions

$\{\cdot\}^{\mathbf{T}}$ Vector/matrix transpose

TABLE OF CONTENTS

Page

Page

Page

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

I.1. Problem Statement

Vehicles capable of hypersonic flight have a monopoly on access to space and are capable of the fastest available means of point-to-point transportation on Earth. Until the recent advancements in scramjet engines, hypersonic flight has been solely the domain of rocket-propelled vehicles. As scramjet propulsion continues to improve [1, 2], airbreathing propulsion is set to become a viable option for hypersonic flight, both with single stage to orbit space access and hypersonic cruise vehicles for exceptionally fast point-to-point transportation.

The dream of airbreathing propulsion is to overtake rocket propulsion economically. The economic viability hinges primarily on three aspects of vehicle performance: the weight of the airframe, the specific impulse of the engine, and the reusability of the aircraft (including maintenance costs and turnaround time). The purpose of this research is to investigate the key physics behind the structural requirements of the airframe. The most stringent structural requirements for hypersonic aircraft are due to the phenomena of panel flutter and thermal buckling, especially when the two are coupled. Panel flutter can induce significant fatigue on structural elements of the aircraft, shortening the lifetime of hypersonic vehicles significantly. Furthering the understanding of aerothermoelastic panel flutter will help define the structural requirements, and reduce need for excessive safety margins and thus the overall weight

of structural elements.

Another key element of the weight is the thermal protection system used to protect the vehicle from the high heat loads associated with hypersonic flight. Modern heat load mitigation strategies include hot-structure designs, ablation-based thermal protection systems, and active cooling systems. Hot-structure designs seek to use sufficiently durable materials to allow parts of the structure to reach a near-equilibrium temperature, while insulating parts of the vehicle that need to remain cool [3]. Ablative thermal protection systems are typically used for shorter flights with very high heating rates [4], but are not typically suitable for reusable vehicles with long flight times, as the weight scales with the amount of total heat dissipation. Active cooling promises both sufficiently high cooling rates and large amounts of total heat dissipation [5], but also requires a significant increase in the overall complexity and weight. The benefits of an active cooling system in suppressing panel flutter will be investigated as a possible means of weight reduction.

I.2. Literature Review

As the computational treatment of hypersonic aerothermoelasticity requires knowledge of a broad range of disciplines, this review will give a brief historical perspective of the fields of hypersonics, aeroelasticity, and computational aerodynamics. The aeroelastic and aerothermoelastic study of panel flutter will be covered in more depth. Finally, the recent work in computational hypersonic aerothermoelasticity will be reviewed.

I.2.1. Early Work in Hypersonics

The majority of early work in hypersonics was devoted to building up a significant body of experimental data [6, 7] and building new models to explain hypersonic phenomena [8]. Additionally, a number of mathematical and computational methods were developed for the simulation of hypersonic flow over blunt bodies [9, 10] and slender bodies with blunted noses [11, 12]. Understanding of basic hypersonic phenomena enabled the design of early spacecraft and missiles.

I.2.2. Early Work in Aeroelasticity

The study of aeroelasticity stemmed from the need to avoid deleterious phenomena such as flutter. While several studies [13, 14] investigated various aspects of aeroelasticity, the most pivotal early work was that of Theodorsen [15], whose model is still in use today. Von Karman applied this theory to gusts and other unsteady motion [16]. Goland applied these theories to the flutter of a cantilever wing [17]. These studies increased the basic understanding of aeroelastic phenomena and helped designers produce lightweight, safe aircraft.

I.2.3. Computational Fluid Dynamics

While the exact definition and starting point of computational fluid dynamics can be argued, it clearly has its roots in the arbitrary division of a wing surface or fluid domain. Vortex-lattice [18, 19] and doublet-lattice [20] methods were some of the earliest to be able to determine the characteristics of arbitrary wings. As computers improved, the method of characteristics [21] and finite difference formulas

[22] extended solvable domains to transonic and supersonic flowfields.

I.2.4. Computational Aeroelasticity

In order to use computational fluid dynamics for aeroelastic simulations, the movement of the solid domain must be accurately represented in the fluid domain. One theory that is commonly used today is the Geometric Conservation Law (GCL) first described by [23]. Over the last two decades, a number of papers have been written by Farhat, Mavriplis and others, evaluating and extending the GCL to a large number of time discretization schemes [24, 25, 26, 27, 28]. This remains the basis of accurately computing the effect of moving boundaries for computational fluid dynamics with a finite-volume framework.

I.2.5. Panel Flutter

Panel flutter is a concern for any vehicles "using a lifting-body concept that will operate at supersonic/hypersonic Mach numbers" [29]. This is primarily due to the combined restrictions of low structural weight as required by vehicle design and high panel stiffness required to prevent flutter. The analysis of panel flutter allows for the reduction of structural weight by eliminating the need for additional safety factor.

Panel flutter has been analyzed using aerodynamics varying from piston theory [30, 31] and full linearized potential theory [32, 33] to Navier–Stokes aerodynamics [34, 35] and surrogate models [36, 37]. The simpler theories can be augmented with boundary layers [38] or approximate heating [37], depending on the case being studied.

A large variety of structural geometries have also been studied. While rectangular or two-dimensional infinite panels dominated the majority of early work [33, 38, 39], triangular panels [40, 31], cylindrical shells [41, 42], and panels with streamwise or spanwise curvature [43, 44] have all been extensively studied.

Because panel flutter occurs at supersonic and hypersonic speeds, the study of heated panels is of interest. The primary effect of heating a panel is to lower the flutter threshold [33]. For sufficiently hot plates, thermal buckling [45] and snap-through [46, 47] may also occur. These effects have primarily been studying by applying an arbitrary temperature field to the structure, though more recent work has investigated the effects of using aerodynamic heating [48, 49, 50].

I.2.6. Hypersonic Aerothermoelasticity

Much of the work in hypersonic aerothermoelasticity could also be mentioned in discussions on panel flutter, as there is a large overlap in the two disciplines. In general, however, hypersonic aerothermoelasticity also covers the investigation of blunt leading edges [51], aerodynamic decelerators, scramjet engines, and full aircraft simulations [52].

Simulation of aerothermoelastic phenomena typically requires some sort of finite element method for the structure, though modal methods have also been used with success. The aerodynamic heating rates have been successfully determined using Navier–Stokes aerodynamics [53], surrogate methods [36], and approximate heating techniques such as the Eckert reference enthalpy method [54, 55]. The aerodynamic loads have been calculated using theories ranging from piston theory to Navier–

Stokes.

A more thorough overview of recent aerothermoelastic research can be found in [56].

I.3. Research Problem

The study of panel flutter in the context of a single panel has been thoroughly explored, though not yet in the context of a full wing analysis. However, these panels assume relatively simple boundary conditions, such as simply supported, free, or clamped. In reality, the deformations of the panel are determined by the complete aeroelastic response of the entire aircraft. For this reason, the present work attempts to systematically analyze the panel flutter of increasingly complex structures, which can eventually aid in the aerothermoelastic analysis and design of complete wings and vehicles.

Because of the importance of the increase in temperature for panel flutter response, understanding the aerodynamic heating of the structure is critical for design of hypersonic vehicles. This work will look at the steady-state heat transfer problem with aerodynamic heating on the exterior surface of the structure, and cooling on the interior surface of the structure. The investigation of active cooling over a complete airfoil in hypersonic flow is novel.

The goal of the present research is to further the understanding of aerothermoelastic phenomena critical to the design of hypersonic cruise vehicles. The phenomena of interest are the aeroelastic stability of panels supported by an underlying deformable structure, the effect of realistic thermal loading on the aeroelastic sta-

bility, and the effectiveness of heat mitigation strategies in suppressing aeroelastic instabilities.

The present research has been accomplished through the following research objectives:

- Creation and validation of a coupled solver incorporating a Reynolds-averaged Navier–Stokes aerodynamic solver, a nonlinear elastic solver, and a thermal diffusion solver
- Deformable supports are added to panels subject to hypersonic flow, and the aerothermoelastic response is characterized by determining flutter thresholds and investigating post-flutter dynamics
- Investigation of active cooling as a means to expand the flight envelope for sustained hypersonic flight, specifically by increasing the expected panel flutter boundary and reducing panel susceptibility to thermal buckling

I.4. Original Contributions of Present Work

The novelty of the present work is primarily in the application of existing computational methods to new problems in hypersonic aerothermoelasticity. Specifically, two aspects of hypersonic flight have been analyzed.

First, the effect of active cooling on the steady-state surface temperature of a double-wedge airfoil in hypersonic flight was investigated, and several variations of the distribution of cooling were taken into account, specifically augmenting the cooling near the leading edge, where the heating rates were highest. To the author's

knowledge, this is a significantly more complex geometry than has been studied for long-term heating with active cooling. Correlations for a given cooling methodology were obtained by testing the same cooling profile with different average cooling rates.

Second, the aerothermoelastic response of panels with deformable substructures was analyzed for the first time. Panel flutter thresholds were found for panels supported by stringers in two configurations, and post-flutter dynamics were also investigated. The improvement in structural characteristics, both the resistance to buckling and to flutter was quantified as the thickness of the stringers was increased. Additionally, an asymmetric mode was discovered for high temperature post-flutter vibration of simply-supported panels.

I.5. Outline of Dissertation

This work is organized as follows: in Chapter II, the aerodynamic, structural elastic, and thermal-diffusion models used in the dissertation are presented. Chapter III contains the numerical methods used to discretize and solve the models are shown. In Chapter IV, the hypersonic aerodynamic solver is validated and novel studies of a hypersonic double-wedge airfoil using only the aerodynamic solver are performed. Chapter V describes the validation of the coupled aerothermal solver and presents the aerothermal studies of a hypersonic double-wedge airfoil with internal cooling. Chapter VI validates the aerothermoelastic solver and describes the aerothermoelastic studies of panels with supporting stringers. Finally, Chapter VII presents both conclusions from the novel work performed in this dissertation and provides suggestions for future work.

CHAPTER II

PHYSICAL MODELS

This chapter presents the governing equations of the aerodynamic, elastic, and thermal-diffusion models and the procedure used to couple them. Each of these three models were governed by some combination of the conservation of mass, conservation of momentum, and conservation of energy, along with relevant constitutive relationships. The three models were coupled using a partitioned analysis procedure [57], where the system was spatially decomposed into partitions.

The first section of this chapter describes the aerodynamics models. The second section presents the structural linear elasticity model. This is followed by the thermal-diffusion model. The final section gives the equations used to couple the three models.

II.1. Aerodynamics Models

Two aerodynamics models were used in this research. The first aerodynamic model is governed by the Reynolds-Averaged Navier–Stokes (RANS) equations and was used for the simulation of viscous heating and for aerothermal analyses. Additionally, the RANS solver, neglecting viscous effects, was used for one of the aeroelasticity studies. This is referenced as the Euler solver. The second aerodynamic model was piston theory, which was used for aeroelasticity studies.

II.1.1. The Reynolds-Averaged Navier–Stokes Model

The RANS equations were applied to a moving control volume through the Reynolds Transport Theorem using the conservative state vector $\mathbf{Q} = \{\rho,\, \rho u_x,\, \rho u_y,\, \rho u_z,\, \rho e_0\}^{\mathbf{T}}$, where ρ is the density of the fluid, u_ℓ is the ℓ-component of the velocity, and e_0 is the mass-specific stagnation energy. Additionally, volume forces were neglected and the surface forces were split into pressure and viscous fluxes, $\mathbf{F}_p$ and $\mathbf{F}_v$, respectively.

$$\frac{d}{dt}\int_{\Omega(t)} \mathbf{Q}\; dV = -\oint_{\partial\Omega(t)} \left[\mathbf{F}_p - \mathbf{F}_v + \mathbf{Q}\left(\vec{u}_r \cdot \hat{n}\right)\right] dS, \tag{2.1}$$

where $\vec{u}_r$ is the flow velocity relative to the grid face and $\Omega(t)$ is the volume of a cell of the grid (See Appendix A). $\mathbf{F}_p$ and $\mathbf{F}_v$ are defined as follows, where duplicated indices imply summation over the possible values of the index:

$$\mathbf{F}_p = \begin{Bmatrix} 0 \\ n_1 p \\ n_2 p \\ n_3 p \\ n_i u_i p \end{Bmatrix}, \qquad \mathbf{F}_v = \begin{Bmatrix} 0 \\ n_i \tau_{1i} \\ n_i \tau_{2i} \\ n_i \tau_{3i} \\ -n_i q_i + n_j u_i \tau_{ij} \end{Bmatrix}, \qquad i, j = 1, 2, 3, \tag{2.2}$$

where n_i is the ith-component of the unit vector normal to the surface, and q_i is the ith-component of the heat transfer rate per unit area via conduction in the fluid given by Fourier's Law:

$$q_i = -\kappa\; \partial T / \partial x_i, \qquad\qquad i = x, y, z \tag{2.3}$$

where κ is the thermal conductivity of the fluid. The pressure p of the fluid particle is related to the density ρ, temperature T, and gas constant R through the equation of state: $p = \rho RT$. τ_{ij} is the component of the viscous stress tensor defined as in [58, pp.356-357]

$$\tau_{ij} = \begin{cases} \mu\left(\dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i}\right), & \text{if } i \neq j \\ 2\mu\left(\dfrac{\partial u_i}{\partial x_j} - \dfrac{1}{3}\dfrac{\partial u_k}{\partial x_k}\right), & \text{if } i = j \end{cases}, \qquad k = 1, 2, 3, \tag{2.4}$$

where μ is the dynamic viscosity of the fluid.

The effects of turbulence were modeled by the two-equation eddy-viscosity shear stress transport model [59].

II.1.2. Piston Theory

Piston theory [30, 31] was used in this research to calculate the aerodynamic forces for aeroelastic analyses. Specifically, first-order and third-order piston theory were used, as presented in [32, 33]. The first-order piston theory equation used in this study was simply:

$$\Delta p = \frac{\rho U_\infty^2}{\sqrt{M_\infty^2 - 1}}\left(\frac{\partial w}{\partial x} + \frac{1}{U_\infty}\frac{\partial w}{\partial t}\frac{M_\infty^2 - 2}{M_\infty^2 - 1}\right), \tag{2.5}$$

where U_∞ is the velocity of the fluid, M_∞ is the fluid Mach number, w is the panel displacement into the fluid, and the upwash velocity V_n can be written as follows:

$$V_n = U_\infty\frac{\partial w}{\partial x} + \frac{\partial w}{\partial t}\frac{M_\infty^2 - 2}{M_\infty^2 - 1} \tag{2.6}$$

At higher Mach numbers, typically at Mach 5 or above, $\sqrt{M_\infty^2 - 1} \rightarrow M$ and $\frac{M_\infty^2 - 2}{M_\infty^2 - 1} \rightarrow 1$. Additionally, as $\frac{V_n}{U_\infty}M_\infty$ increases, the assumptions used for linearization

of full-order piston theory break down (see Appendix B), and it becomes necessary to expand to third-order piston theory, which is stated as:

$$\Delta p = \frac{\rho U_\infty^2}{M_\infty}\frac{V_n}{U_\infty}\left(1 + \frac{\gamma+1}{4}\frac{V_n}{U_\infty}M_\infty + \frac{\gamma+1}{12}\left(\frac{V_n}{U_\infty}M_\infty\right)^2\right) \tag{2.7}$$

For panel flutter studies, the pressure difference is typically separated into a non-dimensional pressure parameter, λ, and a non-dimensional damping parameter, g_a. These are defined as:

$$\lambda = \frac{\rho U_\infty^2}{\sqrt{M_\infty^2 - 1}}\frac{a^3}{h^3}\frac{12(1-\nu^2)}{E}, \quad g_a = \frac{1}{\pi^2}\frac{\rho U_\infty}{\sqrt{M_\infty^2-1}}\frac{M_\infty^2 - 2}{M_\infty^2 - 1}\frac{a^2}{h^2}\sqrt{\frac{12(1-\nu^2)}{\rho_m E}}, \tag{2.8}$$

where a is the panel length, h is the panel thickness, ν is the Poisson's ratio, E is the Young's modulus, and ρ_m is the density of the panel material.

The pressure parameter λ is an accurate representation of the pressure forces when the upwash velocity V_n is significantly less than the speed of sound, c_∞. This is true for small deflections and deflection frequencies. For initial flutter analyses, when panel deflections are small, any order of piston theory should yield the same results. Differences between first order and higher-order piston theory models are only seen in post-flutter analysis.

II.2. Linear Elasticity Model

The elastic behavior of the structure was modeled based on conservation of momentum, including the effects of thermal expansion. The model follows the equation [58, pp.232]:

$$\rho_m \frac{\partial^2 \vec{u}}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma}, \tag{2.9}$$

where $\mathbf{u}$ is the vector of displacements and $\boldsymbol{\sigma}$ is the stress tensor, defined using Einstein notation [58, pp.3-10] as

$$\sigma_{ij} = C_{ijkl}(\epsilon_{kl} - \alpha_{kl}\Delta T), \tag{2.10}$$

where C is the fourth order tensor describing the material constitutive relationship, ϵ is the infinitesimal strain tensor, α is the coefficient of thermal expansion, and ΔT is the temperature difference from the stress-free state.

To close the model, the strain tensor can be related to the displacements by

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{2.11}$$

II.3. Thermal Diffusion Model

The unsteady diffusion of heat through the structure was modeled with the energy conservation equation, expressed in terms of heat conduction as follows:

$$-\frac{\partial(\rho_m c T)}{\partial t} = \frac{\partial q_i}{\partial x_i}, \tag{2.12}$$

where T is the temperature, and c is the specific heat capacity of the solid. The components of the heat flux vector q_i are defined by (2.3), where κ is the thermal conductivity of the elastic material.

II.4. Coupling Equations

The aerothermoelastic solver is coupled by enforcing consistency of the displacement, traction, temperature, and heat flux on the shared points of the solvers. There are three sets of equations needed to fully couple the aerothermoelastic solver. First, the aerodynamic solver and the structural elastic solver are coupled by equating the

displacement and traction at every point, $\vec{x}$, of the shared surface $\partial\Omega$. That is:

$$\boldsymbol{\sigma}_s \cdot \hat{n} = p_a \hat{n} + \boldsymbol{\tau}_a \cdot \hat{n} \qquad \forall \vec{x} \in \partial\Omega \tag{2.13a}$$

$$\mathbf{u}_s = \mathbf{u}_a \qquad \forall \vec{x} \in \partial\Omega, \tag{2.13b}$$

where $\boldsymbol{\sigma}_s$ is the stress tensor for the structure on the surface, p_a is the aerodynamic pressure on the surface, $\boldsymbol{\tau}_a$ is the viscous stress tensor for the aerodynamic solver, and $\mathbf{u}_s$ and $\mathbf{u}_a$ are the displacements on the surface for the structural and aerodynamic solvers, respectively.

Next, the aerodynamic solver and the thermal-diffusion solver are coupled by equating the temperature and heat flux on the shared surface.

$$\vec{q}_s \cdot \hat{n} = \vec{q}_a \cdot \hat{n} \qquad \forall \vec{x} \in \partial\Omega \tag{2.14a}$$

$$T_s = T_a \qquad \forall \vec{x} \in \partial\Omega, \tag{2.14b}$$

where q_s and q_a are the structural and aerodynamic heat fluxes into the boundary, respectively, and T_s and T_a are the temperatures of the structure and air at the undeformed shared boundary.

Finally, the structural elastic solver is coupled to the thermal-diffusion solver by requiring that the temperature at every location in the structure is equivalent for both solvers:

$$T_e = T_s \qquad \forall \vec{x} \in \Omega_s, \tag{2.15}$$

where T_e is the temperature in the structural elastic solver, and Ω_s represents the entire structural domain.

CHAPTER III

NUMERICAL METHODS

This chapter briefly presents the numerical methods used to calculate the flow around the airfoil, the temperature distribution within the structure, and the deformation of the structure. The finite volume methodology utilized to solve the RANS equations (2.1) is presented first, followed by the finite element approach used to solve both the thermal-diffusion equation (2.12) and the structural elasticity equation (2.9). This chapter ends with a presentation of the coupling algorithm between the aerodynamic, thermal-diffusion and structural elasticity models.

III.1. Reynolds-Averaged Navier–Stokes Solver

This section presents the spatial and temporal discretization of the RANS equations. The methodology presented in previous works [60, 61] was modified to allow for accurate simulation of hypersonic flow and moving domains. It was also modified to allow coupling to both the thermal-diffusion solver and the structural elasticity solver.

III.1.1. Spatial Discretization

A finite volume method was used to solve the RANS equations. The governing equations were discretized using an unstructured grid with dual cells as control volumes [60]. The median dual-mesh was adopted herein because of its flexibility of handling unstructured mixed meshes [62]. The cell-averaged variables were stored at

the nodes of the grid, that is, the vertices of the cells.

Equation (2.1) is rewritten for a node I as

$$\frac{d}{dt}\left(\Omega_I \mathbf{Q}_I\right) = -\sum_{J \in N(I)} \oint_{\partial\Omega_{IJ}} \left[\mathbf{F}_p + \mathbf{F}_v - \mathbf{Q}\left(\vec{u}_r \cdot \hat{n}\right)\right] dS \tag{3.1}$$

where $\mathbf{Q}_I$ denotes the vector of state variables averaged over the dual-mesh cell volume Ω_I. Node J belongs to $N(I)$, the set of vertices neighboring node I, and $\partial\Omega_{IJ}$ is the surface between nodes I and J.

The surface integral of the convective fluxes was approximated by

$$\oint_{\partial\Omega_{IJ}} \mathbf{F}_c \, dS = \oint_{\partial\Omega_{IJ}} \left(\mathbf{F}_p - \mathbf{Q}\left(\vec{u}_r \cdot \hat{n}\right)\right) dS \approx \mathbf{F}_{c_{IJ}} S_{IJ} \tag{3.2}$$

where $\mathbf{F}_{c_{IJ}}$ is the approximate convective flux normal to the dual-mesh cell face and S_{IJ} is the surface area of the face $\partial\Omega_{IJ}$.

The convective flux $\mathbf{F}_{c_{IJ}}$ was evaluated using the RoeM scheme [63]. The RoeM scheme was chosen because it conserves stagnation enthalpy and remedies problems with shock instabilities, which are both necessary properties for accurate hypersonic heating calculations.

The surface integral of the viscous fluxes was approximated by

$$\oint_{\partial\Omega_{IJ}} \mathbf{F}_v \, dS \approx \mathbf{F}_{v_{IJ}} S_{IJ}, \tag{3.3}$$

where $F_{v_{IJ}}$ is the approximate average viscous flux over the surface $\partial\Omega_{IJ}$. $F_{v_{IJ}}$ was calculated using an arithmetic average of the state variables and gradients at nodes I and J. These averages were used to evaluate (2.2), (2.3), and (2.4). At each node, the least-squares method was used to compute the gradients [64]. Along each edge,

the edge-direction gradient was replaced by the value calculated from the two nodes:

$$\nabla \mathbf{Q} \cdot \hat{t}_{IJ} \equiv \frac{\mathbf{Q}_J - \mathbf{Q}_I}{d_{IJ}}, \tag{3.4}$$

where t_{IJ} is the unit vector tangent to the edge connecting nodes I and J and d_{IJ} is the distance between nodes I and J. This method increases the accuracy of the solution by preventing decoupling of the viscous fluxes along the edge [65].

Combining (3.2) and (3.3) allows us to rewrite (3.1) in terms of a residual $\mathbf{R}_I$:

$$\mathbf{R}_I = -\frac{\partial}{\partial t}\left(\Omega \mathbf{Q}_I\right) = \sum_{J \in N(I)} \left(\mathbf{F}_{c_{IJ}} + \mathbf{F}_{v_{IJ}}\right) S_{IJ} \tag{3.5}$$

The gradients calculated through the least squares method were used to obtain second-order spatial accuracy. A linear reconstruction of the solution was defined for a dual-cell defined by node I by the solution and gradient at that node:

$$\mathbf{Q}\left(\vec{x}\right) = \mathbf{Q}_I + \nabla \mathbf{Q}_I \cdot \left(\vec{x} - \vec{x}_I\right) \tag{3.6}$$

For each flux calculation, the edge midpoint was used as the quadrature point for the integration over the face. The values at the quadrature points were calculated using (3.6).

III.1.2. Temporal Discretization

The RANS solver was discretized using a dual-time stepping approach [66, p. 212]. The solution at time t_{n+1} was calculated using existing solutions at times t_n and t_{n-1} through a second order backwards difference formula:

$$\frac{3}{2\Delta t}\left(\Omega \mathbf{Q}\right)^{n+1} - \frac{2}{\Delta t}\left(\Omega \mathbf{Q}\right)^{n} + \frac{1}{2\Delta t}\left(\Omega \mathbf{Q}\right)^{n-1} = -\mathbf{R}^{n+1}. \tag{3.7}$$

Since $\mathbf{R}^{n+1}$ is dependent on the solution at time t_{n+1}, an approximation $\mathbf{Q}^*$ was made to the solution $\mathbf{Q}^{n+1}$, which was iteratively improved through pseudo-time stepping as follows:

$$\Omega^{n+1}\frac{\partial}{\partial t^*}\mathbf{Q}^* = -\mathbf{R}^*_{unsteady} = -\mathbf{R}^* + \frac{1}{2\Delta t}\left(3\Omega^{n+1}\mathbf{Q}^* - 4\Omega^n\mathbf{Q}^n + \Omega^{n-1}\mathbf{Q}^{n-1}\right) \quad (3.8)$$

As $t^* \to \infty$, $\mathbf{Q}^*$ converges to $\mathbf{Q}^{n+1}$. Time-stepping in pseudo-time was accomplished through a low-storage n-stage Runge–Kutta time-integration scheme. Integration from t^*_n to time t^*_{n+1} is shown below:

$$\begin{aligned} \mathbf{Q}^*_{(0)} &= \mathbf{Q}^*_n \\ \mathbf{Q}^*_{(k)} &= \mathbf{Q}^*_{(0)} - \alpha_k \Delta t \mathbf{R}^{*,(k-1)}_{unsteady}, \qquad 1 \le k \le m \\ \mathbf{Q}^*_{n+1} &= \mathbf{Q}^*_{(m)} \end{aligned} \quad (3.9)$$

where α_k are the coefficients of the Runge-Kutta matrix [66, p. 182]. The solver was marched forward in pseudo-time until the norm $\mathbf{R}^*_{unsteady}$ was reduced below a predetermined tolerance, meaning that a steady state solution was reached. Convergence to steady state in pseudo-time was accelerated through several means: local time stepping, preconditioning, and implicit residual smoothing. Local time stepping updates the solution at each vertex based on the local CFL stability criterion [67], rather than limiting the time step to the smallest in the domain. Preconditioning modifies the eigenvalues of the convective Jacobian in regions of slow flow to reduce the difference between the velocities of the acoustic and convective waves [68]. The preconditioning methodology in the present work follows that of [69]. Implicit residual smoothing increases the stability bounds of the time stepping algorithm by adding artificial dissipation based on $\mathbf{Q}^*_{n+1} - \mathbf{Q}^*_n$ [70]. This does not affect the steady

state solution since $\mathbf{Q}^*_{n+1} - \mathbf{Q}^*_n$ approaches zero as the solution converges.

III.1.3. Parallelization Algorithm

The RANS flow solver was parallelized to reduce the turnaround time. A domain decomposition approach was used to divide the workload between processes. Communication was accomplished through an overlap at the domain boundaries. The Message Passing Interface (MPI) paradigm was used to allow scaling to large numbers of processes. The current procedure follows that of [61], except that arbitrary groupings of nodes were allowed and the domain was not split only in layers.

The parallelization procedure was comprised of the following steps:

- Splitting the global mesh into local meshes for each process
- Preprocessing to determine non-local nodes in mesh
- Initializing the RANS solver by creating a list of communication nodes
- Communicating during each time step

III.1.3.1. Mesh Splitting Step

First, the nodes of the global mesh were distributed among n local meshes, where n is the number of processes used to parallelize the flow solver. The distribution was accomplished by sorting the nodes in three dimensions and recursively splitting the global mesh based on the rank of the nodes.

For example, consider a mesh split into eight parts (twice each in x, y, and z). First, the mesh would have been split in x, resulting in two sets of nodes. The first

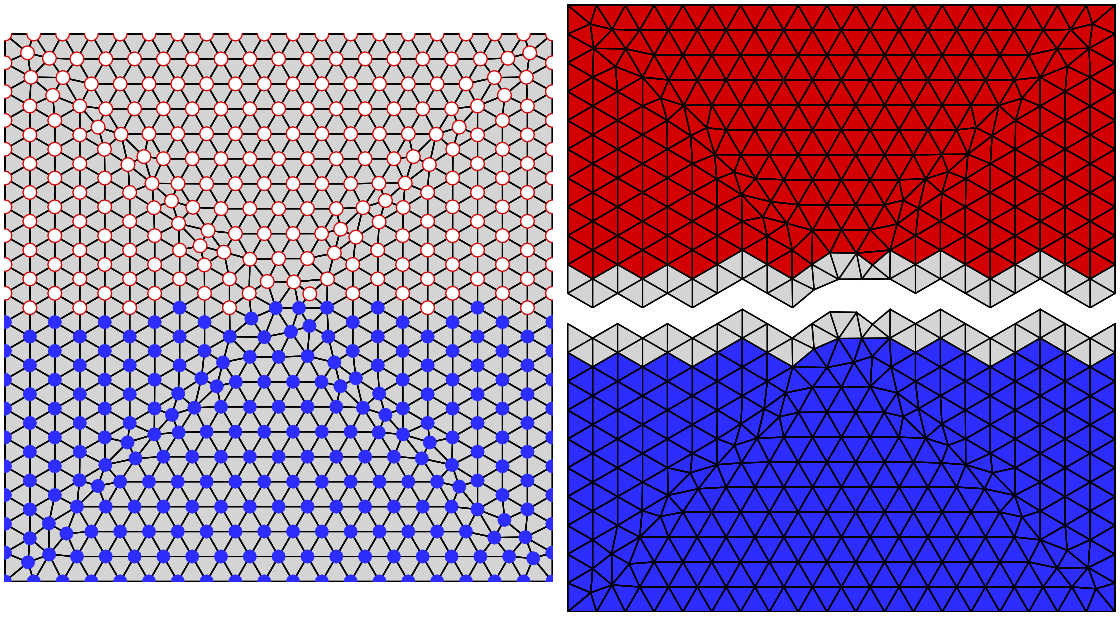


Figure III.1: Visualization of mesh parallelization procedure. Left: Complete mesh. Filled blue nodes are present on the first process, and empty red nodes are present on the second process. Right: Mesh split for two processors. Blue cells are present only on the first process, red cells only on the second, and grey cells are present on both.

set of nodes would have had all of the nodes with x smaller than the median x-value, and the second set would have had the all the nodes with x larger than the median. Both of these sets were then split separately in y, possibly with different median y-values, and all four of those were finally split in z. This yielded local meshes which were node-balanced within a single node, with reasonably smooth communication boundaries. A simple mesh split once in the y-direction is shown in Figure III.1.

III.1.3.2. Preprocessing Step

In the preprocessing step, the cell, face, and edge information for each process was determined. For each local node on a process, the cells, faces, and edges connected to that node were necessary to fully define the local processor mesh. These criteria meant that along the process boundary, there was always an overlap with the neighboring processes. These overlapping cells are shown in gray in Figure III.1.

The local mesh for each process contained some number of non-local nodes. For each of those nodes, the process number on which it was local and the local number on that process was determined. This information was used in the initialization step to create the final list nodes describing the communication step.

III.1.3.3. Initialization Step

The initialization step was performed once at start-up for the RANS solver. Each process created two lists: 1) a list of nodes for which data was needed from other processes (receive nodes) and 2) a list of nodes from which other processes needed data (send nodes). The receive nodes were exactly the list of nodes created by the preprocessing step. In order to generate the list of send nodes, each process performed the following steps:

1. The receive nodes were sorted and counted based on the process for which those nodes were local (the sending process).

2. The node count for each sending process was sent to that process.

3. The node count was received by each process.

4. The node list for each sending process with at least one send node was sent to that process.

5. The node list was received by each process which had at least one send node.

The send node list was created by compiling all of the node lists sent from the receiving processes. Depending on how the original mesh was split, the same node could have been required by several processes (i.e. it could have been non-local on more than one process). However, each node was local on a single process, meaning that only one process would send the information of any given node.

III.1.3.4. Communication Step

The communication step occurred each time the gradients or state variables at each node were updated. In this step, each process sent values for its send nodes to the appropriate receiving processes, and received values for its receive nodes from the appropriate sending processes. The values of the state variables for each communication node were updated before any further computation.

The details of the communication were determined by the available implementation of MPI. For older setups, when only MPI 2.0 is available, the parallelization is accomplished through a series of `MPI_ISEND` and `MPI_IRECV` commands. Each process sent an individual message to every process that requested information from it. Likewise, each process received an individual message from every process from which it requested information. This meant that for a simulation with n processes, there were up to $2n(n-1)$ individual communications between the processes.

When MPI 3.0 was available, a new feature was used, known as non-blocking

collective message passing [71]. The procedure which implements this feature, `MPI_INEIGHBOR_ALLTOALLV`, uses a distributed graph topology to communicate all requested nodes in a single collective communication. The actual amount of data communication was no less than that in the mass of individual communications, but the complexity was managed by the MPI commands. The primary benefit of using collective communications was that they can be optimized for the architecture of the computer running the simulation.

III.2. Finite Element Solver for Structural Elasticity and Thermal-Diffusion

The finite element method was employed to convert the boundary-value problems posed in Sections II.2 and II.3 into sets of ordinary differential equations. Quadratic 20-node serendipity elements were used to discretize the structure. For each element, the displacement u and temperature T were given by a summation of shape functions ψ:

$$\vec{u}(\vec{x}) = \sum_{N=1}^{20} \vec{u}^N \psi^N(\vec{x}) \qquad\qquad T(\vec{x}) = \sum_{N=1}^{20} T^N \psi^N(\vec{x}), \tag{3.10}$$

where the $\vec{u}^N$ and T^N are the nodal values at node N of the displacement and temperature, respectively. ψ is defined as a second order polynomial in x, y, and z such that:

$$\psi^I(\vec{x}^J) = \begin{cases} 1, & I = J \\ 0, & I \neq J \end{cases}, \tag{3.11}$$

where $\vec{x}^J$ is the location of node J.

III.2.1. Discretization of Structural Elasticity Solver

Equation (2.9) was formulated in the weak form using a test function δu. This is written using Einstein notation as follows:

$$\int_\Omega \delta u \left(\rho_m \frac{\partial^2 u_i}{\partial t^2}\right) dV = \int_\Omega \delta u \left(\frac{\partial}{\partial x_j}\sigma_{ij}\right) dV, \qquad i = 1, 2, 3 \tag{3.12}$$

Using integration by parts and Gauss's divergence theorem, the weak form is simplified to:

$$\int_\Omega \delta u \left(\rho_m \frac{\partial^2 u_i}{\partial t^2}\right) dV = \int_{\partial\Omega} (\delta u \sigma_{ij} n_j)\, dS - \int_\Omega \left(\frac{\partial \delta u}{\partial x_j}\sigma_{ij}\right) dV, \tag{3.13}$$

where n_j is the j component of the unit vector $\hat{n}$, which is normal to the boundary.

In order to discretize the structural elasticity equation, u and δu were replaced in (3.13) by shape functions based on (3.10):

$$\begin{aligned}\rho_m \sum_{n=1}^{20} \left[\int_\Omega \psi^M \psi^N dV \frac{\partial^2 u_i^N}{\partial t^2}\right] &= \int_{\partial\Omega} \left(\psi^M \sigma_{ij} n_j\right) dS \\ &\quad - \sum_{n=1}^{20} \left[\frac{C_{ijkl}}{2}\left(\int_\Omega \frac{\partial \psi^M}{\partial x_j}\frac{\partial \psi^N}{\partial x_l} dV u_k^N + \int_\Omega \frac{\partial \psi^M}{\partial x_j}\frac{\partial \psi^N}{\partial x_k} dV u_l^N\right)\right] \\ &\quad + \sum_{n=1}^{20} \left[C_{ijkl}\alpha_{kl}\int_\Omega \frac{\partial \psi^M}{\partial x_j}\psi^N dV \Delta T^N\right],\end{aligned} \tag{3.14}$$

The integrals of the shape function products were evaluated in a preprocessing step, and (3.14) was simplified to the following:

$$\mathbf{M}_e \frac{\partial^2 \mathbf{u}_e}{\partial t^2} = \int_{\partial\Omega} \left(\psi^M \vec{t}\right) dS - \mathbf{K}_e \mathbf{u}_e + \mathbf{F}_{\Delta T, e}, \tag{3.15}$$

where $\mathbf{M}_e$ is the elemental mass matrix, $\vec{t} = \boldsymbol{\sigma} \cdot \hat{n}$ is the traction on the surface of the element, $\mathbf{K}_e$ is the elemental stiffness matrix, $\mathbf{F}_{\Delta T, e}$ is the thermal load on the element, $\mathbf{u_e}$ is the vector of nodal displacements for the element, and $\int_{\partial\Omega} \left(\psi^M \vec{t}\right) dS$

is the boundary traction term. Summation of (3.15) over all elements (also known as assembly) accounts for each of the boundary terms on the interior, while the remaining terms are determined based on the stresses applied to the boundaries of the structure.

Assembly of the elemental equations yields the semi-discretized global structural elasticity equation:

$$\mathbf{M}\frac{\partial^2 \mathbf{u}}{\partial t^2} = -\mathbf{K}\mathbf{u} + \mathbf{F}_{\Delta T} + \mathbf{F}_{bound}, \tag{3.16}$$

where $\mathbf{M}$ is the global mass matrix, $\mathbf{K}$ is the global stiffness matrix, $\mathbf{F}_{\Delta T}$ is the thermal load vector, $\mathbf{u}$ is the vector of nodal displacements, and $\mathbf{F}_{bound}$ is the boundary load vector.

The temporal discretization was accomplished through the Newmark-beta method. Representing the acceleration, velocity, and displacement at time t_n as $\mathbf{a}^n$, $\mathbf{v}^n$, and $\mathbf{u}^n$, respectively, the Newmark-beta method defines the following [72]:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\ \mathbf{v}^n + \frac{\Delta t^2}{2}\mathbf{a}^n + \beta \Delta t^2 \left(\mathbf{a}^{n+1} - \mathbf{a}^n\right) \tag{3.17a}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t\ \mathbf{a}^n + \gamma \Delta t \left(\mathbf{a}^{n+1} - \mathbf{a}^n\right), \tag{3.17b}$$

where Δt is the time step. γ and β are parameters of the Newmark-beta method, and are set to 0.5 and 0.25, respectively to result in a second-order constant acceleration scheme.

Substituting (3.17) into (3.16) results in the following time-stepping scheme:

$$\left(\mathbf{K} + \frac{1}{\beta \Delta t^2}\mathbf{M}\right)\mathbf{u}^{n+1} = \mathbf{F}_{\Delta T}^{n+1} + \mathbf{F}_{bound}^{n+1} + \mathbf{M}\left(\frac{1}{\beta \Delta t^2}\mathbf{u}^n + \frac{1}{\beta \Delta t}\mathbf{v}^n + \left(\frac{1}{2\beta} - 1\right)\mathbf{a}^n\right) \tag{3.18}$$

Equation (3.18) is solved for $\mathbf{u}^{n+1}$ through LU decomposition.

III.2.2. Discretization of Thermal-Diffusion Solver

The weak form of (2.12) was generated by multiplying it by a test function δT and integrating over a volume Ω. The weak form is given as follows:

$$\int_{\Omega} -\delta T \frac{\partial(\rho_m c T)}{\partial t} dV = \int_{\Omega} \delta T \left(\frac{\partial}{\partial x_i} q_i \right) dV \tag{3.19}$$

Application of integration by parts and Gauss's divergence theorem results in:

$$\int_{\Omega} -\delta T \frac{\partial(\rho_m c T)}{\partial t} dV = \int_{\partial\Omega} \delta T q_i n_i \; dS - \int_{\Omega} q_i \left(\frac{\partial}{\partial x_i} \delta T \right) dV \tag{3.20}$$

Similar to the discretization of the structural elasticity equation, (3.10) is used to replace T and δT in (3.20):

$$\rho_m c \sum_{n=1}^{20} \left[\int_{\Omega} \psi^M \psi^N dV \; \frac{\partial T^N}{\partial t} \right] = - \int_{\partial\Omega} \psi^M q_i n_i \; dS - \kappa \sum_{n=1}^{20} \left[\int_{\Omega} \frac{\partial \psi^M}{\partial x_i} \frac{\partial \psi^N}{\partial x_i} dV \; T^N \right] \tag{3.21}$$

The integrals of the shape function were evaluated in the preprocessing step, resulting in the following simplification:

$$\mathbf{M}_e \frac{\partial \mathbf{T}_e}{\partial t} = - \int_{\partial\Omega} \left(\psi^M \vec{q} \cdot \hat{n} \right) dS - \mathbf{K}_e \mathbf{T}_e, \tag{3.22}$$

where $\mathbf{M}_e$ and $\mathbf{K}_e$ are again the elemental mass and stiffness matrices, $\mathbf{T_e}$ is the vector of nodal temperatures, and $\int_{\partial\Omega} \left(\psi^M q_i n_i \right) dS$ is the boundary term. Assembly of the elemental equations yields the semi-discretized thermal-diffusion equation:

$$\mathbf{M} \frac{\partial \mathbf{T}}{\partial t} = \mathbf{Q}_{bound} - \mathbf{K}\mathbf{T}, \tag{3.23}$$

where $\mathbf{M}$ and $\mathbf{K}$ are again the global mass and stiffness matrices, $\mathbf{T}$ is the vector of nodal temperatures, and $\mathbf{Q}_{bound}$ is the boundary flux vector.

For the thermal-diffusion equations, the alpha-family of approximations is used [72]:

$$\mathbf{T}^{n+1} = \mathbf{T}^n + \Delta t \left(\frac{\partial \mathbf{T}}{\partial t}\right)^n + \alpha \Delta t \left(\left(\frac{\partial \mathbf{T}}{\partial t}\right)^{n+1} - \left(\frac{\partial \mathbf{T}}{\partial t}\right)^n\right) \tag{3.24}$$

with $\alpha = 0.5$ for Crank-Nicolson's scheme or $\alpha = 0.0$ for the Euler forward difference scheme.

Application of (3.24) to (3.23) yields:

$$\begin{aligned}(\mathbf{M} + \alpha \Delta t\ \mathbf{K})\, \mathbf{T}^{n+1} = (\mathbf{M} - \Delta t\,(1-\alpha)\,\mathbf{K})\, \mathbf{T}^n \\ + \Delta t \left(\alpha\ \mathbf{Q}_{bound}^{n+1} + (1-\alpha)\, \mathbf{Q}_{bound}^n\right)\end{aligned} \tag{3.25}$$

Equation (3.25) is solved for $\mathbf{T}^{n+1}$ through LU decomposition.

III.3. Coupling Algorithm

A three-way coupling algorithm was implemented to satisfy the boundary conditions in Section II.4. The simplest and most stable [73] means of coupling is to implement Dirichlet boundary conditions on the fluid, while imposing Neumann boundary conditions on the solid. Therefore, the traction and heat flux on the surface are calculated in the RANS solver and given to the structural elastic and thermal-diffusion solvers as boundary conditions. Likewise, the deformation and temperature are determined in the structural elastic and thermal-diffusion solvers and set as boundary conditions in the RANS solver. In the case of (2.13a), (2.14a), and (2.15), the coupling terms are built into the finite element models as $\mathbf{F}_{bound}$, $\mathbf{Q}_{bound}$, and $\mathbf{F}_{\Delta T}$, respectively. (2.13b) and (2.14b) are applied to the boundary conditions of the RANS solver.

For non-matching boundary meshes, the values of force, displacement, temperature, and heat flux at each of the nodes were interpolated to communicate between the RANS and FEM solvers. For the present study, the nodes were matched on the shared boundary, so no mapping of values was necessary. The nodal displacements, forces, heat flux, and temperature were directly communicated using a Message Passing Interface between the various solvers in the coupled algorithm.

III.3.1. Data Dependencies

The structure of the coupling algorithm is determined by the data dependencies between the solvers, which are shown in Figure III.2. In order to determine the flowfield at time t_{n+1}, the RANS solver requires the temperature and position of the boundary at time t_{n+1}, represented as $\mathbf{T}^{n+1}$ and $\mathbf{x}^{n+1}$, respectively.

Simultaneous dependencies were avoided between the thermal-diffusion solver and the RANS solver by implementing the Euler forward difference scheme so that only $\mathbf{q}^n$ was required from the RANS solver. While the Euler forward difference scheme places requirements on the time step to maintain stability, these requirements are typically less strict than those of the RANS solver, due to the long time scale of the thermal solution.

In order to integrate using the Newmark-beta scheme, the structural elasticity solver requires both the boundary forces and internal temperatures at time t_{n+1}, represented as $\mathbf{T}^{n+1}$ and $\mathbf{p}^{n+1}$, respectively. Note that the RANS solver and structural elasticity solver have simultaneous dependencies, which must be resolved through a method such as interfield iteration [57].

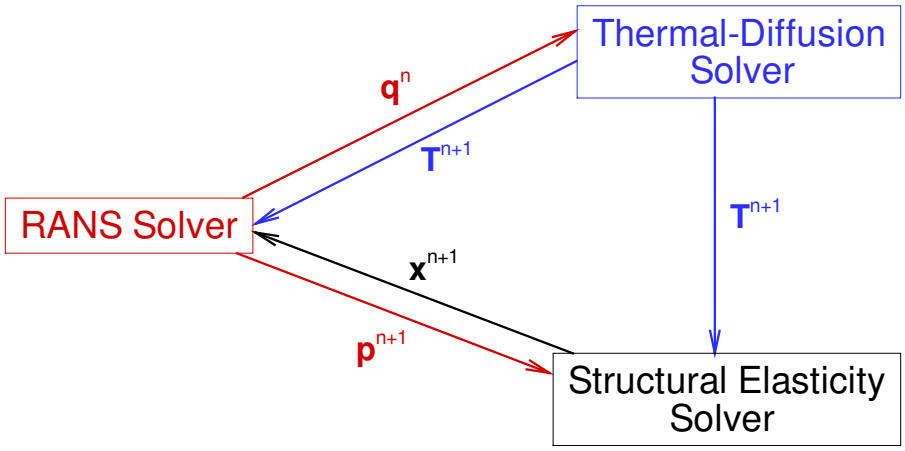


Figure III.2: Data dependencies in the aerothermoelastic coupled solver.

III.3.2. Aerothermal Coupling

Because the Euler forward difference scheme was used to calculate the evolution of the temperature in the thermal-diffusion solver, the heat flux was calculated in the aerodynamic solver at time t_n. This was accomplished differently for the RANS, piston theory, and Euler solvers. With the RANS model, the heat flux was calculated using Fourier's Law on the boundary. If the piston theory or Euler solvers were used, the heat flux into the coupled surface was either specified or calculated using a convective constant h, such that $q = h(T_{0,\infty} - T_w)$. q is the heat flux into the surface, $T_{0,\infty}$ is the adiabatic wall temperature, and T_w is the current wall temperature. The convective constant h was chosen depending on the requirements of the simulation. For the aerothermoelastic studies in Chapter VI, h represented the time required to heat the panel. Larger h reduced the computational cost of a simulation, while smaller h provided greater accuracy when determining threshold temperatures.

After communication of the heat flux, the temperature at time t_{n+1} was calcu-

lated, communicated to the RANS solver, and used as the boundary condition for the pseudo-time iteration for time t_{n+1}.

III.3.3. Aeroelastic Coupling

A modified interfield iteration method was applied to satisfy the simultaneous dependencies of the RANS solver and structural elasticity solver. For standard interfield iteration, the entire time step is repeated, with updated boundary conditions for each iteration, as seen in Figure III.3a. This requires that the lengthy pseudo-time convergence process for the RANS solver be repeated. The modified method only requires partial convergence through each iteration, and then updates the original guess for the RANS solver before the next iteration, as seen in Figure III.3b. This modification significantly reduces the required number of pseudo-time steps between iterations, therefore shortening the total computation time.

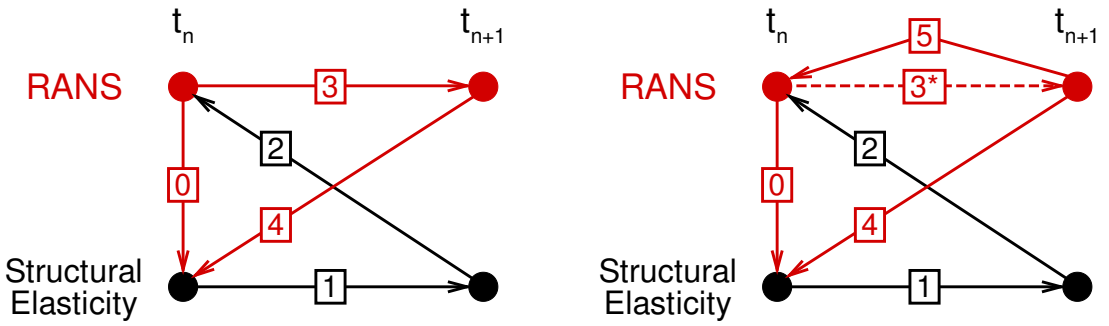
III.3.4. Aerothermoelastic Coupling

The coupling algorithm was composed of the following steps:

1. The heat flux into the coupled surface at time t_n was calculated by the RANS solver and was passed to the thermal-diffusion solver.

2. The thermal-diffusion solver calculated the temperature in the entire structural domain at time t_{n+1} using forward Euler integration. The temperatures of the full domain were passed to the structural elasticity solver. The temperatures at the shared boundary were passed to the RANS solver.

3. The forces on the coupled surface were calculated by the RANS solver at time



(a) Original interfield iteration procedure.

(b) Modified interfield iteration procedure.

Figure III.3: Comparison of iteration procedures. The steps are as follows: **Step 0** – Boundary load prediction using previous time step. **Step 1** – Time integration using Newmark-beta scheme. **Step 2** – Correction of boundary position. **Step 3** – Pseudo-time iteration to convergence. **Step 3*** – Pseudo-time iteration to partial convergence. **Step 4** – Correction of boundary load. **Step 5** – Update of initial guess.

t_n and were passed to the structural elasticity solver.

4. The structural elasticity solver calculated the deformations over the entire structural domain, including the effects of thermal expansion, passing the deformation of the shared surface to the RANS solver.

5. The RANS solver partially converged the solution through iteration in pseudo-time and updated the forces on the shared boundary, passing the corrected loads to the structural elasticity solver.

6. If the loads had been modified significantly from the previous iteration, steps 4 and 5 are repeated. Otherwise, computation proceeded to the next time step.

III.3.5. Aerodynamic Domain Grid Deformation

The structural elastic solver specifies the evolving deformation of the structural domain. As expressed in (2.13b), the deformation of the domains for the aerodynamic and structural solvers were required to match on the shared boundary. The deformation of the structural domain is specified by (2.9). The deformation of the mesh within the aerodynamic domain is only constrained by (2.13b). To maintain a high quality aerodynamic mesh, radial basis function interpolation [74, 75] was used to interpolate the deformation on the boundary to the interior of the domain.

The radial basis function interpolation of a scalar function f is defined as [76]:

$$f(\vec{x}) = \sum_{i=1}^{N_b} a_i \phi \left(\| \vec{x} - \vec{x}_{b_i} \| \right) + \psi(\vec{x}), \tag{3.26}$$

where N_b is the number of boundary nodes, ϕ is the radial basis function, $\vec{x}$ is the location within the aerodynamic domain, $\vec{x}_{b_i}$ is the location of boundary node i, $\psi(\vec{x}) = b_0 + b_1 x + b_2 y + b_3 z$ is a linear polynomial, and the coefficients a_i and b_0, b_1, b_2, and b_3 are the solution of the radial basis function interpolation problem. Because this is a scalar formulation, separate interpolation functions are required to find the deformation in x, y, and z.

As an interpolation, the evaluation of f on the boundary nodes must be exact:

$$f(\vec{x}_{b_i}) = d_{b_i}, \tag{3.27}$$

where d_{b_i} is the displacement of boundary node i in x, y, or z. This results in N_b equations:

$$f(\vec{x}_{b_i}) = \sum_{j=1}^{N_b} a_j \phi_{ij} + \psi(\vec{x}_{b_i}) = d_{b_i}, \qquad i = 1, ..., N_b, \tag{3.28}$$

where $\phi_{ij} = \phi\left(\|\vec{x}_{b_j} - \vec{x}_{b_i}\|\right)$.

Four additional constraints are necessary because of the four parameters b_0, b_1, b_2, and b_3. These constraints are:

$$\sum_{i=1}^{N_b} a_i = 0 \qquad\qquad \sum_{i=1}^{N_b} a_i \vec{x}_{b_i} = \vec{0} \tag{3.29}$$

Together, these $N_b + 4$ constraints form a system of equations [76]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P^T} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \begin{Bmatrix} \mathbf{d}_b \\ \mathbf{0} \end{Bmatrix}, \tag{3.30}$$

where $\mathbf{a}= \{a_1, a_2, \ldots, a_n\}^{\mathbf{T}}$, $\mathbf{b}= \{b_0, b_1, b_2, b_3\}^{\mathbf{T}}$, $\mathbf{d}_b = \{d_{b_1}, d_{b_2}, \ldots, d_{b_n}\}^{\mathbf{T}}$.

$\mathbf{M}$ and $\mathbf{P}$ are defined as:

$$\mathbf{M} = \begin{bmatrix} \phi_{11} & \phi_{12} & \ldots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \ldots & \phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \ldots & \phi_{nn} \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} 1 & x_{b_1} & y_{b_1} & z_{b_1} \\ 1 & x_{b_2} & y_{b_2} & z_{b_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{b_n} & y_{b_n} & z_{b_n} \end{bmatrix}$$

Once (3.30) is solved for $\mathbf{a}$ and $\mathbf{b}$, (3.26) can be evaluated for each node in the aerodynamic mesh, resulting in a smooth, high-quality mesh which conforms to the required deformation on the shared boundary of the structural and aerodynamic solver.

CHAPTER IV

HYPERSONIC AERODYNAMICS*

In this chapter, the stand-alone RANS solver is validated for hypersonic heat transfer. Once validated, numerical simulations are performed on a hypersonic double-wedge airfoil, calculating heat flux variations under a variety of flow conditions.

IV.1. Hypersonic Flow Solver Validation

The hypersonic flow solver is the key component of the aerothermal, aeroelastic, and aerothermoelastic solvers. Consequently, it is critical that it correctly predict the loads and heat transfer rates. In the original code, the carbuncle phenomenon caused a great deal of inaccuracy in the heat transfer rate prediction. For this reason, modifications were made to increase the accuracy of the hypersonic flow solver. Because very little data exists to validate the coupled solvers, validation is presented for the stand-alone hypersonic solver. This validation focuses on heat transfer predictions for known problems.

Two validation cases were performed for the standalone hypersonic flow solver. The first of these tested the effectiveness of the new flux function, RoeM, which was implemented to address the carbuncle phenomenon. The validation case, therefore,

*Reprinted with permission from "Numerical Investigation of Actively Cooled Structures in Hypersonic Flow" by Robert L. Brown, Kaushik Das, Paul G. A. Cizmas, and John D. Whitcomb, 2014. *Journal of Aircraft*, Vol. 51, No. 5, 1522-1531, Copyright 2013 by Robert L. Brown, Kaushik Das, Paul G. A. Cizmas, and John D. Whitcomb.

is hypersonic flow on a blunt body. The peak heat flux calculated by the flow solver is compared to that predicted by the Fay-Riddell equation [8].

The second validation case for the standalone hypersonic flow solver tested the heat flux along a flat plate. It is expected that the heat flux for a flat plate can be reasonably modeled using compressible boundary layer flow [77]. The heat fluxes calculated both using the hypersonic solver and boundary layer flow are compared for a flat plate with no angle of attack.

IV.1.1. Blunt Body Heat Flux in Hypersonic Flow

The heat flux generated by hypersonic flow around a cylinder was predicted using the RANS flow solver and a semi-empirical equation proposed by Fay and Riddell [8]. The Fay–Riddell equation predicts the peak heat flux, q_{FR}, at the leading edge of the cylinder:

$$q_{FR} = 0.76 Pr^{-0.6} (\rho_e \mu_e)^{0.4} (\rho_w \mu_w)^{0.1} \sqrt{K} \left(h_{0_\infty} - h_w \right), \tag{4.1}$$

where Pr is the Prandtl number, ρ_e and μ_e are the density and viscosity behind the shock, ρ_w and μ_w are the density and viscosity at the wall, K is the velocity gradient at the edge of the boundary layer, h_{0_∞} is the far-field stagnation enthalpy, and h_w is the enthalpy at the wall.

The velocity gradient K at high Mach numbers can be approximated using Newtonian impact theory and Bernoulli's equation, which results in [6]:

$$K = \left. \frac{du}{dx} \right|_e \approx \frac{U_\infty}{D} \sqrt{\frac{8\rho_\infty}{\rho_e} \left(1 - \frac{p_\infty}{p_e} \right)} \tag{4.2}$$

The edge velocity gradient K predicted using (4.2) is uniformly too large compared to the experimental data of Korobkin [7], as shown in Figure IV.1. To correct this

discrepancy, herein a least-squares fit of the experimental data was used to scale the edge velocity gradient. The scaling factor was found to be 0.958, so that the expression of the edge velocity gradient becomes

$$K = \left.\frac{du}{dx}\right|_e \approx 0.958 \frac{U_\infty}{D} \sqrt{\frac{8\rho_\infty}{\rho_e}\left(1 - \frac{p_\infty}{p_e}\right)}. \tag{4.3}$$

Figure IV.1 shows good agreement of the edge velocity gradient obtained from the experimental data and the scaled correction to Newtonian theory proposed herein.

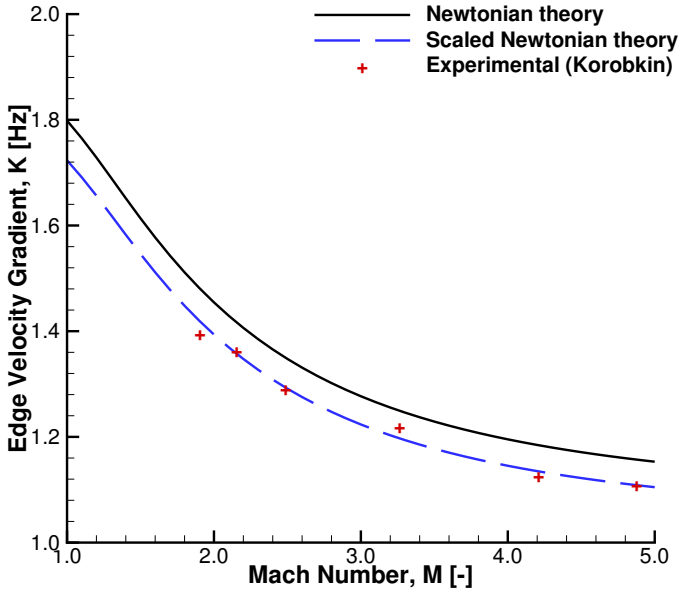


Figure IV.1: Circular cylinder leading edge velocity gradient versus Mach number. Experimental data [7] is compared to Newtonian impact theory and scaled Newtonian theory.

The flow over a cylinder with radius 0.5 mm was simulated assuming a wall temperature, T_w, of 800 K and a Mach number, M_∞, of 8, at an altitude of 25 km. For these conditions, the Fay–Riddell equation predicts a peak heat flux of 11.448 MW/m^2

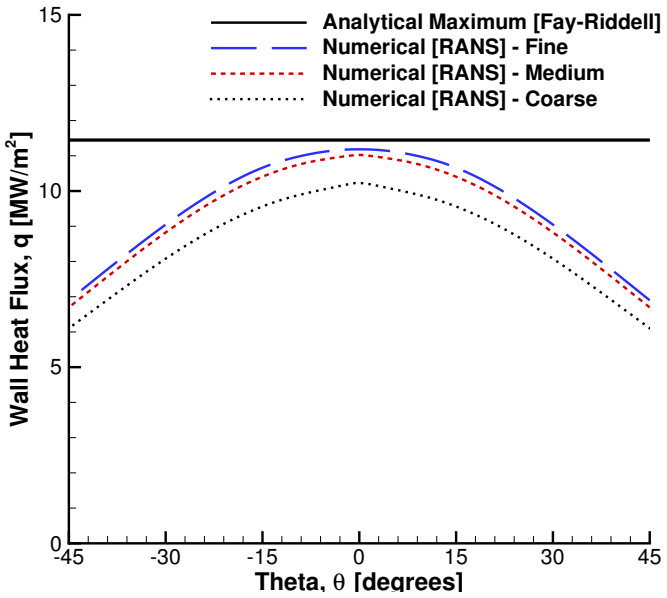


Figure IV.2: Heat flux variation on a hypersonic cylinder. The cylinder has a wall temperature of 800 K and is subjected to flow at Mach 8 and 25 km altitude. Numerical [RANS] and analytical [Fay–Riddell] predictions are shown.

when using the scaled edge velocity gradient (4.3).

The RANS flow solver described in Section III was used to predict the peak heat flux on three meshes of increasing refinement. The coarsest mesh had 20 points evenly spaced over the surface, 1.4 radians in each direction from the stagnation point. 40 points were distributed normal to the surface to a distance 0.5 mm from the surface. The height of the cells adjacent to the wall was 1.6 μm for the coarsest mesh, which was chosen so that the maximum y^+ number was approximately 2. With each successive refinement, the number of points in each direction was doubled, and the height of the cells adjacent to the wall was reduced by half. RANS simulation using the finest grid resulted in a peak heat flux prediction of 11.248 MW/m^2, which

is approximately 1.75% less than that predicted by (4.1). The distribution of heat flux over the front of the cylinder is shown in Figure IV.2. The solutions for all three meshes are free from the distortions normally associated with the carbuncle effect.

IV.1.2. Heat Flux on a Flat Plate in Hypersonic Flow

The regions away from the leading edge and the inflection points at mid-chord on a double-wedge airfoil can be reasonably modeled with flat plate boundary layer governing equations. Since these regions make up most of the surface of the airfoil, the validation of heat flux rates over a flat plate was performed. Heat flux rates were calculated for a flat plate parallel to the flow using both the solution to the boundary layer equations [77] and RANS simulations. This plate was simulated at a speed of Mach 5 through air at sea level. The temperature of the plate was set to 400 K.

Three meshes of increasing refinement were used to assess solution grid independence for RANS simulations. Each mesh covered 50 mm in the x direction, which was the direction oriented with the flow, and 20 mm in the y direction, which was normal to the plate. The coarsest mesh had 50 and 100 nodes in the x and y directions, respectively. The cell adjacent to the wall at the leading edge had a height of 1 μm, which was chosen to yield a y^+ number of less than 1 past 2 mm from the leading edge. The length of the same cell was 100 μm. A constant growth ratio of 1.076 was used in each direction. For each successive refinement, the number of nodes in each direction was doubled, and the initial spacing was cut in half.

Figure IV.3 shows the heat flux rates calculated by the RANS solver, compared to those predicted using the solution to the boundary layer equations. The heat

Table IV.1: Grid convergence study for a flat plate.

Model		Grid Data [μm]		Output Coef.[b]	
		Δx_1[a]	Δy_1[a]	A [kW/m^2]	k [–]
RANS	Coarse	100.0	1.00	487.5	-0.4639
	Medium	50.0	0.50	493.6	-0.4806
	Fine	25.0	0.25	497.6	-0.4911
Boundary Layer		–	–	507.0	-0.5000

[a] x and y grid spacing at the leading edge.
[b] Coefficients in the equation: $q_w(x) = A\,(x/(25\text{mm}))^k$.

flux values are nearly identical between the three cases, especially far away from the leading edge. Table IV.1 shows a summary of results from the three RANS simulations and the solution to the boundary layer equations. The solution of the boundary layer equations predicted a curve of the form $q_w = Ax^k$. Fits to this curve were calculated for results from each of the RANS simulations. As the mesh was refined, both coefficients A and k approached the values calculated with boundary layer analysis, indicating that the RANS solver can accurately capture the heat flux profile over the flat regions of the airfoil.

IV.2. Heat Flux on a Double-Wedge Airfoil with Isothermal Walls

The flow over the double-wedge airfoil shown in Figure IV.4 was simulated with the following conditions, chosen to include a representative envelope for a vehicle in hypersonic cruise: Mach number ranging from 3 to 8, altitude ranging from sea level to 45 km, and wall temperature ranging from 400 K to 800 K. The geometry of the

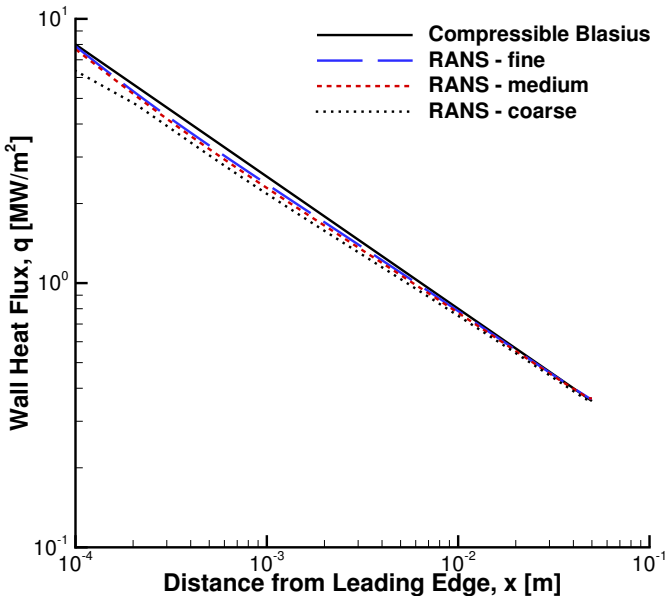


Figure IV.3: Heat flux on a hypersonic flat plate. The heat flux is calculated using compressible Blasius flow and the RANS solver on a medium and coarse grid.

airfoil was chosen to be the root airfoil of the fin defined by [78, 79], except that the leading edge was rounded with a radius of curvature of 5 mm. This value was chosen to avoid the heat flux singularity that occurs as the leading edge diameter, D, approaches zero in (4.3).

Unless otherwise indicated, the simulations were run with an angle of attack of 0.6 degrees. A grid convergence study was completed first to determine the mesh that provides the best compromise between accuracy and computational cost. Additionally, studies were performed to determine the effect of angle of attack, altitude, speed, and wall temperature on the heat flux into the airfoil.

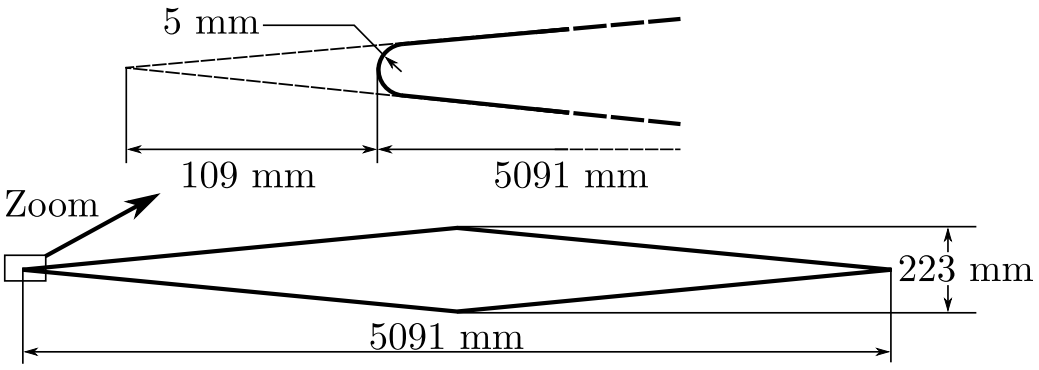


Figure IV.4: Double-wedge airfoil geometry. The leading edge is rounded with a radius of curvature of 5 mm. Without the rounded leading edge, the airfoil would be 5.2 m long.

IV.2.1. Grid Convergence Study

A grid convergence study was performed to ensure that the solution was grid independent. High refinement was necessary in regions where the flow direction changed. For this reason, the grids tested had more nodes near the midpoint and leading and trailing edges than over the rest of the airfoil. Points were also clustered near the surface of the airfoil to capture the boundary layer. Figure IV.5 shows the region of the coarse mesh immediately surrounding the wedge airfoil. For each successive refinement, the initial spacing along the chord at the leading edge, trailing edge, and midpoint were halved, and the number of points on the airfoil surface was doubled. Additionally, the initial spacing normal to the surface of the airfoil was halved with each refinement.

A summary of the results is given in Table IV.2, and the chordwise variation of the heat flux for each of the grids is shown in Figure IV.6. The results showed

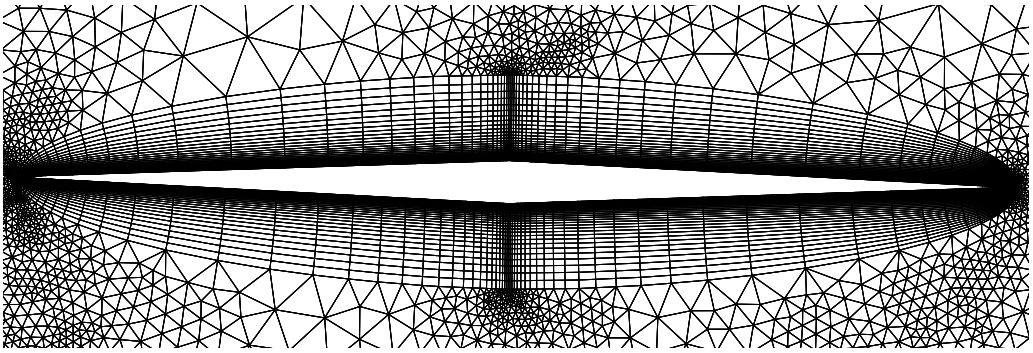


Figure IV.5: Zoomed view of coarse mesh near the double-wedge airfoil.

Table IV.2: Grid convergence study for a wedge airfoil.

		Grid Spacings [μm]				Heat Flux [kW/m^2]		
	N[a]	Δs_{LE}[b]	Δs_{mid}[b]	Δs_{TE}[b]	Δy_1[c]	q_{LE}[d]	q_{avg}[d]	CPU Time [hr]
Coarse	150	1200	8000	1600	200	139.5	1.110	1.06
Medium	300	600	4000	800	100	186.6	1.271	4.68
Fine	600	300	2000	400	50	188.6	1.360	13.16

[a] Number of points on airfoil.
[b] Chordwise grid spacing (LE – leading edge, mid – midpoint, TE – trailing edge).
[c] Grid spacing normal to airfoil surface [d] Heat flux (LE – leading edge, avg – averaged over airfoil).

no appreciable change in the heat flux as the mesh was refined. The distribution of heat flux over the airfoil did not significantly change between the medium and fine grids, indicating that the numerical solution is grid independent.

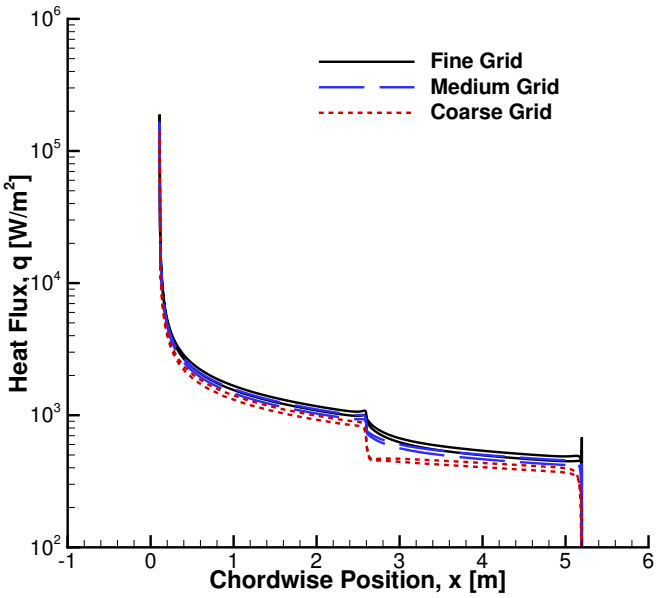


Figure IV.6: Heat flux at the airfoil surface calculated by the RANS solver. The flow was simulated at Mach 5, 45 km altitude, with a wall temperature of 800 K.

IV.2.2. Angle of Attack Variation

This section investigates the dependence of the heat flux on the angle of attack. The angle of attack of an airfoil is dictated by the flight conditions of the vehicle. The analysis is motivated by the possibility that a significant change in heat flux due to angle of attack variation could limit the flight envelope.

In this analysis, the altitude and speed were set to 45 km and Mach 5, respectively. The heat flux profile was calculated over the airfoil as the angle of attack was varied between -10 and 10 degrees.

Figure IV.7 shows the increase in heat flux due to angle of attack on the bottom surface of the airfoil for angles of attack from 1 degree to 5 degrees. The leading edge shows the smallest percentage change due to angle of attack. For all angles of attack,

local maxima occur near 21% chord ($x = 1.1$ m), 82% chord ($x = 4.3$ m), and 100% chord ($x = 5.2$ m). The percentage change remains relatively constant over the flat regions of the airfoil, with disturbances near the inflection point at midchord and at the trailing edge. At 5 degrees angle of attack, the heat flux increases at most 68% due to the increased angle of attack, and does not exceed 60% over the first half of the airfoil, where absolute heat flux values are higher. This is a significant increase that must be taken into account during the design of cooling systems.

Figure IV.8 shows the heat flux variation with angle of attack for three points on the bottom surface of the airfoil: 4% chord ($x = 0.2$ m), 20% chord ($x = 1.0$ m), and 82% chord ($x = 4.2$ m). The average heat fluxes over the entire airfoil and on the bottom surface are also shown. Although the percentage increase was higher further from the leading edge, the largest change occurred immediately past the leading edge of the airfoil.

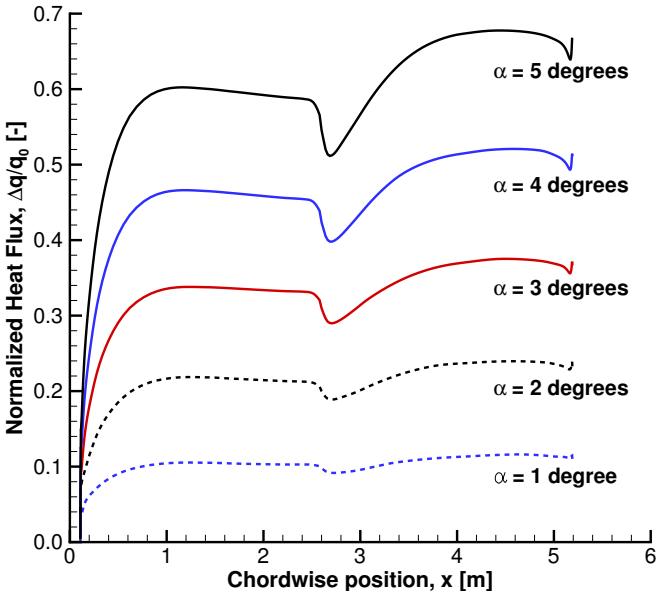


Figure IV.7: Normalized change in heat flux on the bottom surface of the airfoil due to angle of attack. Flow conditions are Mach 5, 45 km altitude, with a wall temperature of 800 K.

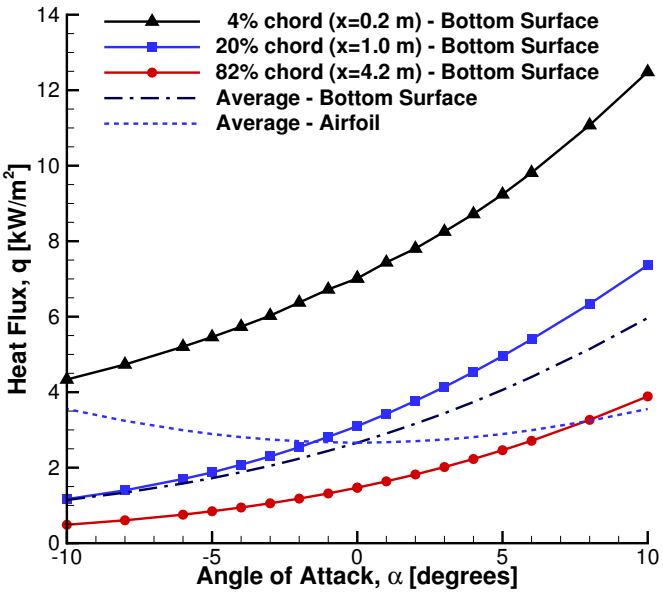


Figure IV.8: Heat flux variation with angle of attack. Simulated at Mach 5, 45 km altitude, with a wall temperature of 800 K.

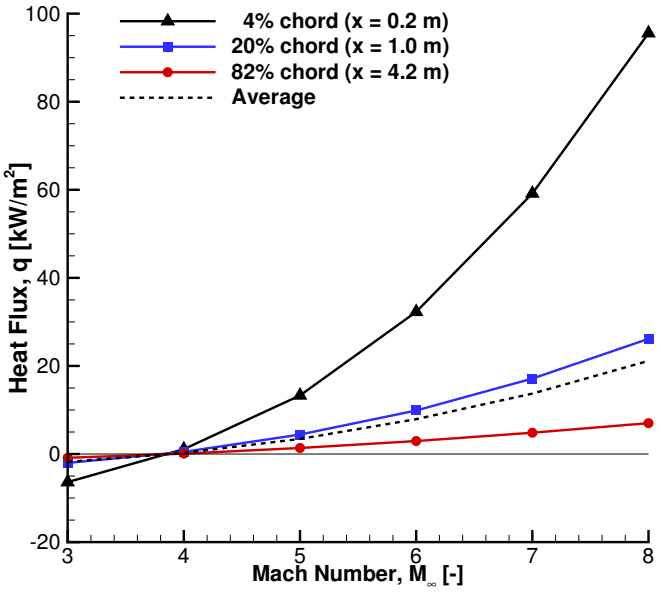


Figure IV.9: Heat flux variation with Mach number. The heat flux was measured at three locations along a double-wedge airfoil for an altitude of 25 km and a wall temperature of 800 K.

IV.2.3. Flight Environment Variation

This section presents the effect of flight conditions on the heat flux into the airfoil. Simulations were performed for Mach numbers ranging from 3 to 8, for wall temperatures ranging from 400 K to 800 K, and for altitudes ranging from sea level to 45 km. These results were then used to develop flight envelopes for different levels of acceptable heat flux.

Figure IV.9 shows the heat flux variation with Mach number at several chordwise positions for the subset of the cases run at an altitude of 25 km and a wall temperature of 800 K. Cooling is required to maintain a wall temperature of 800 K at speeds exceeding Mach 4. At speeds slower than Mach 4, the adiabatic wall temperature is cooler than 800 K, and the flow cools the structure instead of heating it.

The cooling requirements at 4% chord ($x = 0.2$ m) are 3 to 4 times larger than the heat flux at 20% chord ($x = 1.0$ m). This difference is significant enough to suggest that the cooling at the leading edge should be handled separately from cooling over the rest of the airfoil.

Figure IV.10 shows the limitations to the flight envelope of the double-wedge airfoil due to the available average cooling and the allowable wall temperature. The maximum speed increases significantly as altitude is increased for a given heat flux and wall temperature. To fly at lower altitudes and/or higher speeds, the average heat flux and/or the allowable wall temperature must be increased. This indicates a trade-off between the cooling system, the materials and structure, and the flight envelope of the airfoil.

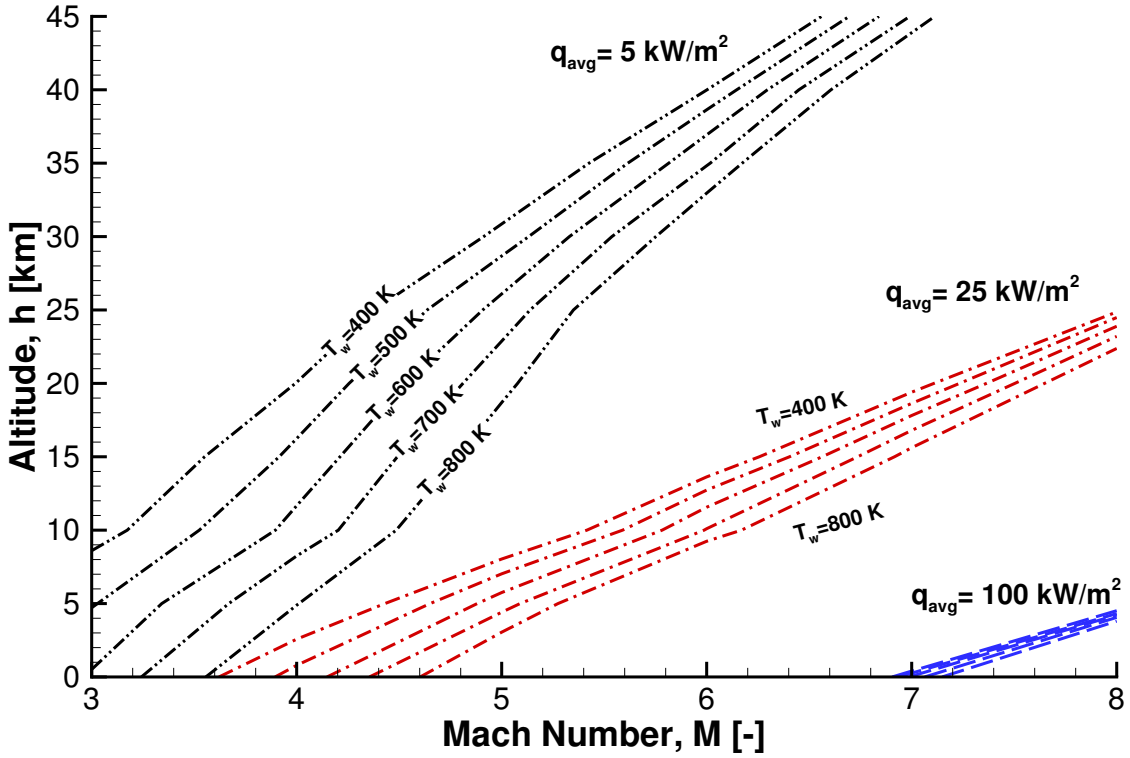


Figure IV.10: Lines of constant average heat flux and wall temperature calculated using the RANS solver. For each set of lines representing a single average heat flux, T_w increases with increasing Mach number and decreasing altitude. The maximum calculated value of q_{avg} is 145.6 kW/m^2, which occurs at sea level, flying at Mach 8 with T_w=400 K.

CHAPTER V

HYPERSONIC AERODYNAMICS WITH THERMAL COUPLING*

This chapter presents the validation of the coupled aerothermal solver, followed by numerical results for a hypersonic double-wedge airfoil with several internal active cooling profiles.

V.1. Aerothermal Solver Validation – Spherical Protuberance

The validation of the aerothermal solver was designed to ensure the accurate handling of the communicated heat flux and temperature between the RANS solver and the thermal-diffusion solver. Since the individual solvers have already been validated, this was solely a validation of the communication technique. The chosen validation case was a spherical protuberance of a flat plate in hypersonic flow.

Hypersonic flow over a spherical protuberance has previously been studied experimentally by [80] and later computationally by [81], which form the basis for the present simulation. The spherical protuberance used here matched the computational study in [81] and run 14 in [80]. The protuberance was 28 inches in diameter with a height of 0.76 inches. It was attached to a flat plate subject to a Mach 6.59 flow at 5 degrees angle of attack, with a boundary layer developed over 57.8 inches, resulting in a boundary layer height of 12.7 mm.

---

*Section V.2 reprinted with permission from "Numerical Investigation of Actively Cooled Structures in Hypersonic Flow" by Robert L. Brown, Kaushik Das, Paul G. A. Cizmas, and John D. Whitcomb, 2014. *Journal of Aircraft*, Vol. 51, No. 5, 1522-1531, Copyright 2013 by Robert L. Brown, Kaushik Das, Paul G. A. Cizmas, and John D. Whitcomb.

Table V.1: Spherical protuberance boundary layer grid summary.

				s_i[a] ($s = 0.82$m)			s_c[b] ($s = 1.48$m)		
	N_{LE}	N_s	N_y	y_1 (μm)	y^+	N_{BL}	y_1 (μm)	y^+	N_{BL}
Coarse	30	80	120	181.7	3.84	38	244.5	4.46	36
Medium	40	130	160	90.9	1.98	65	122.3	2.28	60
Fine	60	210	200	45.4	1.00	91	61.1	1.15	86

[a] Spherical protuberance inlet location [b] Location of comparison to experiment

V.1.1. Calculation of Boundary Layer

To determine the boundary layer profile at the inlet of the spherical protuberance region, a two-dimensional representation of the experimental setup of Glass [80] was simulated. The simulation was performed on coarse, medium and fine grids, which are described in Table V.1. Figure V.1 shows the relationship between the 2D flat plate domain and the 3D spherical protuberance domain. Figure V.2 shows the coarse mesh for the boundary layer domain.

The results of the grid convergence study can be seen in Figure V.3a, indicating that the boundary layer given by the medium and fine grids are nearly identical. The boundary layer on the medium grid was compared to both experimental [80] and computational [81] data 1.48 m (58.1 inches) downstream of the leading edge. This comparison is shown in Figure V.3b, and shows that we match both studies in the region of flow close to the wall. The error increases further from the wall. At y =50 mm, the error compared to previous computational data is 3.62%, and the error compared to experimental data is 12.82%. This gives us good confidence in the boundary conditions of our protuberance domain.

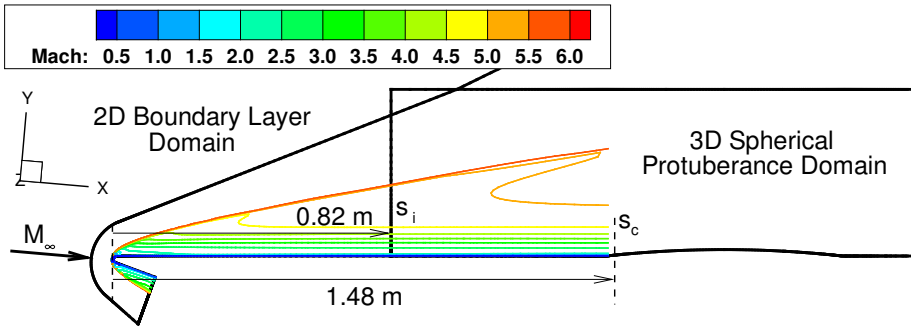


Figure V.1: 2D and 3D domains for spherical protuberance study. The flat plate has a rounded leading edge with a radius of curvature 9.65 mm, subject to a Mach 6.59 flow at 5 degrees angle of attack.

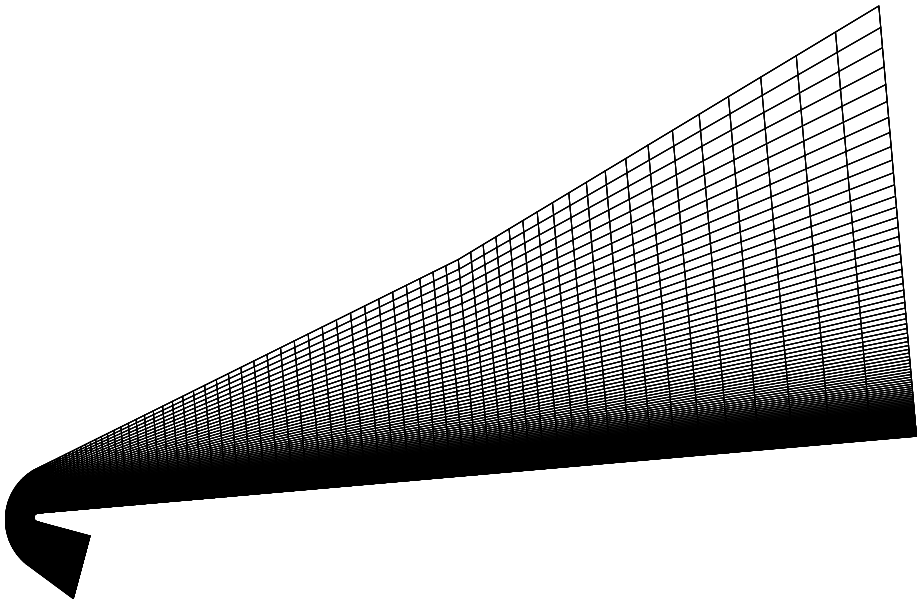
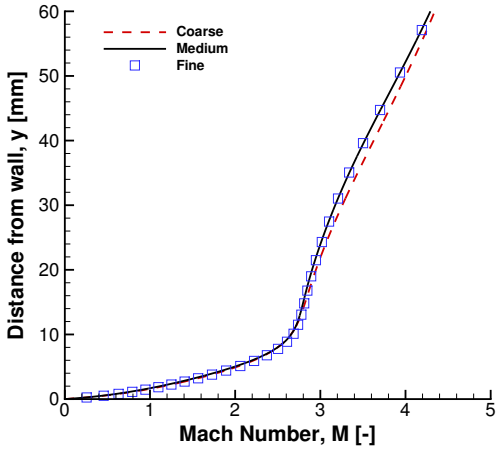
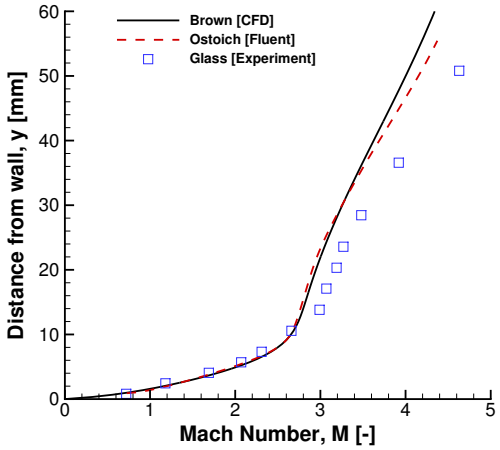


Figure V.2: Coarse mesh for 2D boundary layer domain.



(a) RANS solver grid convergence results.



(b) Comparison of RANS solver results to both Ostoich [81] and Glass [80].

Figure V.3: Boundary layer profile comparison 1.48 m from the leading edge of the flat plate.

V.1.2. Validation Setup

The domain extended from 1 m (1.41 protuberance lengths) in front of the center of the dome to 2 m (2.81 protuberance lengths) behind it, with 1 m simulated on either side. The top surface of the domain was 0.5 m (0.70 protuberance lengths) from the wall.

The boundary conditions for the spherical protuberance are shown in Figure V.4. The inlet and top surfaces of the domain were prescribed based on the boundary layer calculated in Section V.1.1. The bottom surface was a viscous wall, set to a temperature of 300 K for the grid convergence study, and coupled to the thermal-diffusion solver for the aerothermal study. The outlet surface was defined by the static pressure if the flow was supersonic, and defined by the Riemann invariant if

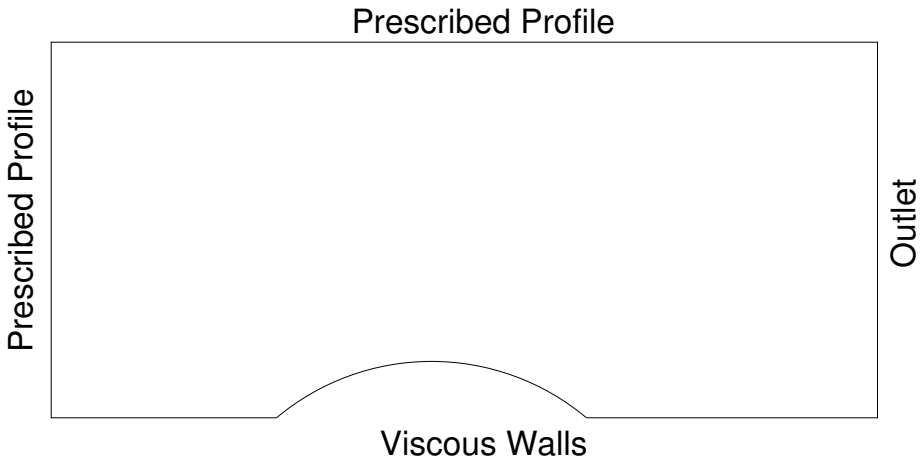


Figure V.4: Boundary conditions for the spherical protuberance case.

the flow was subsonic. The sides, which are not pictured, were free-stream conditions at Mach 6.59.

The spherical protuberance was meshed using Pointwise meshing software. The surface of the dome was divided into five blocks: a square in the center, with four structured domains surrounding it. These blocks were then smoothed using the Laplace smoother in Pointwise. An O-grid was created around the dome, and a structured grid was used in the wake area. The remaining area was filled with an unstructured domain. The surface mesh can be seen in Figure V.5.

The mesh for the top surface was formed by projection of the surface mesh. The nodes in the 3D volumes were specified algebraically with a hyperbolic tangent profile. The spacing on the bottom surface was specified by the desired refinement and zero curvature was specified on the top surface [82]. The completed three dimensional mesh is shown in Figure V.6.

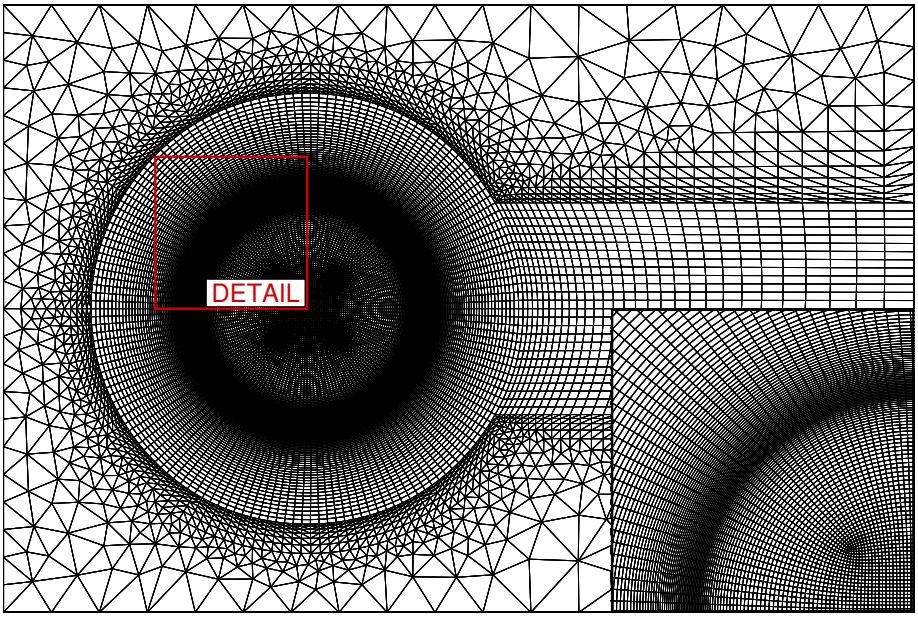


Figure V.5: Surface mesh for the spherical protuberance case with medium in-plane refinement.

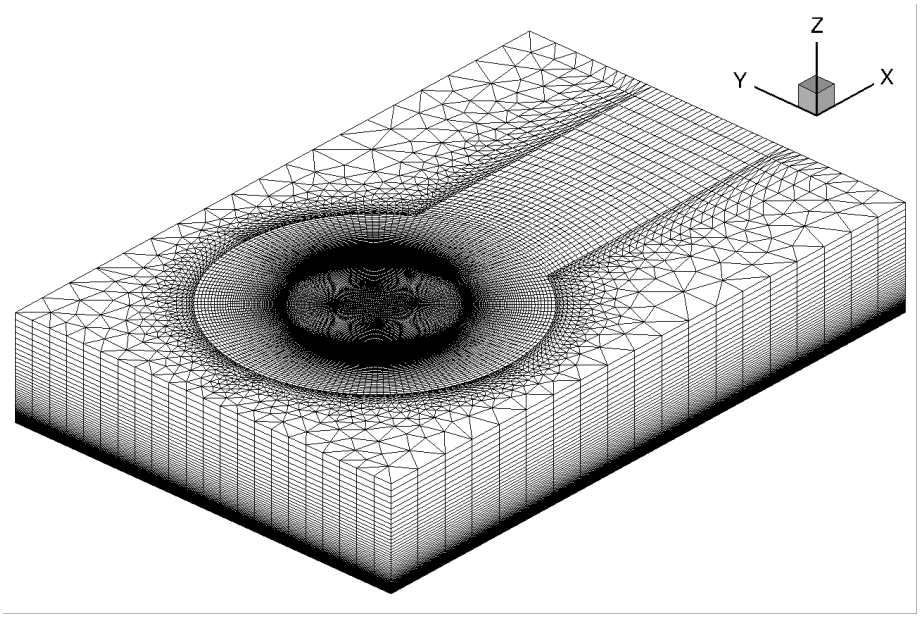


Figure V.6: Spherical protuberance mesh with medium in-plane refinement and coarse out-of-plane refinement.

Table V.2: Spherical protuberance grid summary.

Out-of-plane Distribution			**Surface Mesh**					
	N_y	y_1		N_{circle} [a]	N_{in} [b]	N_{out} [c]	Δs_1 [d]	N_{tot} [e]
1	100	100 μm	α	160	40	60	2.5 mm	19892
2	160	50 μm	β	240	60	70	2.0 mm	39122
3	240	25 μm	γ	320	80	80	1.5 mm	63417

[a] Number of points around circle [b] Number of points interior to circle
[c] Number of points exterior to circle [d] First cell spacing on circle boundary
[e] Number of total points on surface

V.1.3. Grid Convergence Study

In order to determine an appropriate mesh for the aerothermal validation, the grid refinement was analyzed separately in the surface plane and perpendicular to the surface. The heat flux variation due to the out-of plane refinement was analyzed for three point distributions (labeled 1, 2, and 3). The heat flux variation due to the refinement of the surface plane was analyzed for three surface meshes (labeled α, β, and γ). In total, nine meshes were studied, which are indicated through combinations of the out-of-plane point distributions and surface meshes. For instance, the mesh generated from the coarsest out-of-plane point distribution and the medium surface mesh is indicated as Mesh 1β. Details of the out-of-plane distributions and surface meshes are given in Table V.2.

Figure V.7 shows the comparison between Mesh 2α and Mesh 2γ, which are the coarsest and most refined surface meshes using the medium out-of-plane point distribution. The primary difference between the two is in the region of increased

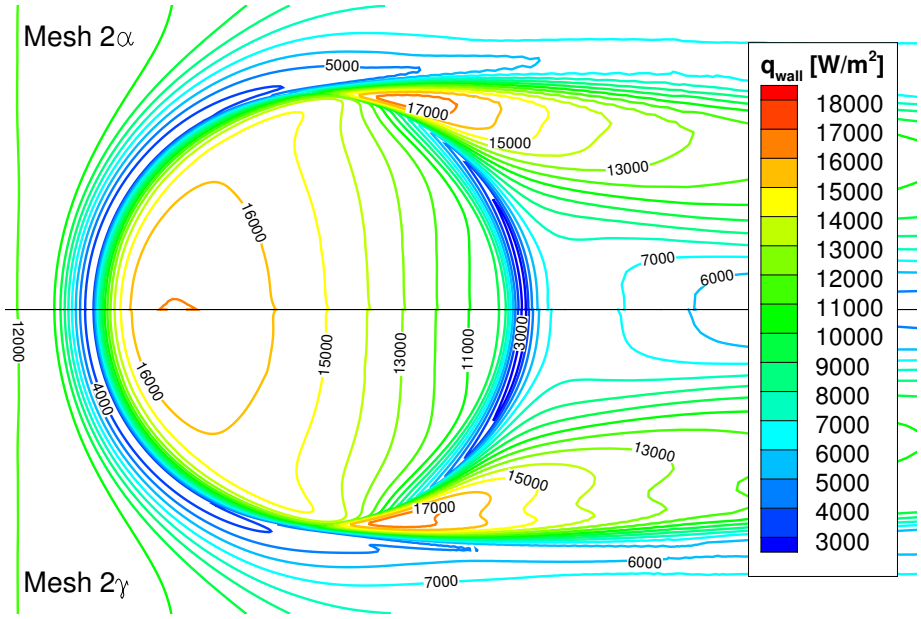


Figure V.7: Comparison of heat flux between coarse and fine surface meshes with medium out-of-plane point distribution.

heat flux due to the vortex formed off the sides of the spherical protuberance. For the case with fine in-plane refinement, there is a clear double-lobed shape of the region of increased heat flux. For the case with medium refinement, however, the vortex region is single-lobed.

Figure V.8 shows the comparison between Mesh 1γ and Mesh 3γ, which use the coarsest and most refined fine out-of-plane point distributions with the finest surface mesh. There is very little qualitative difference between the two sets of results. However, the fine case has a uniform increase in heat flux of about 1kW/m^2 over the entire domain. This suggests that the coarse case does not sufficiently resolve the boundary layer to capture the heat flux.

Figure V.9 shows the comparison between Mesh 3α and Mesh 3γ, which are

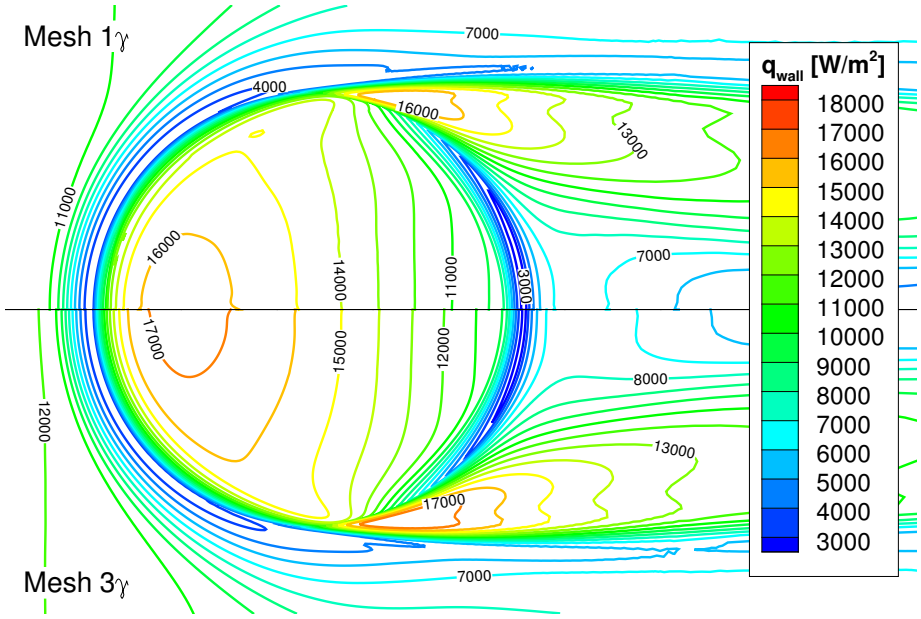


Figure V.8: Comparison of heat flux between coarse and fine out-of-plane point distributions with fine surface mesh.

formed from the coarse and fine surface meshes using the finest out-of-plane point distribution. Unfortunately, there is a clear anomaly in the heat flux of Mesh 3α. Though the precise cause of the anomaly is not clear, the likely cause is the extreme aspect ratio of the cells in the boundary layer. The aspect ratio has a maximum of 556.9 on the surface of the dome, calculated as the ratio of the maximum edge length and the minimum edge length for a given cell.

Finally, the average heat flux measurements between the studied grids are compared. Since we are primarily interested in the temperature evolution of the aluminum dome, only the dome is taken into account for these comparisons. As can be seen in Figure V.10, there is very little difference between the two in-plane refinements (if the clearly anomalous calculations are omitted), and we approach a

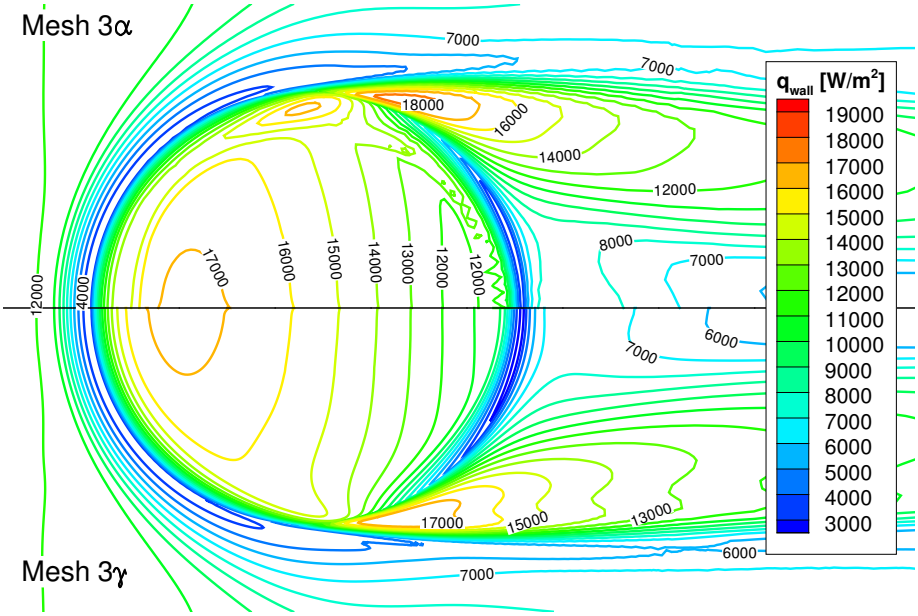


Figure V.9: Comparison of heat flux between coarse and fine surface meshes with fine in-plane point distribution. The mesh generated using the coarse surface mesh has an anomalous hot region, possibly due to excessive grid aspect ratio.

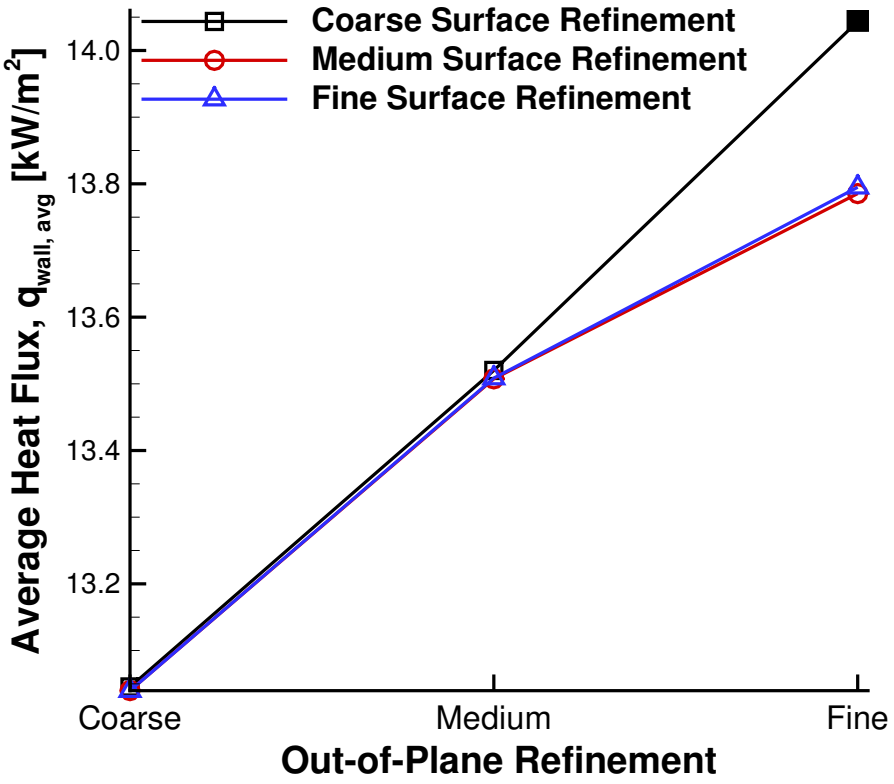


Figure V.10: Comparison of heat flux averaged over the spherical protuberance for all studied refinements. Filled symbol indicates anomalous result.

constant value as we refine in the out-of plane direction.

Based on the results from the grid convergence study, Mesh 2α was chosen for the aerothermal investigation. The coarse in-plane refinement was chosen as there was not enough difference between the solutions on the coarse and medium grids to justify increasing the number of nodes by almost a factor of two, unless the finest out-of-plane point distribution is used. This also motivated the choice of the medium out-of-plane point distribution, which avoids the anomaly shown in Figure V.9.

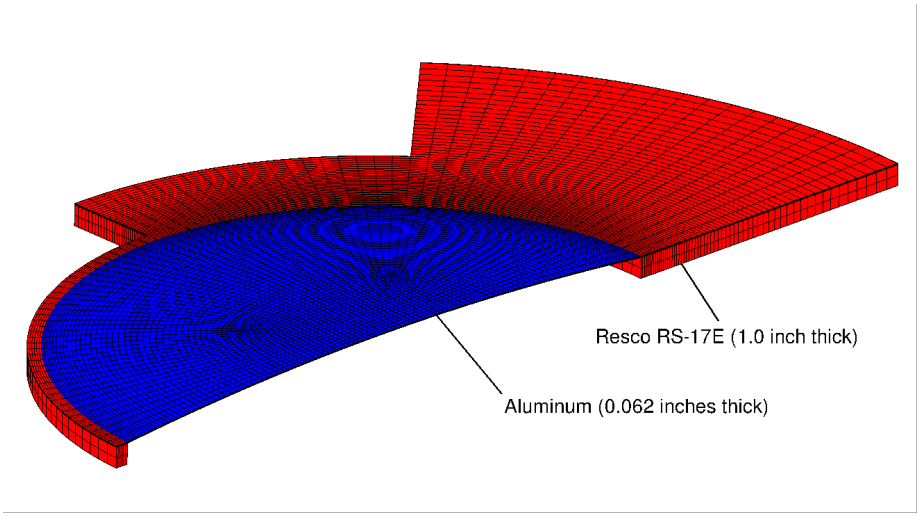


Figure V.11: Mesh used for thermal-diffusion solver.

V.1.4. Aerothermal Results

The aerothermal results require coupling the aerodynamic mesh to a thermal-diffusion solver mesh. The thermal-diffusion mesh matches the surface mesh of the aerodynamic solver exactly, and was extruded downward to generate the material regions. This mesh is shown in Figure V.11. Only the dome itself, the region immediately upstream and part of the downstream region was coupled to the thermal-diffusion solver, while the rest was held at a constant temperature of 300 K. The purpose of this was to reduce the size of the thermal-diffusion domain, as the solver was not as extensively parallelized as the flow solver.

The variation of heat flux over the centerline of the dome is shown in Figure V.12 and is compared to the simulation of Ostoich [81] for thermally perfect gas. Unfortunately, there is a significant discrepancy between the two simulations. However, this is most likely due to Ostoich's simulation using the combustion products of methane

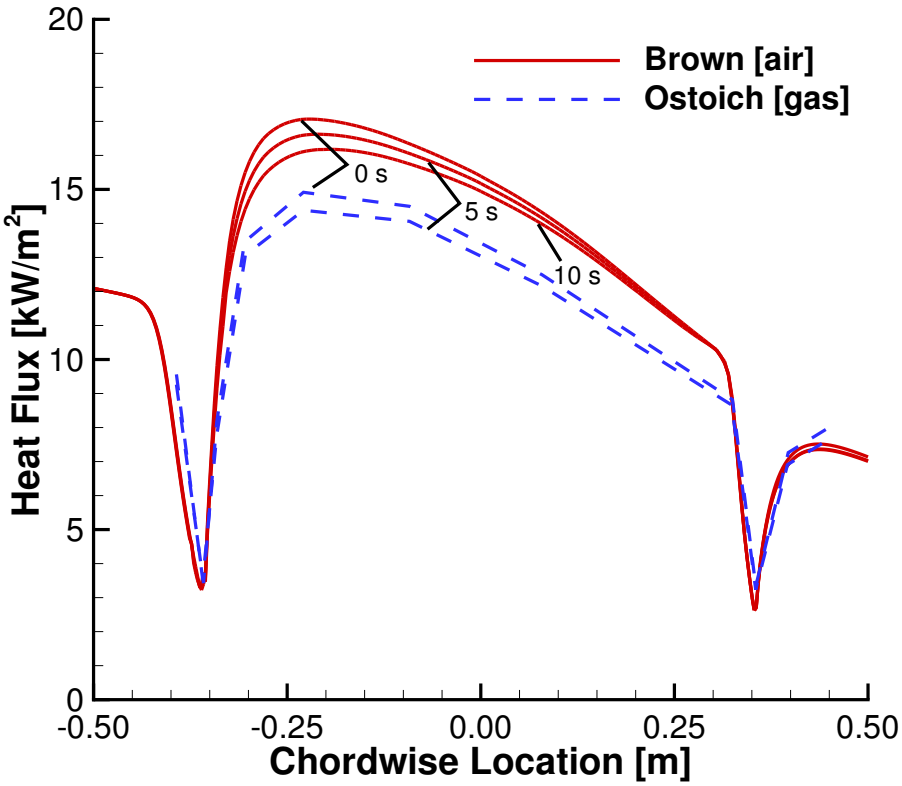


Figure V.12: Variation of the surface heat flux over the centerline of the spherical protuberance.

and air, which more closely match the experimental setup. This significantly effects both the ratio of specific heats, γ and the Prandtl number, which was assumed to be 0.7 in the present analysis. Regardless, the trends of both simulations are nearly identical, both in time and space.

Figure V.13 shows the temperature of the dome and surrounding ceramic tiling at $t = 5$ s. The vortex created on the sides of the spherical protuberance causes the greatest increase in temperature over the entire domain. The leeward half of the dome itself also heats up significantly, though to a maximum temperature of 321.6 K,

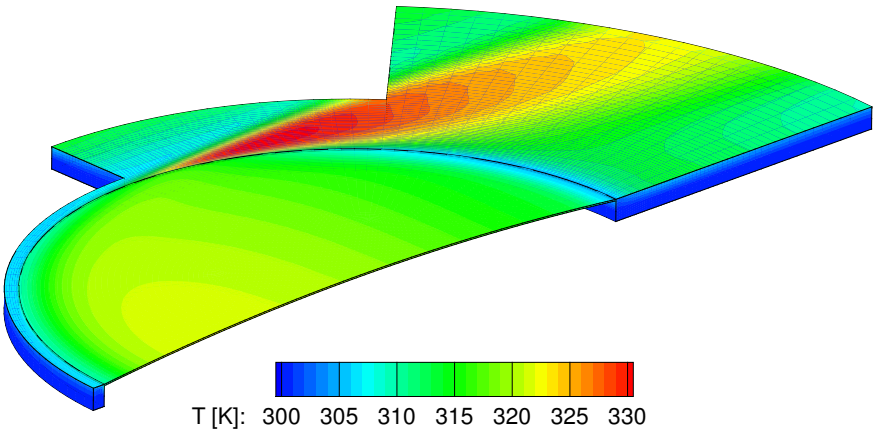


Figure V.13: Temperature in the thermal-diffusion domain at $t = 5$ s.

compared to the 330.8 K seen in the vortex region. This matches the trends seen in [81].

Figure V.14 shows the temperature of the centerline of the thermal-diffusion domain at $t = 1$, 5, and 10 s. There is very little variation through the thickness of the dome, though significant variation can be seen through the thickness of the ceramic tiling in front of and behind the dome. The thermal gradients present at the leading and trailing edges of the dome indicate that the effect of thermal contact between the aluminum dome and ceramic tiling is significant. All of these effects are mirrored in the analysis by [81].

Although there are significant differences between the current analysis and the simulations performed by [81], most of the discrepancies are likely due to the use of an ideal gas model within the aerodynamic solver. All of the trends seen in the previous analysis were mirrored here, which is a strong indication that the methodology used is accurate and trustworthy for further analysis.

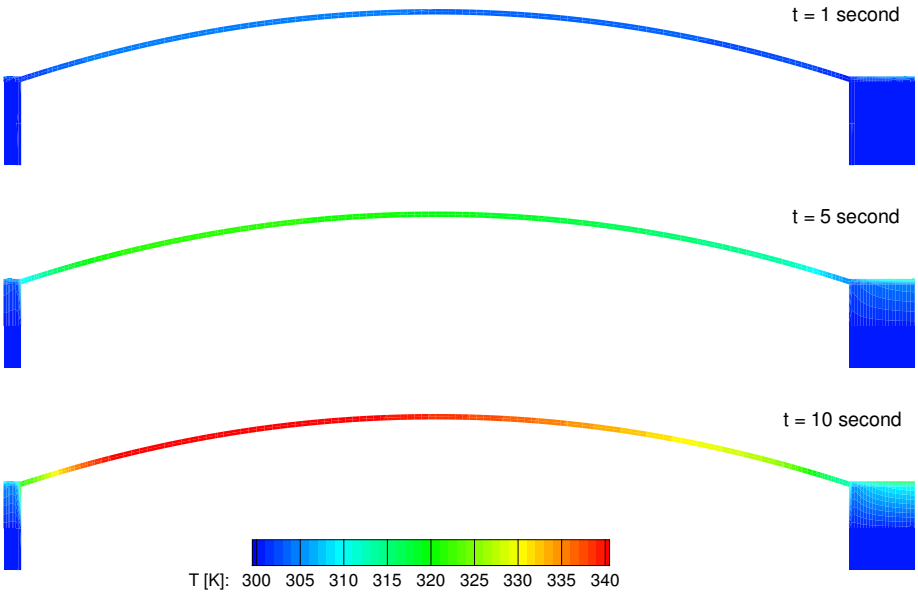


Figure V.14: Temperature along the centerline of the thermal-diffusion domain at $t = 1$, 5, and 10 s.

V.2. Heat Flux on a Double-Wedge Airfoil with Specified Internal Heat Flux

This section investigates the resulting wall temperature on a double-wedge airfoil with a specified internal heat flux. The external geometry of the airfoil was given in Section IV.2. The internal geometry is specified in Figure V.15. The outer surface of the airfoil is composed of Ti_2AlC, a MAX phase ceramic. The middle layer and the spars are NiTi, an alloy known for its shape memory properties. The interior layer is a generic polymer matrix composite. The properties used for the simulation of these materials is given in Table V.3. The resulting thermal resistance through the airfoil skin is 0.65 m^2K/kW. The change in temperature through the airfoil skin is less than 20 K for the cases investigated in this section.

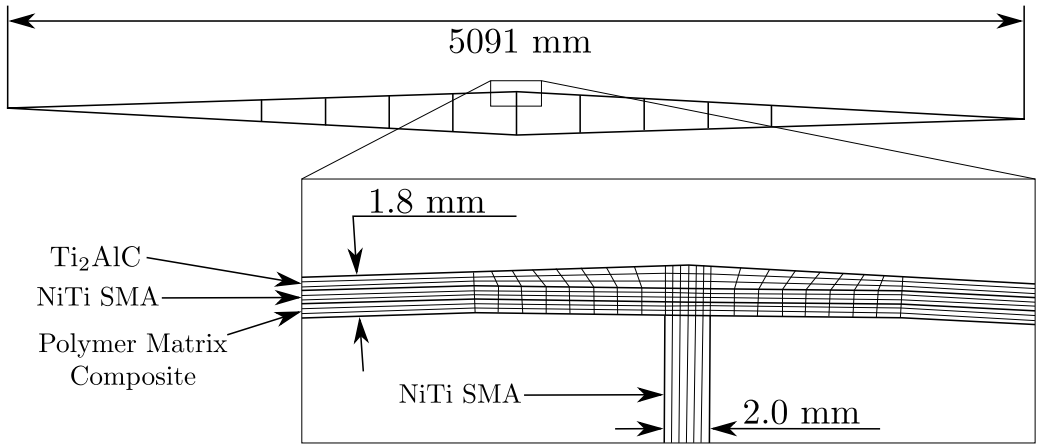


Figure V.15: Geometry of the structure used in the aerothermal study. Each of the three regions is 0.6 mm thick, for a total thickness of 1.8 mm. The spars are 2.0 mm thick.

The external wall temperature was calculated for two flight conditions: (a) Mach 5 at an altitude of 45 km and (b) Mach 8 at an altitude of 25 km. In both cases, the airfoil had an angle of attack of 0.6 degrees, which caused a slight increase in temperature on the pressure side compared to the suction side.

The heat flux necessary to maintain an external temperature of 800 K for both flight conditions is shown in Figure V.16. However, this heat flux profile is complex, and cannot reasonably be obtained with a real-world cooling system. To reduce the complexity of the proposed cooling system, only piecewise constant cooling profiles were investigated herein. Cooling is first applied uniformly over the entire airfoil at several different levels. Then, the cooling at the leading edge is significantly augmented. The effectiveness of each cooling system is evaluated in terms of reduction of maximum temperature and minimization of temperature variation over the airfoil. This analysis was done by coupling the flow and structural solvers described in

Table V.3: Material thermal properties.

	Thermal Cond., k [W/mK]	Density, ρ [kg/m^3]	Spec. Heat, c [J/kgK]
Ti_2AlC	36.3	4110	809.8
SMA[a]	18.0	6450	836.8
PMC[b]	1.0	1600	1500.0

[a] NiTi shape memory alloy. [b] Polymer matrix composite.

Section III.

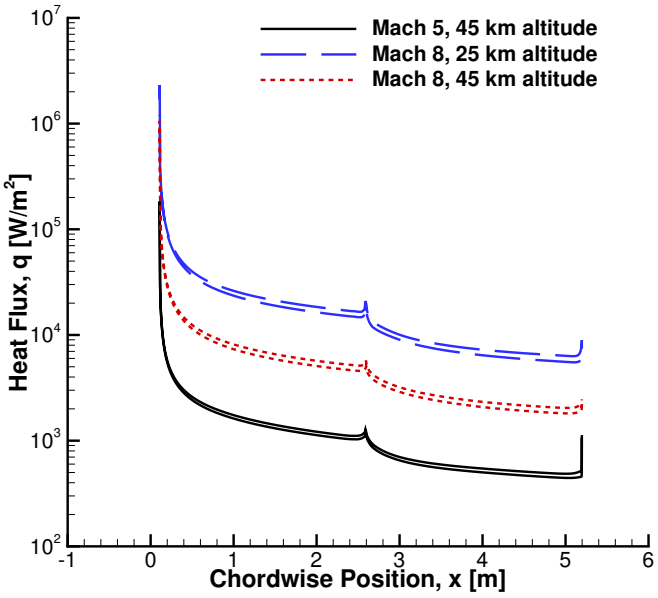


Figure V.16: Heat flux variation with chordwise position along a double-wedge airfoil. Flow conditions are Mach 5 and 45 km altitude and at Mach 8 and 25 km altitude with a wall temperature of 800 K.

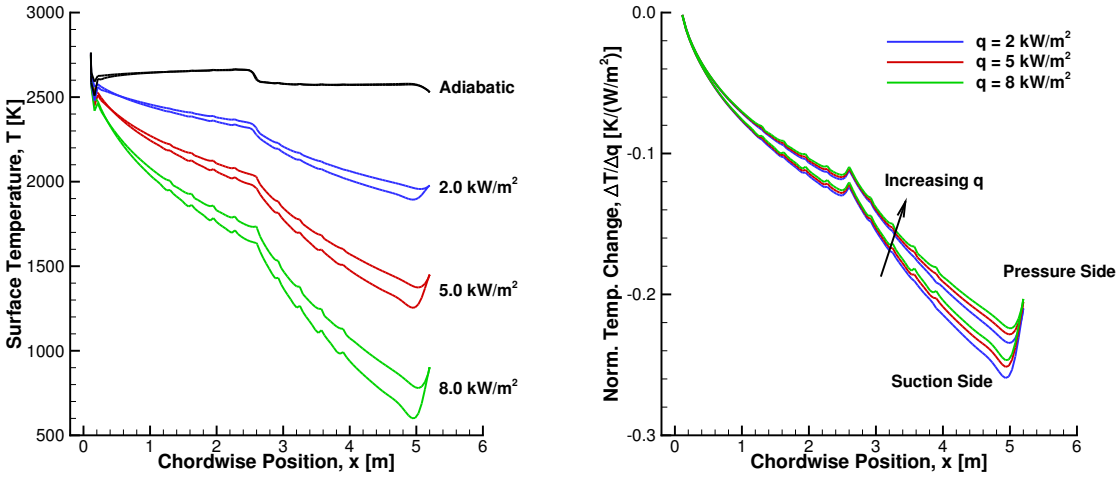
V.2.1. Constant Interior Cooling

In this section, chordwise constant cooling is applied to the interior of the skin over the entire length of the airfoil. While this is the most straightforward means of cooling the airfoil, it is also the least efficient.

There is a strong correlation between the temperature on the airfoil and the applied cooling rate. Figure V.17a shows the airfoil temperature with respect to cooling rate. Figure V.17b gives the normalized temperature change, which is defined as $\Delta T/\Delta q$. ΔT is the decrease in wall temperature due to an increase in the cooling heat flux Δq. The normalized temperature change is almost identical for all three cooling cases, though there is a slight trend towards higher cooling levels being less effective. This is likely due to increased density in the boundary layer as the temperature drops.

Figures V.18a and V.18b show profiles of normalized temperature reduction for flow at Mach 5 and 45 km altitude and flow at Mach 8 and 25 km altitude, respectively. For both cases, the temperature at the leading edge was only slightly affected by the applied cooling, and was close to the stagnation temperature of the flow. The constant interior cooling also led to large temperature variations over the airfoil. As shown in Figures V.18a and V.18b, the exterior wall temperature varied from 1341 K to 310 K for an interior cooling rate of 1.0 kW/m^2 at Mach 5 and an altitude of 45 km, and from 2754 K to 1238 K for an interior cooling rate of 5.0 kW/m^2 at Mach 8 and an altitude of 25 km.

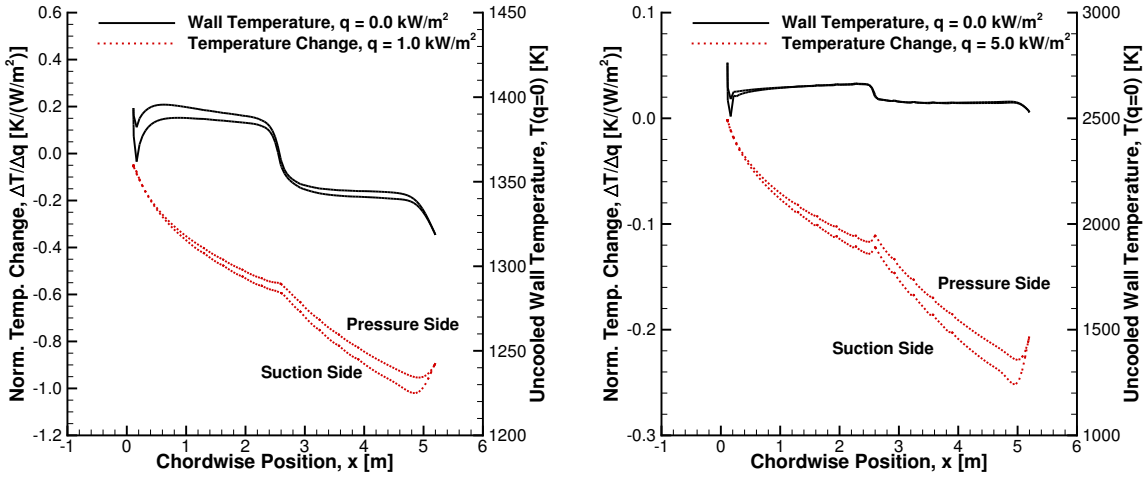
The applied cooling rate was then varied between 0.2 and 1.0 kW/m^2 for the Mach 5 case and between 2.0 and 8.0 kW/m^2 for the Mach 8 case. In these cases, in-



(a) Wall temperature.

(b) Normalized Temperature Change.

Figure V.17: Chordwise variation of wall temperature and normalized temperature change as cooling is increased. Flow is at Mach 8, 25 km altitude.



(a) Mach 5, 45 km altitude.

(b) Mach 8, 25 km altitude.

Figure V.18: Chordwise variation of normalized temperature reduction and wall temperature without cooling.

creasing the applied cooling rate had almost no effect on the normalized temperature reduction, which varied by less than 3%.

The application of constant interior cooling either does not adequately cool the structure near the leading edge or overcools in other regions. Additionally, the constant cooling results in large temperature variations in the chordwise direction.

V.2.2. Augmented Leading Edge Cooling

In this section, high cooling heat fluxes were applied to the leading edge region of the thermal protection system, and moderate cooling was applied over the rest of the airfoil. Two cooling heat flux profiles were considered herein. Profile A consisted of a cooling rate of 7.5 kW/m^2 over the first 4% of the airfoil and a cooling rate of 0.4 kW/m^2 over the rest of the airfoil. Profile B consisted of a cooling rate of 3.75 kW/m^2 over the first 8% of the airfoil and a cooling rate of 0.4 kW/m^2 over the rest of the airfoil. Figure V.19 shows the heat flux variation vs. chordwise position for profiles A and B, and for a heat flux that generates a constant wall temperature of 800 K.

Figure V.20 shows the normalized temperature reduction for the two cooling profiles applied on the interior of the airfoil. In both cases, the temperature difference is normalized by the average interior heat flux. Both the maximum temperature and the temperature variation over the airfoil were significantly reduced compared to the constant interior cooling case for comparable values of total cooling. However, the temperature variation was much more significant near the leading edge, which can result in large thermal stresses. Table V.4 summarizes the results for the constant

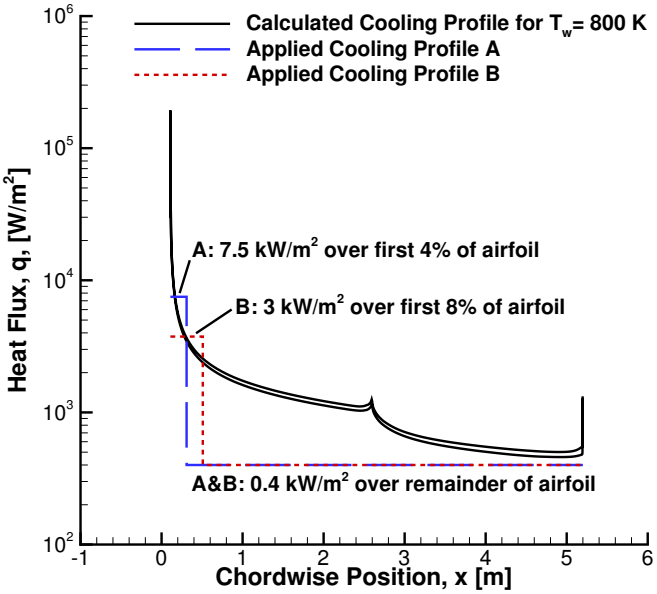


Figure V.19: Applied heat flux profiles A and B compared to heat flux calculated from a constant wall temperature of 800 K at Mach 5 and 45 km altitude.

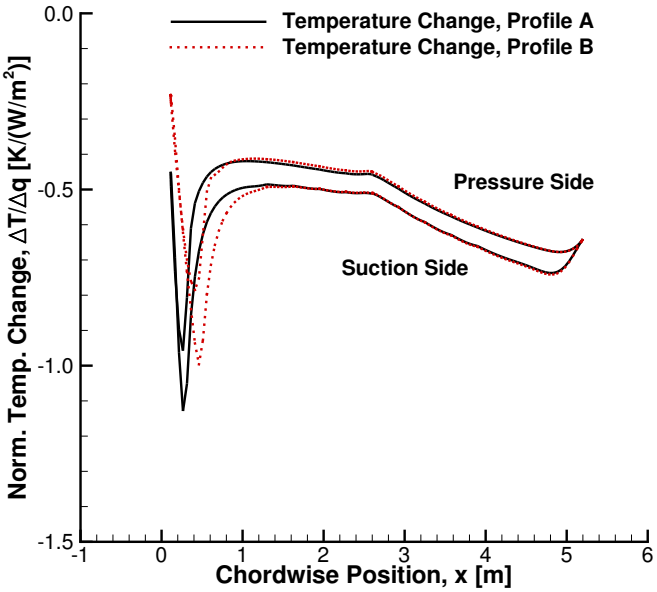


Figure V.20: Normalized temperature reduction caused by augmented leading edge cooling profiles A and B (see Figure V.19) at Mach 5 and 45 km altitude.

interior cooling case presented in section V.2.1 and the cases defined by cooling profiles A and B. While there is still significant room for optimization, the cooling distributions investigated promise the possibility of useful temperature distributions across the entire airfoil with relatively simple cooling systems.

Table V.4: Evaluation of cooling schemes.

	q_{avg}[a] [kW/m^2]	T_{max}[b] [K]	T_{min}[b] [K]	$(\nabla T)_{max}$[c] [K/mm]
Const.	1.00	1341	310	0.85
A	0.68	1038	408	4.44
B	0.66	1198	564	2.32

[a] Average cooling on interior surface of airfoil [b] Maximum (max) and minimum (min) temperature [c] Maximum temperature gradient

CHAPTER VI

AEROTHERMOELASTIC PANEL FLUTTER

This chapter presents a numerical investigation of aerothermoelastic panel flutter. The first section covers the verification of the coupled aerothermoelastic solver. Verification is accomplished by comparing the flutter response of a square panel to that described in [31]. The second section shows the use of the verified solver for the simulation of panel flutter for several configurations of deformable substructure. These configurations were motivated by panel concepts presented in [3] and [83]. The response of the panels with substructure are contrasted to theoretical homogenized panels, and the importance of simulating the full substructure is highlighted.

VI.1. Aerothermoelastic Solver Verification – Panel Flutter

The verification case for the aerothermoelastic solver was designed to test the communication between the three coupled solvers (aerodynamic, elastic, and thermal). The chosen verification case was flutter of a square panel subject to hypersonic flow and temperature variation. We will compare our results to those of Xue and Mei [31].

VI.1.1. Methodology

The response of a square panel subject to hypersonic flow was studied herein using the structural elastic and thermal solvers coupled with either a piston theory or an Euler aerodynamics solver. The results of our analysis, specifically the bound-

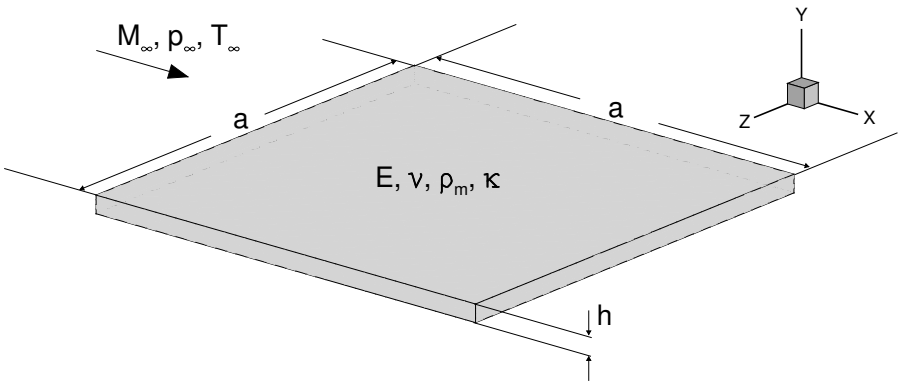


Figure VI.1: Neat panel for aeroelastic panel flutter.

aries between panel responses, will be compared to those of Xue and Mei [31], who performed eigenvalue analysis on panels.

Xue and Mei [31] used the Finite Element Method defined by von Kármán's plate theory for an analysis of the aeroelastic eigensystem. In contrast, the analysis described herein uses the Finite Element Method for brick elements using the Total Lagrangian formulation for unsteady aerothermoelastic analysis. Xue and Mei used simply supported boundaries on all four sides of the panel. To implement a similar boundary condition, the intersection of the midplane of the panel and the side surfaces was restrained in displacement ($\vec{u} = \vec{0}$). A diagram of the panel is given in Figure VI.1.

The panel was discretized using 722 20-node hexahedral brick elements (two elements through the thickness, and 19 elements along the width and length), as shown in Figure VI.2. The panel material properties, properties of the airflow, and the panel dimensions are all given in Table VI.1.

To evaluate the effect of temperature on the stability of the panel, the panel was

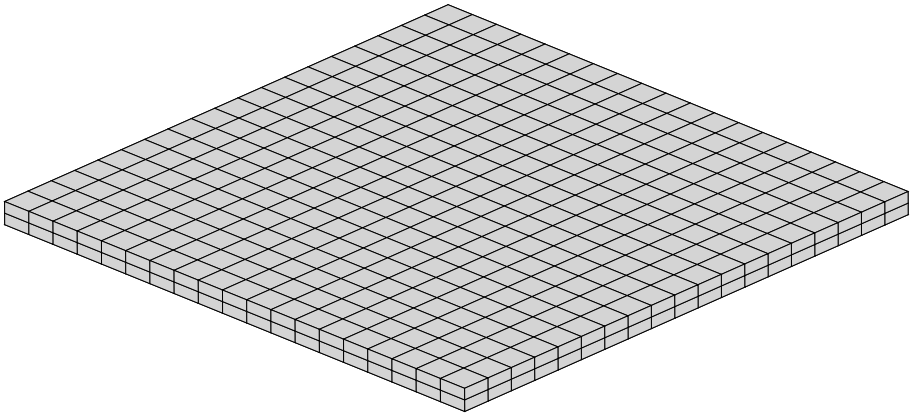


Figure VI.2: Mesh for neat panel.

Table VI.1: Parameters for aeroelastic panel flutter.

Material Properties		Air Properties		Panel Dimensions	
E	200 GPa	M_∞	5.0	a	0.5 m
ν	0.3	p_∞/λ	2560 Pa	h	5.0 mm
ρ_m	7850 kg/m^3	ρ_∞/λ	29.7 g/m^3	N_x‡	19
α	10.5·10^{-6}/K	T_∞	300.0 K	N_y‡	2
κ	237 MW/m^{2}K	λ†	0-600	N_z‡	19

† Non-dimensional pressure parameter ‡ Number of elements in x, y, and z

slowly heated. To match the analysis of Xue and Mei [31], the spatial distribution of temperature should remain uniform at every time, which matches the analysis of Mei [31]. To mimic a uniform temperature distribution with our solver, a heat flux boundary condition was applied on the top of the panel in the thermal diffusion solver, and the thermal conductivity of the panel material was increased by a factor of 10^6. This was found to reduce the temperature variation in the panel to below 0.2K throughout the entire simulation.

Based on the analytical loading of a panel [84], the critical temperature for a rectangular panel with length a, width b, and height h is:

$$\Delta T_{cr} = \frac{\pi^2 h^2}{12\alpha(1+\nu)}\left(\frac{1}{a^2}+\frac{1}{b^2}\right) \tag{6.1}$$

VI.1.2. Comparison of Time-domain and Eigensystem Analyses

The method of Xue and Mei [31] produced the boundaries of different panel behaviors (stable, limit-cycle, buckled, and snap-through). In constrast, the time-domain aerothermoelastic solver produced a time history of the displacement of each node of the panel. Each panel behavior given by Xue and Mei [31] can be related to panel response in the time history of displacement. A stable response is one in which vibrations in the panel are damped out over time. Limit-cycle oscillations (LCO) are vibrations which grow exponentially, but are bounded by the nonlinear treatment of the structural solver. A buckled response is static over time, but is deflected from the undeformed position. Finally, snap-through is a chaotic response which oscillates between two buckled equilibrium locations.

The boundary between stable panel response and LCO was determined through

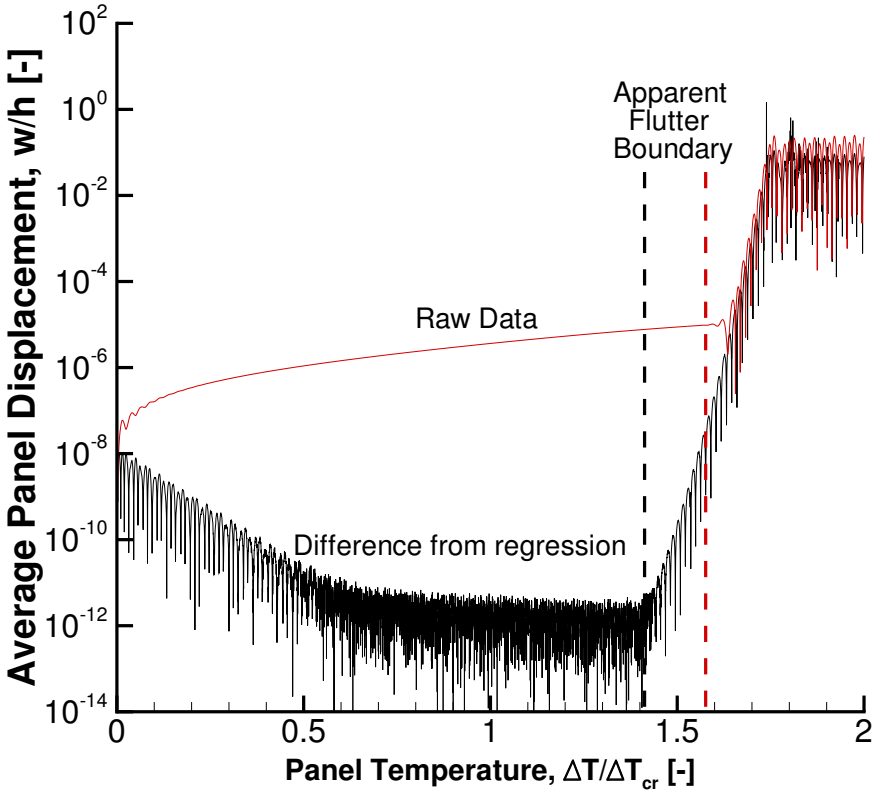


Figure VI.3: Comparison of raw averaged displacement to variation about a moving regression curve.

the following methodology. First, the average displacement, velocity, and kinetic energy of the panel were calculated over time. Then, a moving polynomial regression was performed to determine the mean motion of the panel due to thermal stresses. The mean motion of the panel, however small, overshadowed the initial vibrations caused as the panel transitioned to LCO, as seen in Figure VI.3. Mean motion was therefore discarded to isolate the vibration of the panel. This process is discussed further in Appendix E. Finally, the growth or decay of panel vibration is used to determine the transition from a stable response to LCO.

The boundaries between harmonic LCO, non-harmonic LCO, and snap-through were identified with the phase-plot of average out-of-plane displacement and velocity and the variation of the panel kinetic energy over time. Repeating patterns in the phase plot are indicative of LCO. If no repeating pattern exists, the phase plot is indicative of snap-through. Harmonic LCO is caused by sinusoidal motion of a single predominant vibrational mode, and can be characterized by an elliptical phase plot and a smoothly oscillating kinetic energy profile. Non-harmonic LCO is characterized by a complex repeating phase plot [85] (slowly growing in size due to the temperature increases over time) and a jagged kinetic energy profile. Snap-through is characterized by a chaotic phase plot, typically switching between the two stably buckled locations at random intervals.

Figure VI.4 demonstrates the process of regime identification for panel flutter at $\lambda = 234$. The transition to non-harmonic LCO occurs almost immediately after the transition to harmonic LCO, so stable harmonic LCO was not witnessed. Snap-through can also be seen in the kinetic energy profile as a switching between exponential decay and sudden resurgence to LCO energies. Finally, the transition to buckling occurs when the panel no longer receives enough energy from the flow to snap between the buckled states and oscillations around one of the buckled states are damped out. The average panel displacement will continue to increase as the temperature is increased.

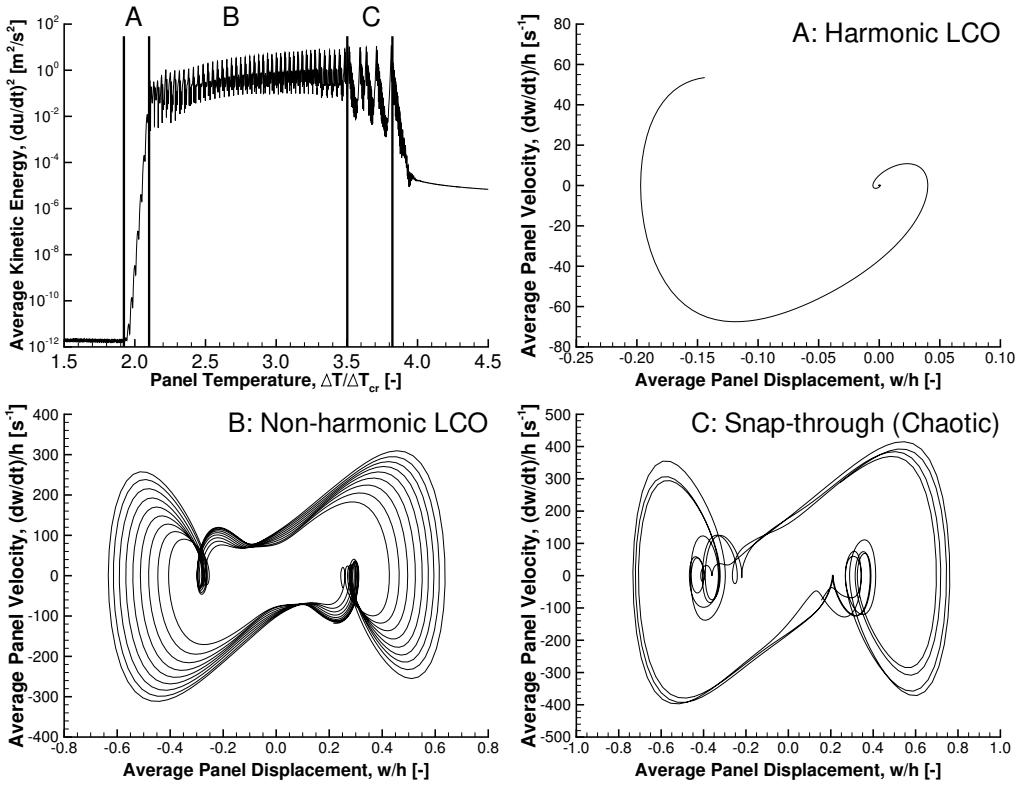


Figure VI.4: Panel flutter regime identification for panel flutter at $\lambda = 234$ and gradually increasing temperature. Top left: Average panel kinetic energy as panel temperature increases. Top right: Phase plot for exponential growth and harmonic LCO. Bottom left: Phase plot for non-harmonic LCO. Bottom right: Phase plot for snap-through.

VI.1.3. Comparison of Panel Response Boundaries

The panel response boundaries are plotted in ΔT–λ space for a given value of aerodynamic damping (g_a constant). It is assumed that the panel behavior is dependent only on the panel temperature difference ΔT, the nondimensional pressure parameter λ, and the non-dimensional damping parameter g_a. This is not a complete picture of the dynamics of panel flutter, as the time-history of temperature and pressure can affect the panel response [46]. However, the present analysis, which uses relatively slow temperature ramping, generates results similar to the eigensystem analysis in [31].

Aerodynamic damping is realized in piston theory through the inclusion of the plate velocity, $\frac{\partial w}{\partial t}$, which is translated to a nondimensional damping parameter, g_a. The analysis performed in Mei [31] assumes no aerodynamic or structural damping. Rather, flutter occurs due to the coalescence of two separate vibration modes as the non-dimensional pressure parameter λ is increased.

A small level of damping is helpful in removing spurious oscillations, while still allowing the dominant flutter modes to grow. A comparison of the flutter with two levels of damping ($g_a = 0.1$ and $g_a = 1.0$) is given in Figure VI.5. A higher damping parameter leads to delayed onset of LCO and also delays the onset of snap-through. In order to match the simulations performed by Mei [31], the lower value of $g_a = 0.1$ will be used for all piston theory simulations.

Damping in the Euler solver is generated as a natural consequence of moving boundaries. The aerodynamic damping in simulations using the Euler solver can be determined by the air properties in Table VI.1 and can be compared to the piston

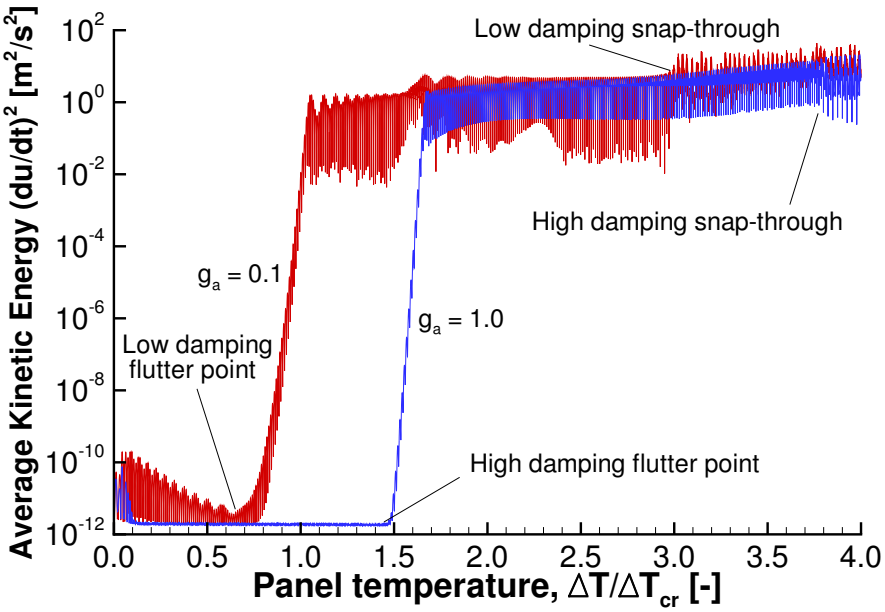
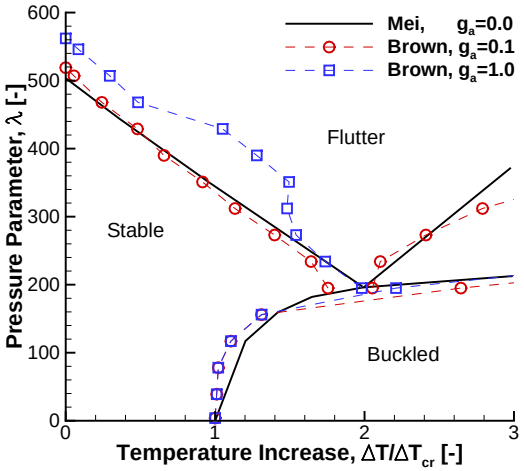


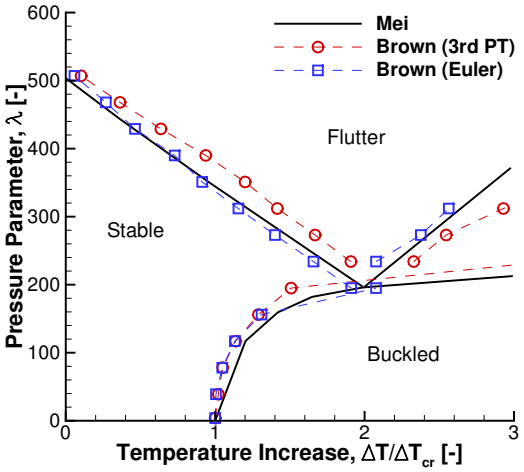
Figure VI.5: Time variation of average kinetic energy for panel flutter with two values of damping.

theory aerodynamic damping through equation 2.8. In this analysis, the damping can be calculated as $g_a/\lambda = 8.54\text{E}{-}04$.

The data from simulation at an array of λ values was accumulated to generate regime boundaries in $\Delta T - \lambda$ space, as seen in Figures VI.6a and VI.6b, which are comparisons of the calculated boundaries in the current analysis to those calculated by Mei [31]. In all cases, the boundaries calculated in the current analysis match those by Mei quite closely, though as damping increases, we expect to stray from his results.



(a) 1st order piston theory with non-dimensional damping parameter $g_a = 0.1$ and $g_a = 1.0$.



(b) 3rd order piston theory and inviscid RANS solver with damping calculated by freestream conditions.

Figure VI.6: Comparison of panel flutter regime boundaries. Regimes were calculated with various aerodynamic models, using [31] as a baseline.

VI.1.4. Response at High Temperature and Pressure

One of the major benefits of simulating in the time domain is the ability to delve more deeply into panel response beyond harmonic LCO. A study of the dynamic response was performed to determine when the panel transitioned from harmonic LCO to non-harmonic LCO. However, the current methodology may be insufficient to pinpoint when these transitions occur. Non-harmonic flutter can be explained by the excitation of additional modes [86]. Because the shape of the phase plot and the kinetic energy of the panel are initially dominated by the primary modal response, the beginning of growth of additional modes will be initially hidden, and cannot be used as the metric for transition from harmonic to non-harmonic LCO.

In the present study, the transition was defined as the point when the non-harmonic disturbances were responsible for approximately the same amount of kinetic energy as the harmonic motion, as indicated in Figure VI.4. Proper orthogonal decomposition or some other modal analysis may prove more suited for determining the transition between these two vibrational states, as the primary mode of vibration could be identified and removed from the analysis.

At high values of ΔT, an asymmetric flutter mode becomes apparent. As the majority of simulations performed in literature were assumed to be symmetric [31, 43], this mode has not been seen in previous studies. However, by looking at the average displacement of the panel in the direction transverse to the flow, a clear region of exponential growth was determined as the panel was heated. A comparison between the symmetric and asymmetric flutter modes is shown in Figure VI.7. The region of maximum displacement for the symmetric mode moves directly from left to right, while the region of maximum displacement also moves from top to bottom.

The region of asymmetric panel flutter also overlaps with the chaotic region. While some portion of the snap-through phenomenon occurs in a symmetric fashion, as the temperature continues to increase, it too becomes asymmetric. All of the calculated regime boundaries are shown in Figures VI.8a and VI.8b. Note especially that additional damping significantly delays both snap-through and the asymmetric mode, while only mildly affecting the onset of buckling and LCO.

The key results from the study were twofold. First, the aerothermoelastic solver was verified, as panel flutter regime boundaries found using the solver closely matched those published in previous studies. Additionally, simulations show that at high

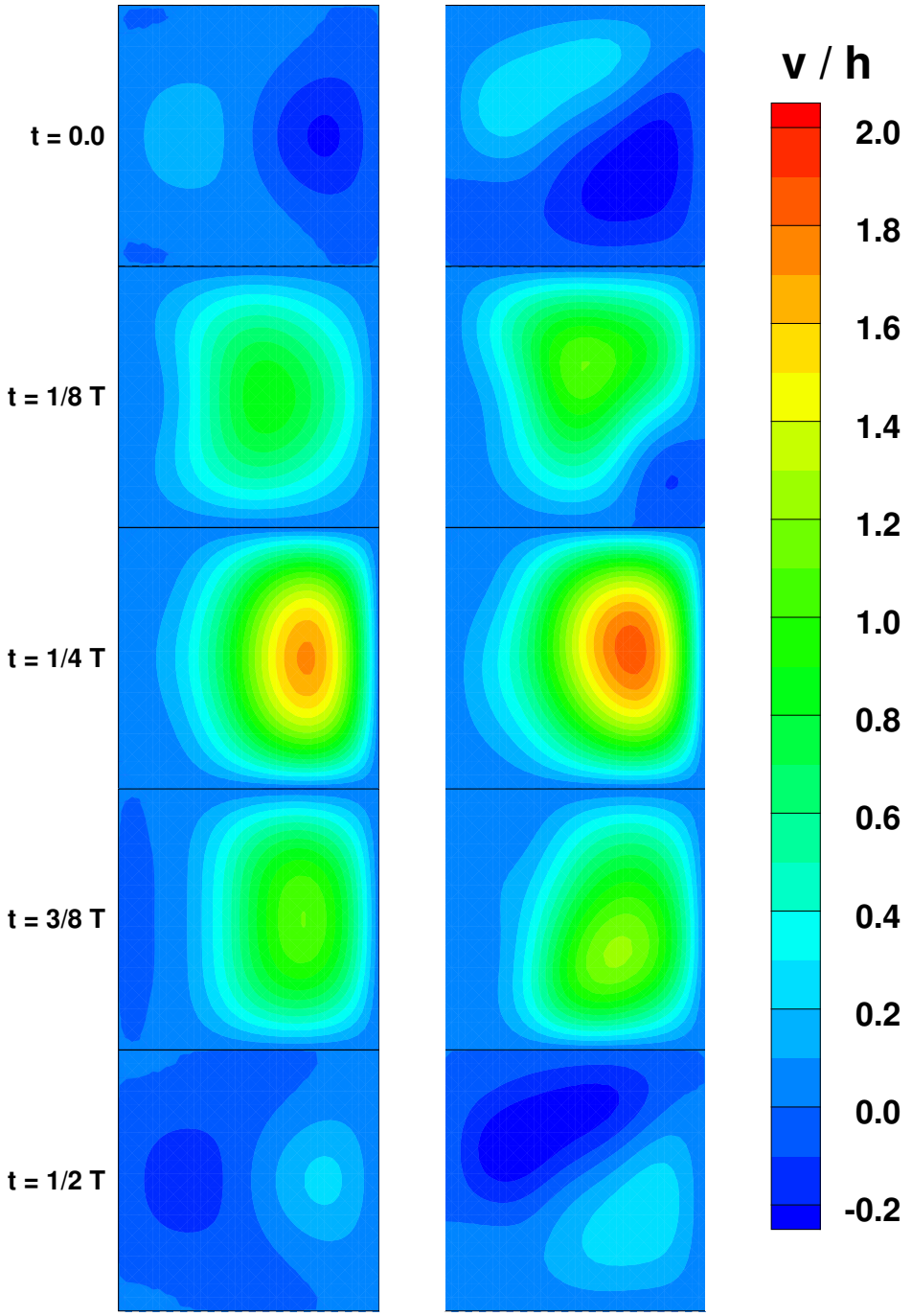
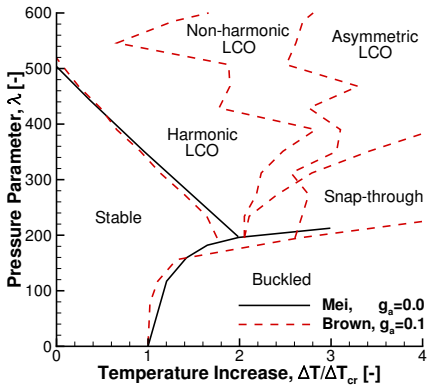
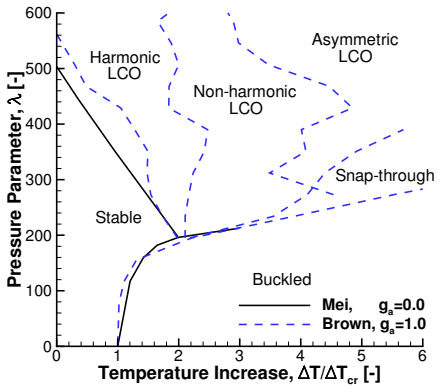


Figure VI.7: Panel flutter modes at $\lambda = 546$ over one half period. Left: symmetric modes, $\Delta T/\Delta T_{cr} = 3.0$. Right: asymmetric modes, $\Delta T/\Delta T_{cr} = 3.5$.



(a) First order piston theory, g_a=0.1.



(b) First order piston theory, g_a=1.0.

Figure VI.8: High-temperature panel response boundaries with first order piston theory and non-dimensional damping parameter g_a specified.

temperatures, the assumption of symmetry breaks down, and a non-symmetric flutter mode begins to dominate. In order to accurately simulate panel response at elevated temperatures, symmetry cannot be assumed.

VI.2. Panel Flutter with Substructure A

This section investigates panel flutter with a substructure configuration motivated by a preliminary panel design in [83]. This substructure is denoted herein as the panel with substructure A. Section VI.2.1 defines substructure A, Section VI.2.2 discusses the modeling and discretization, Section VI.2.3 shows analysis with ABAQUS, Sections VI.2.4 and VI.2.5 give the rationale and methodology behind the rescaling of the panel flutter regime boundaries, and and Section VI.2.6 gives the flutter boundaries for the panel with substructure A.

VI.2.1. Definition of Substructure A

The design in [83] consisted of a carbon-carbon panel with an array of uniaxial blades as stiffeners in the flow direction. The panel was secured to the vehicle by means of a vertical stringer that was bolted to a vertical component of the vehicle substructure. Additionally, the panel was significantly thicker near the vertical stringer.

The panel with substructure A consisted of a panel and four stringers, located on each edge of the panel. The panel had a thickness, h, of 5 mm and a width, a, of 0.5 m. The substructure was defined with two additional parameters, h_s and a_s, which were the stringer thickness and width, respectively. a_s was taken as 0.05 a, and h_s was varied from h to $5h$. The nodes along the line defined by the intersection of the panel midplane and the side surfaces, as shown in Figure VI.9, were restrained in displacement ($\vec{u} = \vec{0}$).

VI.2.2. Discretization

The panel and substructure were discretized using 20-node hexahedral brick elements, as seen in Figure VI.10. 968 elements were used for the panel (two through the thickness and 22 along each side of the panel). 320, 640, and 1120 elements (2 through the width of the stringer, 80 along the perimeter, and 2, 4 and 7 through the stringer thickness) were used to define the substructure with $h_s = h$, $2h$, and $5h$, respectively. Elements were clustered near the stringer locations.

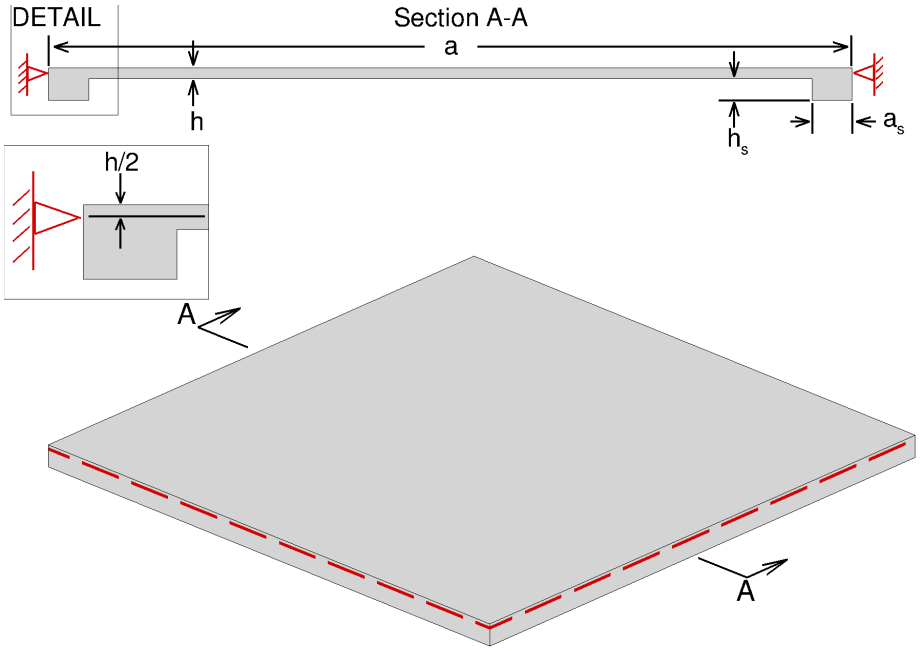


Figure VI.9: Structural domain for aeroelastic panel flutter with stringers along panel border. Nodes (not shown) along dashed line in bottom view were restrained in displacement ($\vec{u} = \vec{0}$).

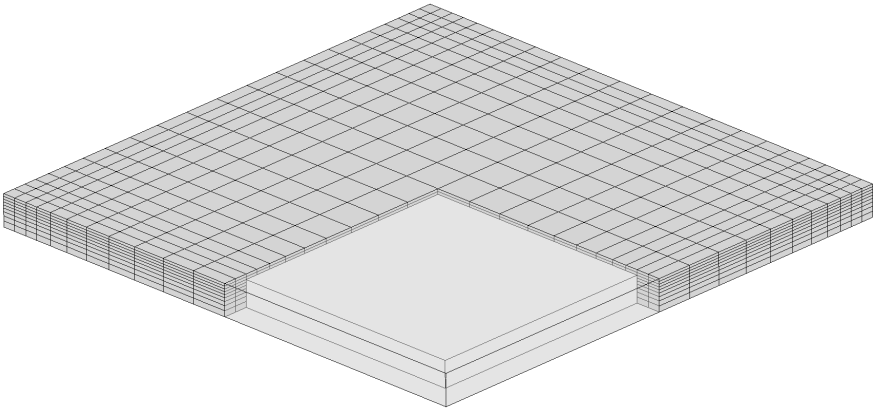


Figure VI.10: Structural mesh used for the panel with substructure A. The stringer thickness was five times the panel thickness ($h_s = 5h$). 2088 hexahedral elements were used for this mesh.

VI.2.3. Analysis of Buckling, Vibration, and Bending

Mei [31] demonstrated analysis the interaction of the buckling and vibrational modes and the aerodynamic loads through frequency analysis. The aerodynamic influence on the mode shapes was related to the slope of the top surface of the panel with respect to the flow direction. As the substructure changed, the *in vacuo* vibration mode changed, which in turn modified the effect of the aerodynamic forces on the panel vibration.

The buckling modes, vibration modes, and bending response of the panel with substructure A were calculated in ABAQUS and compared to the buckling modes, vibration modes, and bending response of the panel without stringers. The bending stiffness of the panel was determined by measuring the maximum displacement in response to a uniform pressure load on the top surface of the panel.

The primary modes of *in vacuo* buckling and vibration are shown for $h_s = 0$, 1, 2, and $5h$ in Figures VI.11 and VI.12, respectively. In each case, the mode shape remained qualitatively similar as the stringer thickness was increased. The primary change was a decrease in the twist of the panel about the restrained nodes. In other words, as the stringer thickness increased, the rotation of the panel became more restricted along its edges.

The panel response to a pressure load is shown in Figure VI.13. As the stringer thickness increased, the deformation of the panel decreased. The reduced deformation was caused by reduction in the twist along the edge of the panel. The stringer acted as a shaft in torsion, increasing the overall torsional rigidity about the restrained nodes. The stringer torsional rigidity was defined as the moment caused by

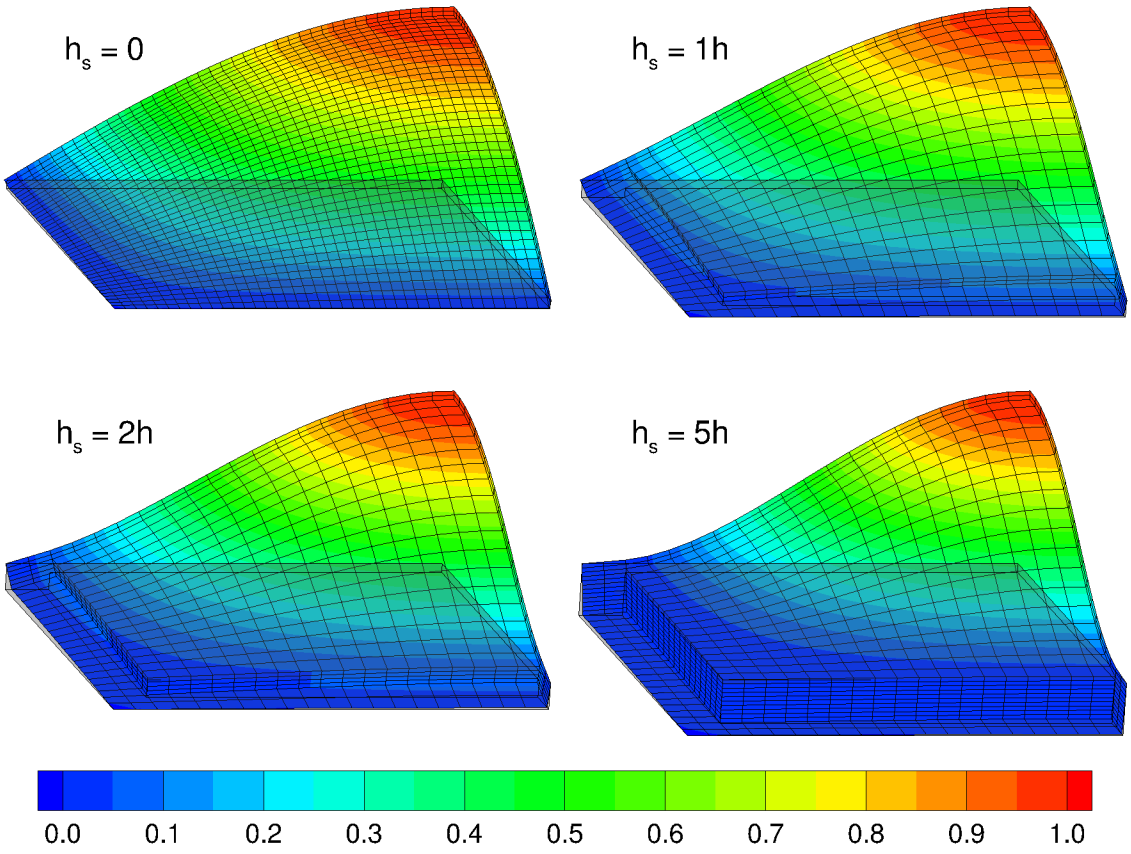


Figure VI.11: Buckling modes for one quarter of the panel with substructure A. Contour levels show out-of-plane displacement mode.

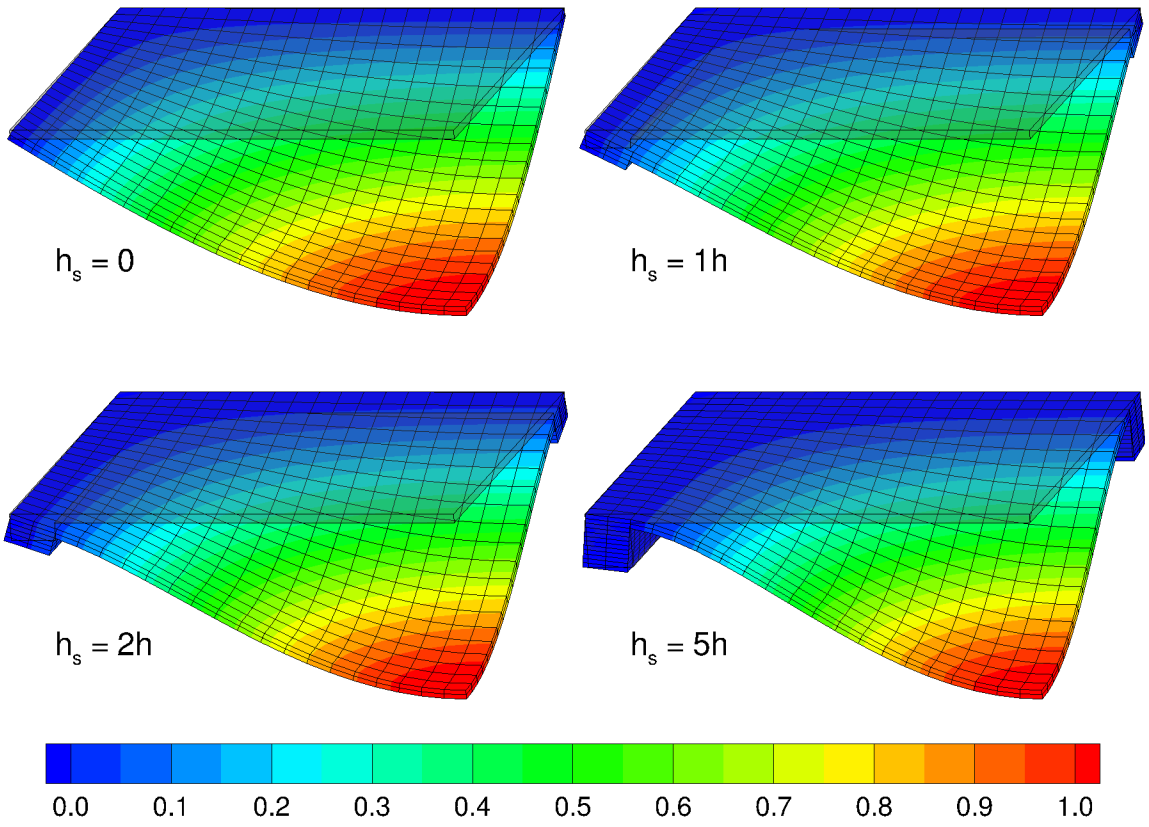


Figure VI.12: Vibrational modes for one quarter of the panel with substructure A. Contour levels show out-of-plane displacement mode.

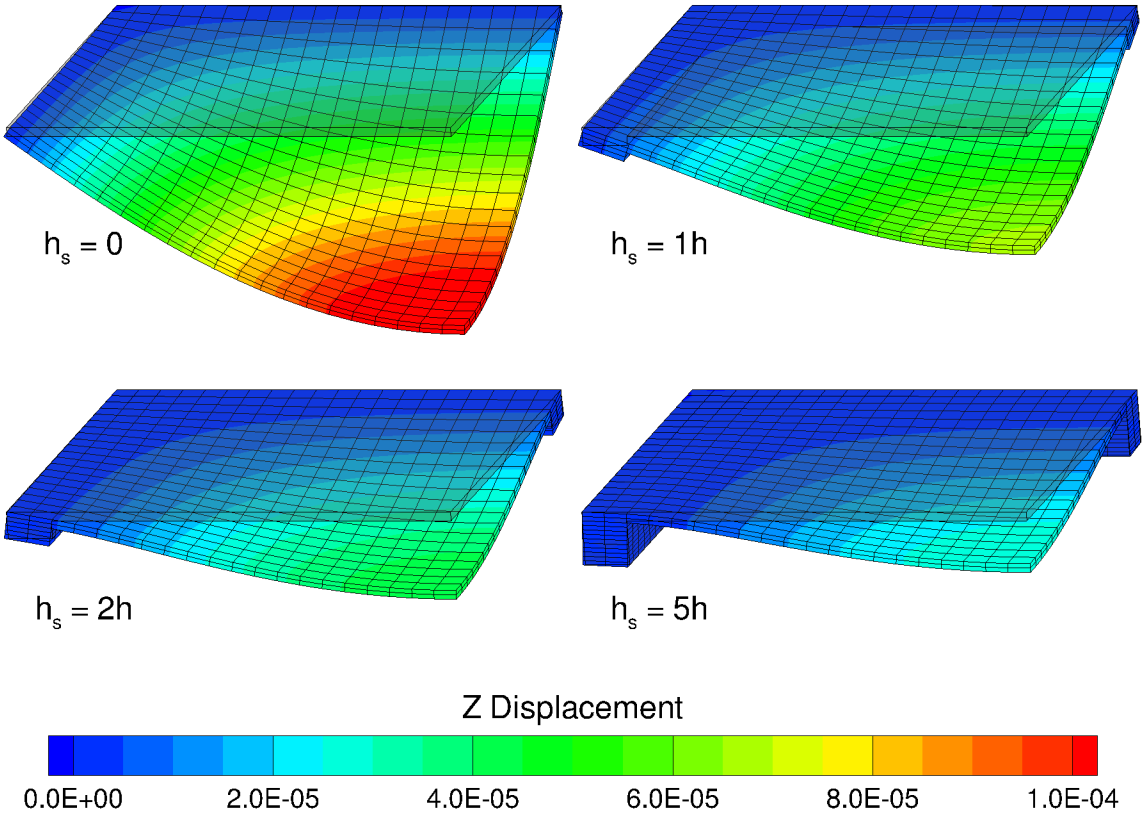


Figure VI.13: Displacement due to a pressure load of 1 kPa for one quarter of the panel with substructure A. Deformation exaggerated by a factor of 1000.

the pressure load on one quarter of the panel divided by the twist per unit length over half of the panel's width:

$$(JG)_s = \frac{\text{Pressure Moment}}{\text{Twist per Length}} = \frac{\int_0^{a/2}\int_0^{a/2} pxdxdy}{\theta_{max}/(a/2)}, \tag{6.2}$$

where θ_{max} was the twist at the midpoint of the stringer.

The buckling temperature, natural vibration frequency, bending stiffness, and torsional rigidity for each stringer thickness are given in Table VI.2. The values for each increased as the stringer thickness increased. However, past $h_s = 2h$, the

buckling temperature, vibration frequency, and bending stiffness suffered diminishing returns, while the torsional rigidity increased superlinearly with the stringer thickness for all thicknesses, as shown in Appendix F.

The stringer becomes rigid as the torsional rigidity approaches infinity. The panel with rigid stringers was modeled with a width of $a - 2a_s$ and with nodes along the entirety of each side surface restrained ($\vec{u} = 0$). The results for this panel are given in the final row of Table VI.2, which represent an upper bound for a panel with substructure A.

The diminishing returns of increasing the stringer thickness were caused by the buckling temperature, vibrational frequency, and bending stiffness asymptotically approaching those of the panel with rigid stringers. The superlinear increase for small stringer thicknesses was caused by the superlinear relationship between the stringer thickness and the torsional rigidity, which is again shown in Appendix F.

Table VI.2: ABAQUS analysis for panel with substructure A.

h_s/h	T_{cr} [K]	f [s^{-1}]	K_p[a][m/MPa]	$(JG)_s$ [kNm^2/rad]
0	35.83	95.53	8.937	2.627
1	58.56	123.8	14.97	5.418
2	87.29	153.2	22.50	11.21
3	106.6	172.2	28.13	19.61
5	124.9	190.2	34.13	41.72
10	140.5	202.7	38.82	116.5
RS[b]	151.9	216.2	44.21	∞

[a] Effective Stiffness (pressure load/displacement) [b] Rigid Stringers

VI.2.4. Rescaling Based on ABAQUS Analysis

The ABAQUS analysis was used as a rationale for further renormalization of the panel response. The generalized mass $\mathbb{M}$ and force Ξ of the primary vibrational mode were found as follows:

$$\mathbb{M} = \rho \int_{\Omega} \phi^2 d\Omega \qquad\qquad \Xi = \frac{q}{\sqrt{M^2-1}} \int_{\partial\Omega_{top}} \phi_z \frac{\partial \phi_z}{\partial x} dS,$$

where ρ is the density of the panel, ϕ is the primary vibrational mode shape, q is the dynamic pressure of the air, M is the Mach number, $\partial\Omega_{top}$ is the top surface of the panel, and ϕ_z is the out-of-plane component of the primary vibrational mode shape.

The nondimensional pressure parameter is proportional to the ratio of the generalized force and generalized stiffness:

$$\lambda \propto \Xi/K, \tag{6.3}$$

where K is the generalized stiffness of the mode. K can be rewritten as $\omega^2\mathbb{M}$, where ω is the natural frequency of the mode. Assuming that the mode shape has a maximum value of 1, Ξ and $\mathbb{M}$ are related proportionally as follows:

$$\Xi \propto \frac{qa}{\sqrt{M^2-1}} \qquad\qquad \mathbb{M} \propto \rho h a^2,$$

where q is the dynamic pressure of the air. Additionally, the natural frequency ω for a plate is

$$\omega = \sqrt{\frac{D}{\rho h a^4}},$$

where $D = \frac{Eh^3}{12(1-\nu^2)}$ is the bending stiffness of the plate.

A new nondimensional parameter, $\tilde{\lambda}$ was defined based on proportionality of λ

Table VI.3: Primary mode data for panel with substructure A.

h_s/h	f^{a} [s^{-1}]	$\mathbb{M}^{\text{b}}$ [kg]	Ξ^{c}/q [m]	$\Xi/(q\mathbb{M}\omega^2)$ [1/MPa]	$\Delta\tilde{T}_{cr}$ [K]
0	95.53	0.6138	0.0625	2.790	36.1
1	123.6	0.5541	0.0587	1.765	54.2
2	153.2	0.4673	0.0544	1.240	72.7
5	190.2	0.3786	0.0490	0.895	101.2

[a] Frequency of primary mode [b] Generalized mass
[c] Generalized aerodynamic force

to modal values defined in Equation 6.3:

$$\frac{\tilde{\lambda}}{\lambda_{neat}} = \frac{\tilde{\Xi}}{\tilde{\mathbb{M}}\tilde{\omega}^2}\frac{\mathbb{M}\omega^2}{\Xi},$$

where $\tilde{\Xi}$, $\tilde{\mathbb{M}}$, and $\tilde{\omega}$ are the generalized aerodynamic force, generalized mass, and natural frequency for the panel with substructure A, and Ξ, $\mathbb{M}$, and ω are the same for the panel without stringers.

$\Delta\tilde{T}_{cr}$ was defined as the point at which the panel began buckling in the coupled solver. This provides a more consistent point of reference than the ABAQUS buckling temperature, as there were slight discrepancies in the buckling temperatures between ABAQUS and the coupled solver. These discrepancies were likely caused by the imprecise means of measuring the buckling temperature in the coupled solver.

For each stringer thickness, the frequency, generalized mass, and generalized aerodynamic force were found for the primary mode shape in vibration. These data, along with the ratio used for $\tilde{\lambda}$ are given in Table VI.3. Additionally, $\Delta\tilde{T}_{cr}$ is given in Table VI.3 for each stringer thickness.

VI.2.5. Rescaling Based on Substructure Mass

Because the stringers add mass to the total system, design studies which use mass as a cost function should consider the panel response rescaled based on the total mass of the panel and substructure. In the present analysis, this was done by defining an effective panel thickness, which was the thickness of a panel with equivalent mass to the entire system with substructure A. This effective thickness, h_{eff}, is defined as:

$$\frac{h_{\text{eff}}}{h} = \frac{m_{subA}}{m_{neat}} = \left(4\frac{h_s}{h}\frac{a_s}{a}\left(1-\frac{a_s}{a}\right)+1\right), \tag{6.4}$$

where m_{subA} is the mass of the panel with substructure A and m_{neat} is the mass of the panel without stringers.

This creates an effective pressure parameter $\bar{\lambda}$ and an effective critical temperature, $\Delta\bar{T}_{cr}$, which were defined as follows:

$$\bar{\lambda} = \frac{\rho U_\infty^2}{\sqrt{M^2-1}}\frac{a^3}{h_{\text{eff}}^3}\frac{12(1-\nu^2)}{E} \qquad \Delta\bar{T}_{cr} = \frac{\pi^2}{6\alpha(1+\nu)}\frac{h_{\text{eff}}^2}{a^2} \tag{6.5}$$

VI.2.6. Results of Panel Flutter

The panel with substructure A was analyzed using the coupled solver for 17 pressure values, from $p_\infty = 0.1$ bar to $p_\infty = 16$ bar. The analysis techniques described in Section VI.1.2 were employed to determine the behavior regime boundaries, which are plotted in Figure VI.14. The addition of stringers positively impacted the stability of the panel in both flutter and buckling. However, there was a nonlinear relationship between the stringer thickness and stability improvement.

In general, the incremental increase in stability from $h_s = 0$ to $h_s = h$ was larger

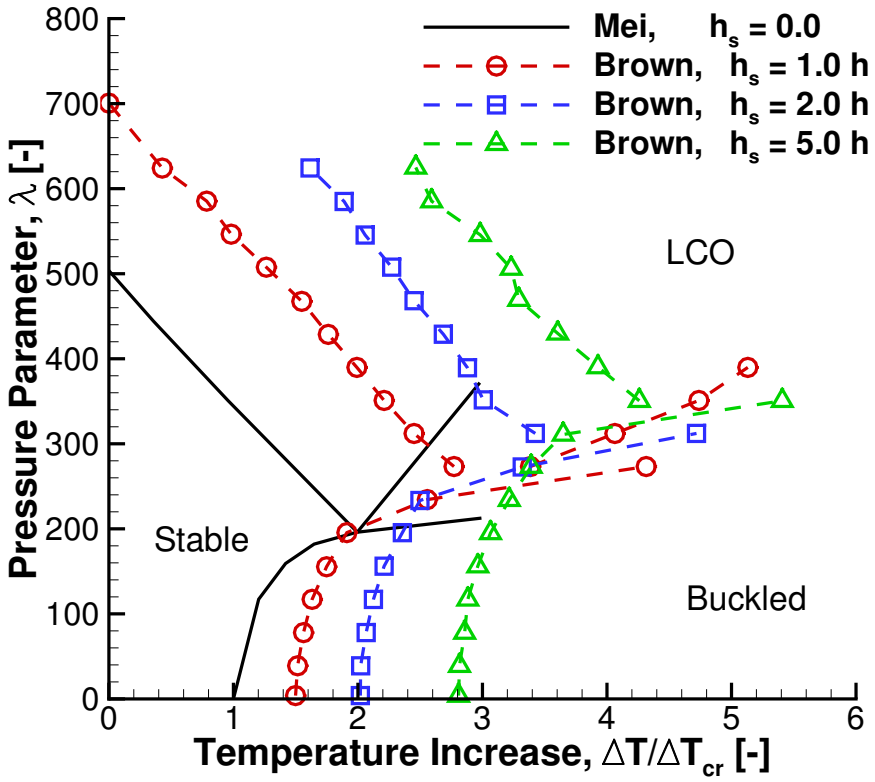


Figure VI.14: Comparison of flutter and buckling boundaries for various stringer thicknesses, without incorporating the effective thickness.

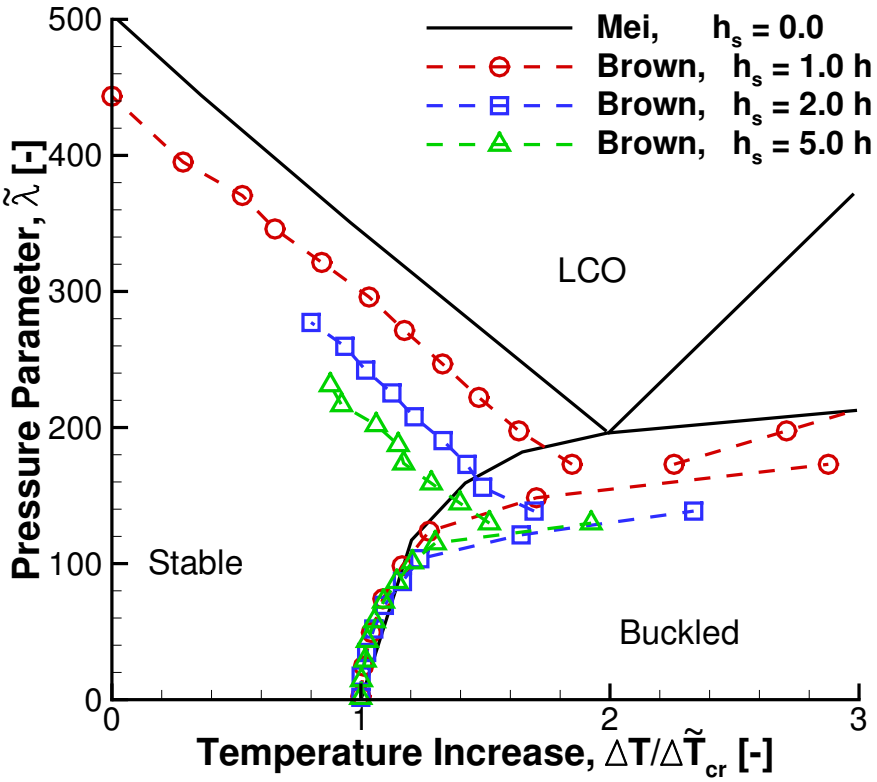


Figure VI.15: Comparison of flutter and buckling boundaries for various stringer thicknesses, using the definition for h_{eff} given in Equation 6.4.

than the incremental increase in stability from $h_s = h$ to $h_s = 2h$. The incremental increase in stability from $h_s = 2h$ to $h_s = 5h$ was far less than expected if a linear relationship held. In other words, the stability region had diminishing returns as the stringer thickness was increased. This is explained by effect of the rotational stiffness of the stringers, as discussed in Section VI.2.3.

The renormalization of the panel flutter regimes using $\tilde{\lambda}$ and $\Delta\tilde{T}_{cr}$ is shown in Figure VI.15. The modal analysis using the *in vacuo* vibration modes was clearly not able to explain the behavior of the panel with substructure A. As λ increased,

the aeroelastic modes of the panel diverge from the *in vacuo* modes. To rescale λ accurately, the generalized mass, generalized force, and frequency of the aeroelastic modes should be determined, meaning that the scaling is nonlinear in λ.

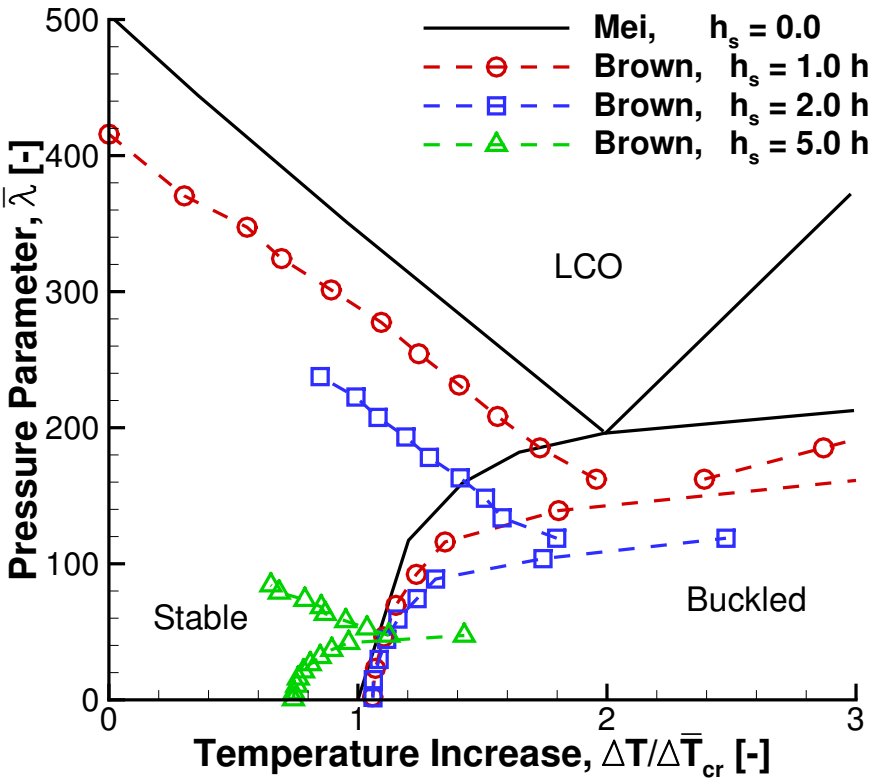


Figure VI.16: Comparison of flutter and buckling boundaries for various stringer thicknesses, using the definition for h_{eff} given in Equation 6.4.

The regime boundaries are shown rescaled based on mass in Figure VI.16. The stable region shrank significantly when rescaled based on a panel without stringers with thickness h_{eff}. The resistance to flutter was far more affected than the resistance to buckling. The additional mass required by the presence of the stringers in substructure A far outweighed the additional torsional rigidity that they provided.

VI.3. Panel Flutter with Substructure B

This section investigates panel flutter with a substructure configuration motivated by a preliminary panel design in [3]. This substructure is denoted herein as the panel with substructure B. Section VI.3.1 defines substructure B, Section VI.3.2 discusses the modeling and discretization, Section VI.3.3 shows analysis with ABAQUS, Sections VI.3.4 and VI.3.5 explain the rationale and methodology behind the rescaling of the panel flutter regime boundaries, and Section VI.3.6 gives the flutter and buckling boundaries for the panel with substructure B.

VI.3.1. Definition of Substructure B

The design in [3] consisted of a carbon-carbon panel with arrays of uniaxial blades as stiffeners in both the flow direction and the transverse direction. The panel substructure was named an "ortho-grid", due to the orthogonal placement of the blades.

The panel with substructure B consisted of four stringers centered one quarter panel width from each edge. The panel had a thickness, h, of 5 mm and a width, a, of 1 m. The stringers were again defined by h_s and a_s. a_s was defined as $0.025a$, while h_s was defined as h and $3h$. The nodes along the line defined by the intersection of the panel midplane and the side surfaces, as shown in Figure VI.17, were restrained in displacement ($\vec{u} = \vec{0}$). Additionally, the stringers were restrained in displacement along the line defined by the intersection of the center plane of each stringer and the side surfaces, as shown in Figure VI.17, in an attempt to remove the tendency of the panel to bend downward when heated due to the thermal expansion of the stringers.

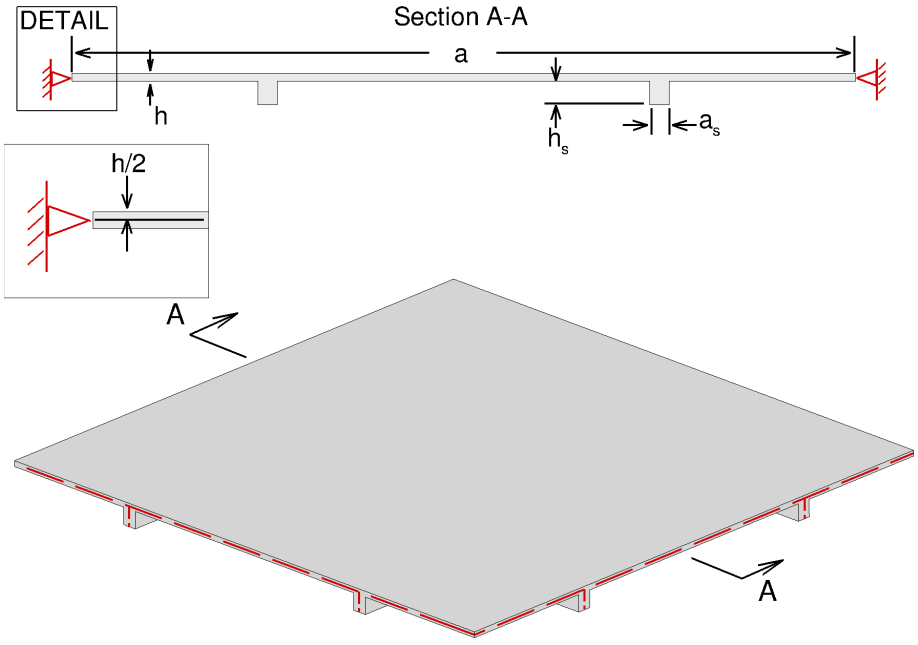


Figure VI.17: Structural domain for aeroelastic panel flutter with stringers on interior of panel. Nodes (not shown) along dashed lines in bottom view were restrained in displacement ($\vec{u} = \vec{0}$)

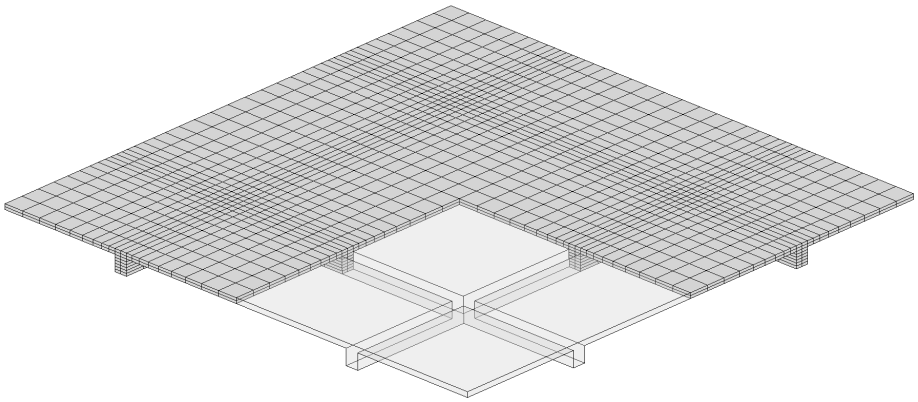


Figure VI.18: Structural mesh used for the panel with substructure B. The stringer thickness was three times the panel thickness ($h_s = 3h$). 5024 hexahedral elements were used for this mesh.

VI.3.2. Discretization

The panel and substructure were defined using 20-node hexahedral brick elements, as seen in Figure VI.18. 3200 elements were used for the panel (two through the thickness and 40 along the length and width of the panel). 608 and 1824 elements were used for the substructure with $h_s = h$ and $3h$, respectively. Elements were clustered near the stringer locations.

VI.3.3. Analysis of Buckling, Vibration, and Bending

In this section, the buckling modes, vibration modes, and bending response of the panel with substructure B are presented and compared to the buckling modes, vibration modes, and bending response of the panel without stringers. The computational methodology that was used in Section VI.2.3 was applied for the panel with substructure B.

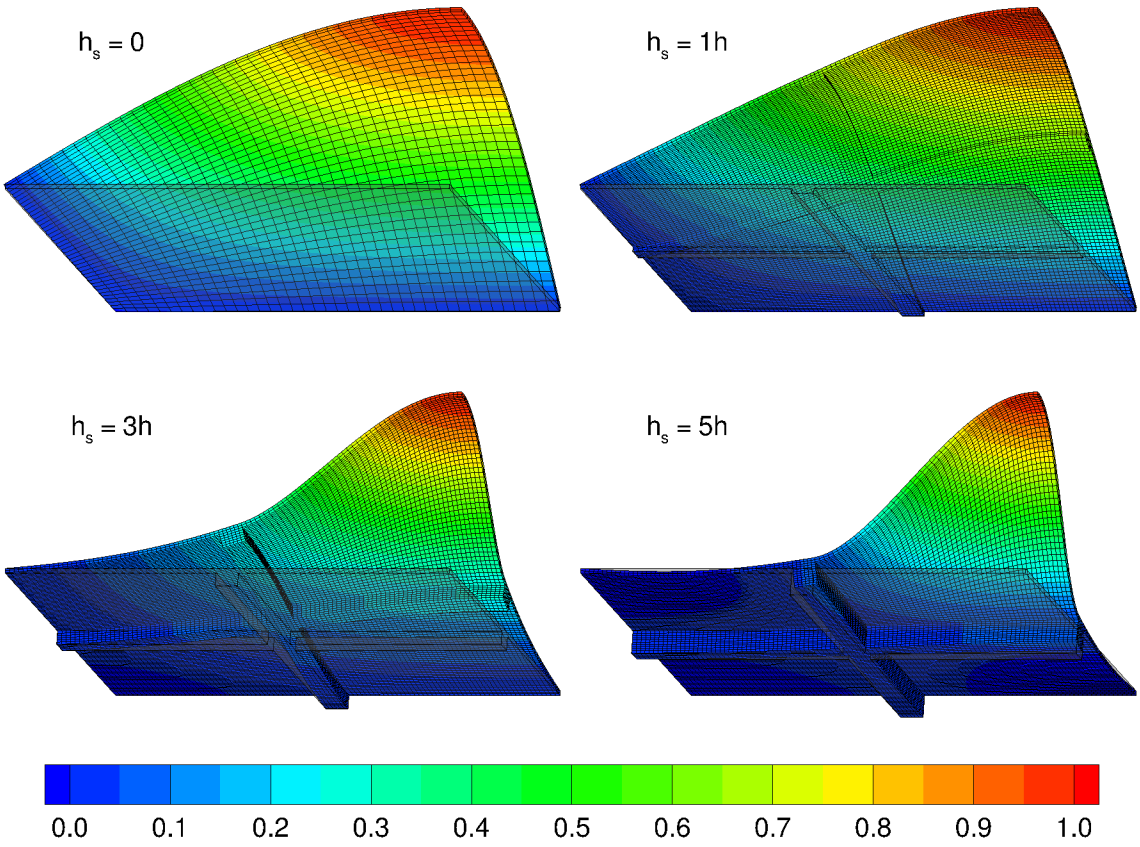


Figure VI.19: Buckling modes for one quarter of the panel with substructure B. Contour levels show out-of-plane displacement mode.

The primary modes of *in vacuo* buckling and vibration are shown for a quarter-panel with stringer thickness $h_s = 0$, 1, 3, and $5h$ in Figures VI.19 and VI.20, respectively. As the stringer thickness increased, the mode shape diverged from the panel without stringers. For higher stringer thicknesses, the deformation in the panel was concentrated in only the center section of the panel.

The panel response to a pressure load is shown in Figure VI.21. As the stringer thickness increased, the deformation of the panel decreased dramatically (note the difference in scale for each deformation). The stringers provided both bending stiff-

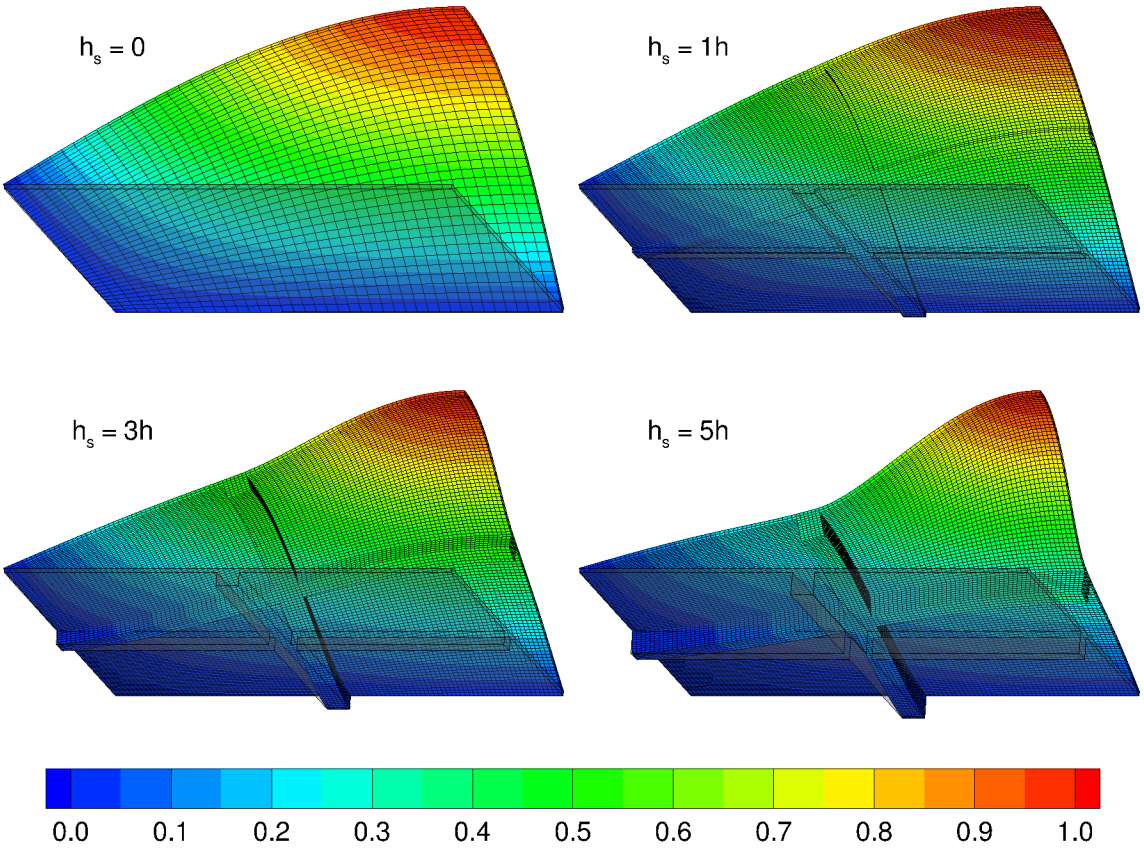


Figure VI.20: Vibrational modes for one quarter of the panel with substructure B. Contour levels show out-of-plane displacement mode.

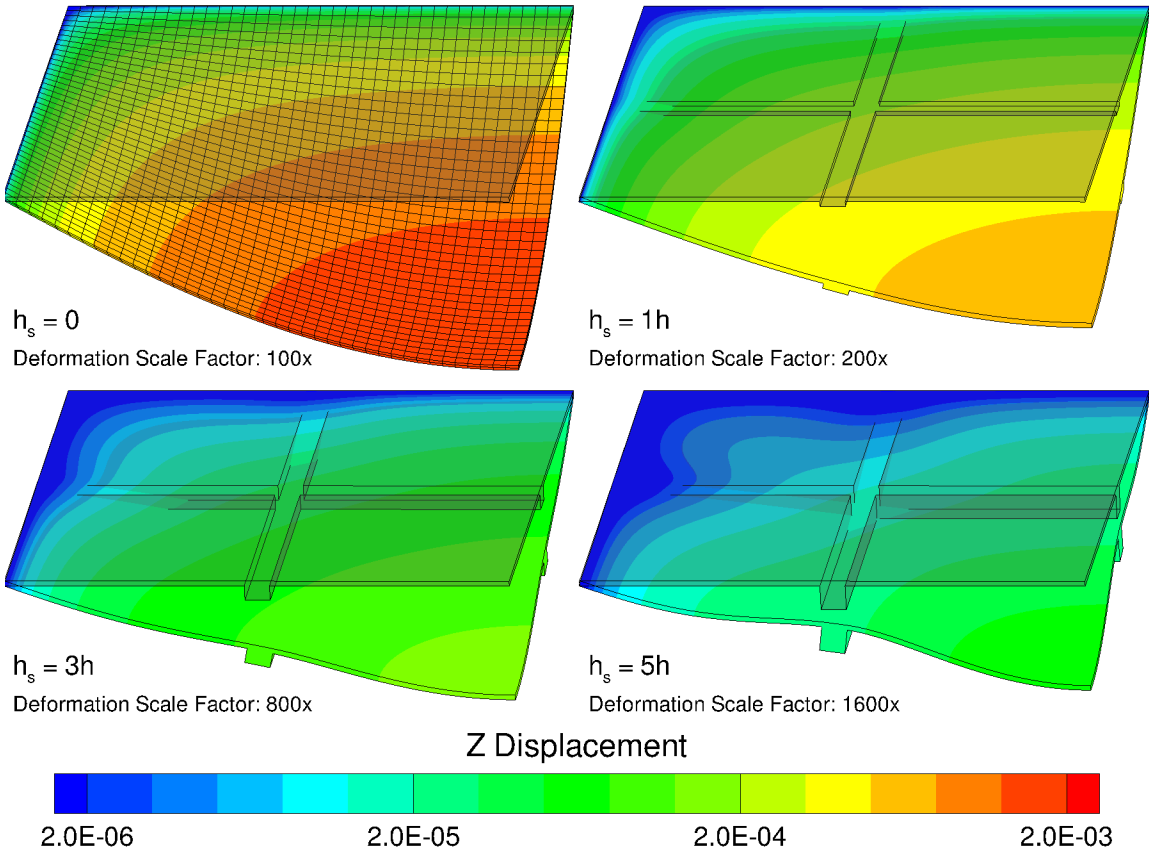


Figure VI.21: Displacement due to a pressure load of 1 kPa for one quarter of the panel with substructure B.

ness and torsional rigidity to the panel. The bending stiffness of the stringer was defined as:

$$K_s = \frac{\text{Pressure Load}}{\text{Out-of-plane Displacement}} = \frac{\int_0^{a/2}\int_0^{a/2} p dx dy}{\Delta z_s}, \tag{6.6}$$

where Δz_s is the out-of-plane displacement at the midpoint of the stringer.

The buckling temperature, natural vibration frequency, panel bending stiffness, stringer bending stiffness, and stringer torsional rigidity for each stringer thickness are given in Table VI.4. The values for each increased as the stringer thickness

increased. The buckling temperature and vibration frequency began to experience diminishing returns at $h_s = 3h$, while the panel stiffness was superlinear for each of the stringer thicknesses tested. Both he stringer bending stiffness and torsional rigidity maintained an approximately quadratic relationship with the stringer thickness (see Appendix F).

The panel with rigid stringers was modeled with a width of $a/2 - a_s$ and with nodes along the entirety of each side surface restrained ($\vec{u} = 0$). The results for this panel are given in the final row of Table VI.4, which represent an upper bound for a panel with substructure B.

The diminishing returns of increasing the stringer thickness were caused by the buckling temperature, vibrational frequency, and bending stiffness asymptotically approaching those of the panel with rigid stringers. The superlinear increase of these values at small stringer thicknesses was caused by the quadratic relationship between the stringer thickness and both the torsional rigidity and bending stiffness of the stringers.

VI.3.4. Rescaling Based on ABAQUS Analysis

The procedure followed in Section VI.2.4 was used for substructure B. The generalized mass $\mathbb{M}$ and the generalized force Ξ were found and used to generate the rescaled pressure parameter, $\tilde{\lambda}$.

For each stringer thickness, the frequency, generalized mass, and generalized aerodynamic force were found for the primary mode shape in vibration. These data, along with the ratio used for $\tilde{\lambda}$, are given in Table VI.5. Additionally, $\Delta\tilde{T}_{cr}$ is given

Table VI.4: ABAQUS analysis for panel with substructure B.

h_s/h	ΔT_{cr} [K]	f [s^{-1}]	K_p[a][m/MPa]	K_s[a][m/MPa]	$(JG)_s$ [kNm^2/rad]
0	9.112	23.94	0.561	0.757	2.035
1	34.12	37.95	1.502	2.219	5.355
3	112.9	79.46	6.628	11.60	28.59
5	135.6	120.7	14.47	35.55	87.39
10	144.3	172.2	28.86	185.8	422.0
RS[b]	151.5	194.0	34.66	∞	∞

[a] Effective Stiffness (pressure load/displacement) [b] Rigid Stringers

in Table VI.5 for each stringer thickness.

Table VI.5: Primary mode data for panel with substructure B.

h_s/h	f[a] [s^{-1}]	$\mathbb{M}$[b] [kg]	Ξ[c]$/q$ [m]	$\Xi/(q\mathbb{M}\omega^2)$ [1/MPa]	$\Delta\tilde{T}_{cr}$ [K]
0	23.94	2.466	0.1251	877.0	9.11
1	37.95	2.227	0.1159	356.7	28.3
3	79.46	1.800	0.0959	83.30	72.7

[a] Frequency of primary mode [b] Generalized mass
[c] Generalized aerodynamic force

There was a substantial difference between the buckling temperatures found in ABAQUS, ΔT_{cr}, and the buckling temperature found in the coupled solver, $\Delta\tilde{T}_{cr}$. While the exact cause is difficult to determine, there may be several factors in play. First, even with the additional restrained nodes along the ends of the stringers, the panel experienced a downward deflection as the temperature was increased. For a stringer thickness of $h_s = 1h$, this manifested as an exponentially increasing deforma-

tion, which made ΔT_{cr} difficult to pinpoint. For a stringer thickness of $h_s = 3h$, the panel first deforms downward, then changes to an upward buckling position. This behavior is documented in Figure VI.22.

A second possible source of error is the specification of restraints in ABAQUS. The buckling analysis was based on specified restraints along the restrained nodes, forcing the panel to experience compressive loading. The specified restraints mimic the thermal expansion of the panel, but there are small differences, such as the lack of thermal expansion in the out-of-plane direction. The effect of these differences should be minor compared to the downward deflection experienced before buckling.

VI.3.5. Rescaling Based on Substructure Mass

As in Section VI.2.5, the effective panel thickness for a panel without stringers with equivalent mass to the system with substructure B. For substructure B, the effective thickness h_{eff} is defined as:

$$h_{\text{eff}} = \left(\frac{h_s}{h} \frac{a_s}{a} \left(2 - \frac{a_s}{a} \right) + 1 \right) h \tag{6.7}$$

The effective pressure parameter $\bar{\lambda}$ and effective critical temperature, $\Delta \bar{T}_{cr}$ were defined as in Equation 6.5.

VI.3.6. Results of Panel Flutter

The panel with substructure B was analyzed using the coupled solver for 14 different pressures ranging from $p_\infty = 0.2$ bar to $p_\infty = 6.0$ bar for the substructure with thickness $h_s = 1h$ and for 25 different pressures ranging from $p_\infty = 0.1$ bar to $p_\infty = 16.0$ bar for the substructure with thickness $h_s = 3h$. The analysis techniques

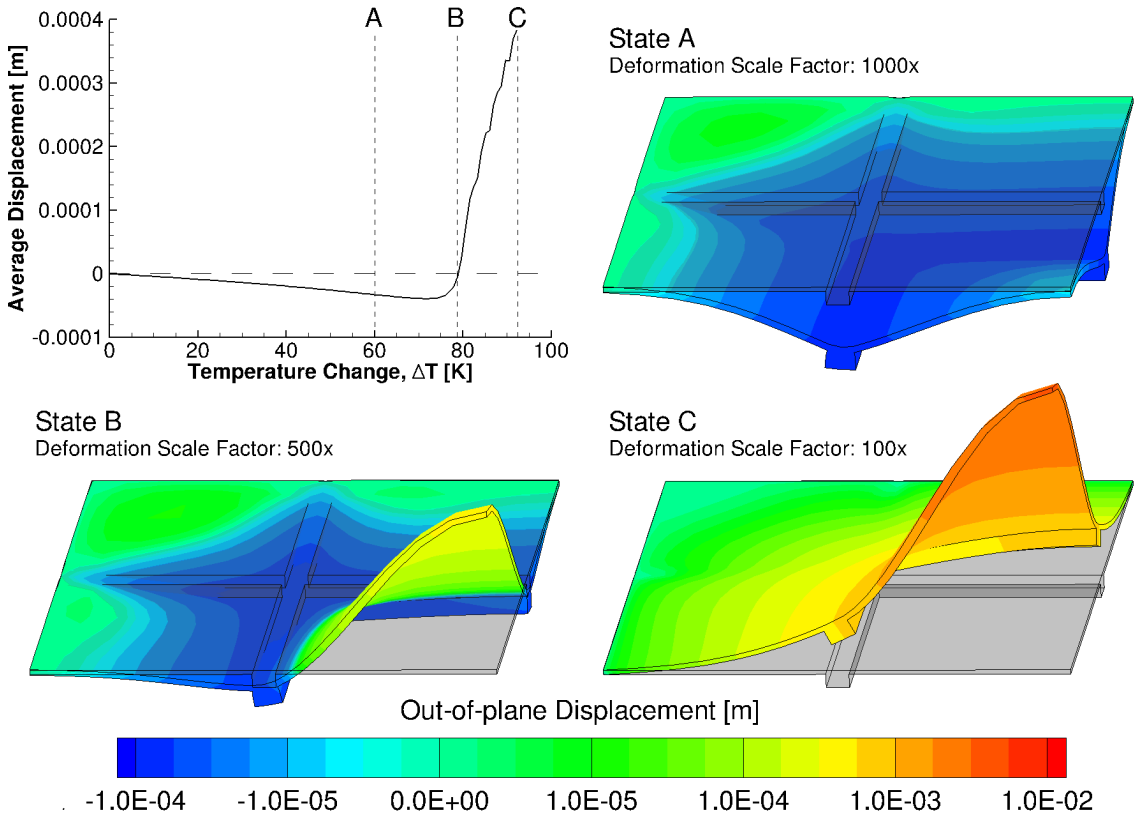


Figure VI.22: Displacement due to temperature increase for one quarter of the panel with substructure B. Three states are shown: A) initial buckling configuration, B) transition from initial to primary buckling, and C) primary buckling configuration.

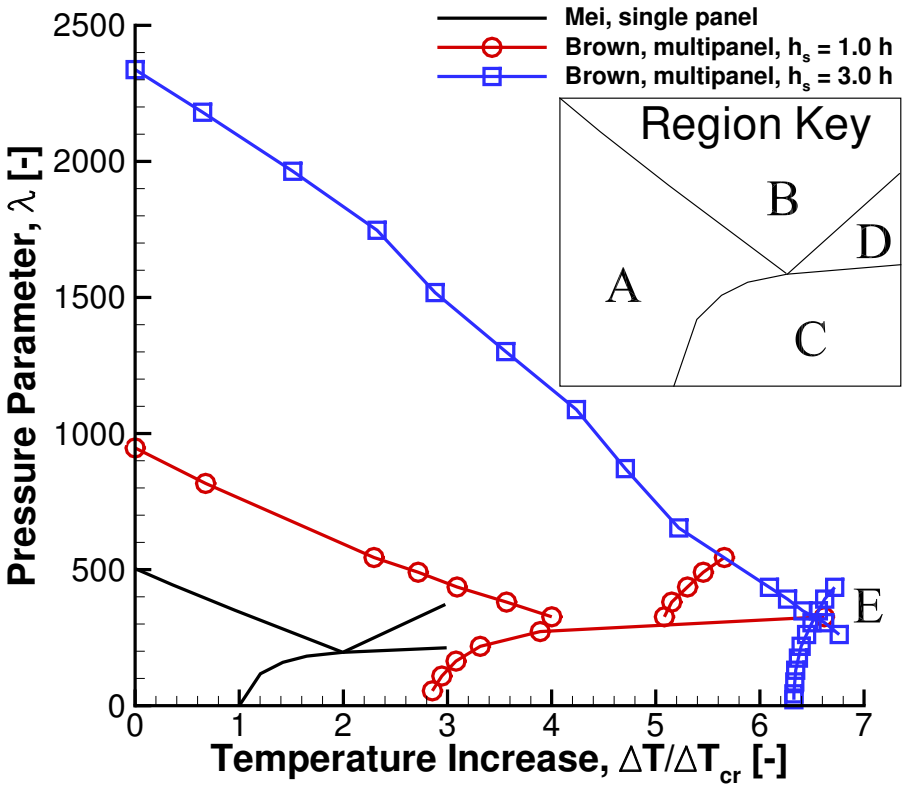


Figure VI.23: Regime boundary comparison between the panel with substructure B and a panel without stringers. A – Stable; B – LCO; C – Buckled; D – Snap-through; E – Buckled with LCO.

from Section VI.1.2 were again used to determine the regime boundaries, which are plotted for the panel with substructure B in Figure VI.23.

The addition of stringers significantly improved the resistance to both flutter and buckling. A nonlinear trend was again experienced, with the resistance to buckling experiencing mildly diminishing returns and the resistance to flutter experiencing increasing returns. This was expected based on the roughly quadratic increase in the stringer bending stiffness and torsional rigidity, as seen in Section VI.3.3.

A new behavior regime was found at high temperatures and relatively low dynamic pressures, marked as region E in Figure VI.23. In this regime, the panel and substructure oscillate about the buckled position, rather than about the undeformed position. This behavior began from two separate initial states. For λ values below 215, buckling occurred first, and then the panel entered LCO about the buckled position. The opposite was documented for λ values above 215 but lower than 300. For those panels, LCO occurred at a lower temperature, and the midpoint of the oscillation shifted as the panel additionally buckled. Above $\lambda = 300$, the temperature of the panel was not increased sufficiently to reach buckling.

The renormalization of the panel flutter regimes using $\tilde{\lambda}$ and $\Delta\tilde{T}_{cr}$ is shown in Figure VI.24. The rescaled boundaries were reduced far below the panel flutter regime boundaries for a panel without stringers. This suggests that the primary *in vacuo* vibration mode does not match the aeroelastic vibration mode.

The vibration mode for the aeroelastic case is shown in Figure VI.25. Compared to the *in vacuo* vibration mode in Figure VI.20, the location of maximum displacement has moved downstream. The aeroelastic mode had a much larger slope per displacement, which results in a much larger aerodynamic force. The imbalance of aerodynamic forces on the two modes is the primary source of disagreement between the panel with substructure B and the panel without stringers in Figure VI.24.

The regime boundaries are shown rescaled based on mass in Figure VI.16. The stable region shrunk, as for substructure A. In this case, however, the addition of stringers results in significantly improved performance, even with the mass taken into account. After taking the effects of mass into account, adding stringers of thickness

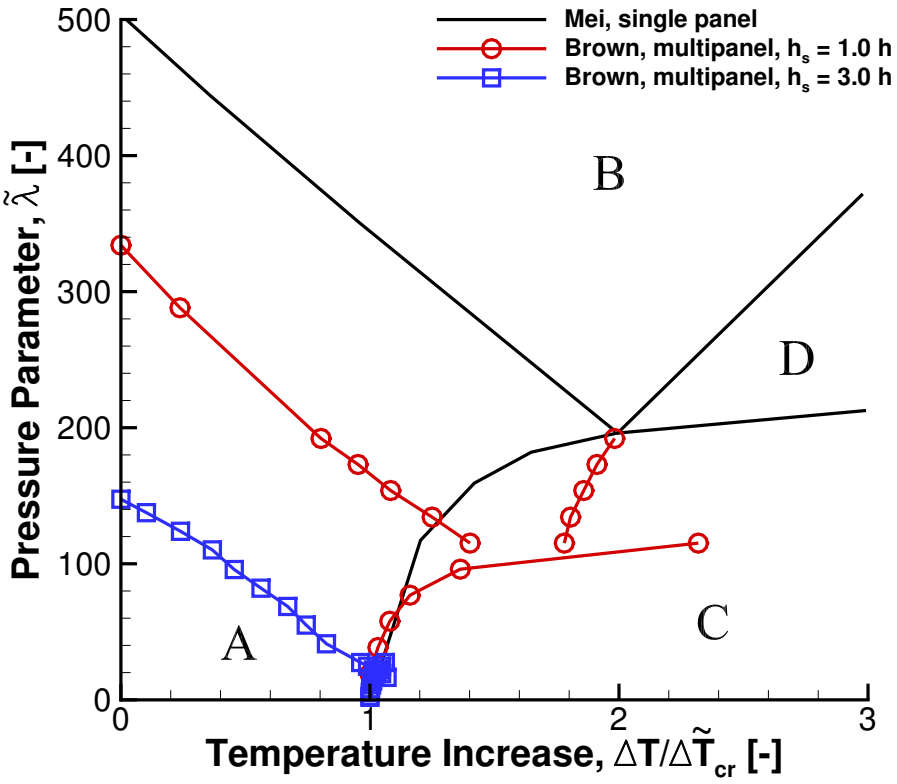


Figure VI.24: Regime boundary comparison between the panel with substructure B and a panel without stringers, using ABAQUS analysis as a motivation for rescaling. A – Stable; B – LCO; C – Buckled; D – Snap-through.

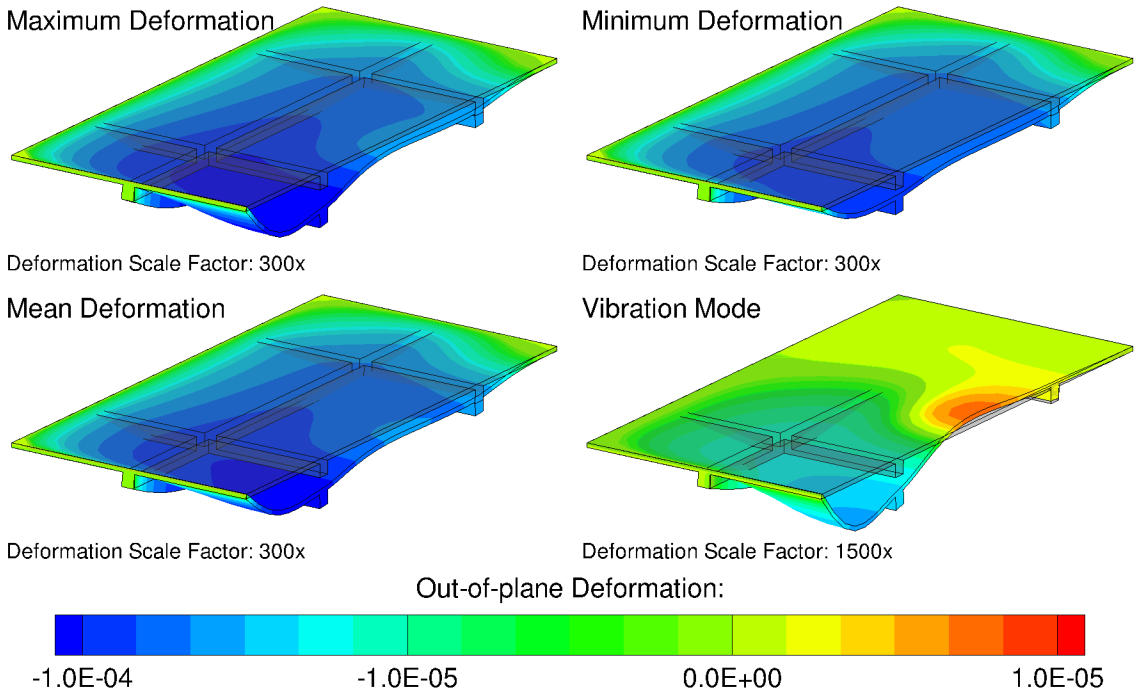


Figure VI.25: Isolation of aeroelastic vibrational mode for substructure B. Plate motion occurred at $\lambda = 585$, $\Delta T = 10$ K. Mean deformation was average of maximum and minimum, while vibrational mode was taken to be half of the difference of the maximum and minimum.

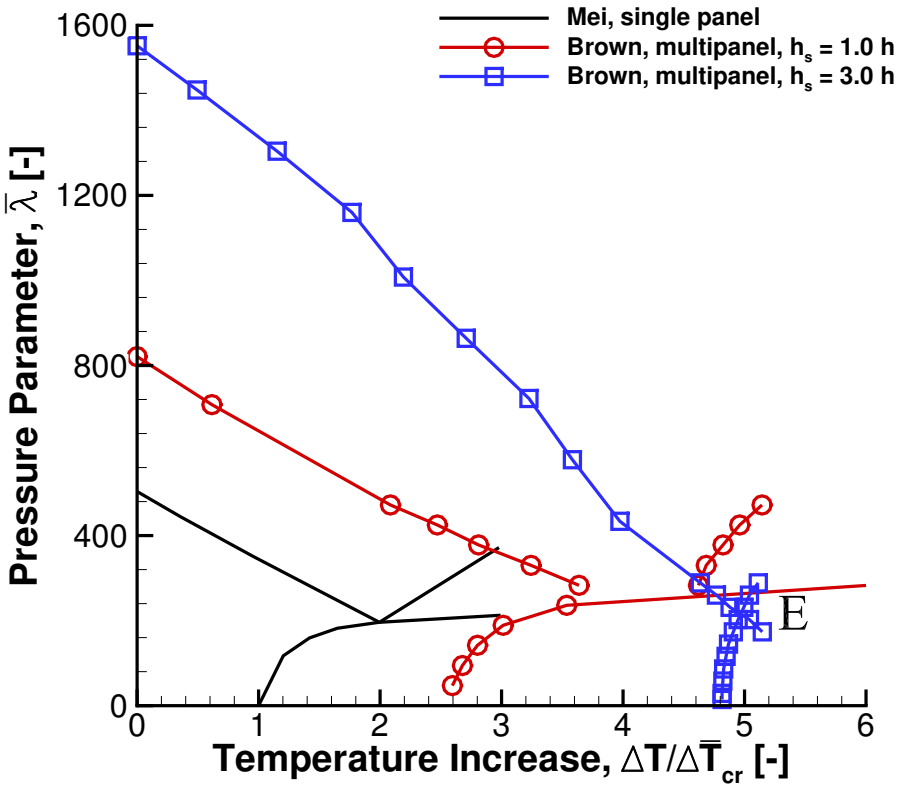


Figure VI.26: Regime boundary comparison between the panel with substructure B and a panel without stringers with thickness h_{eff}. A – Stable; B – LCO; C – Buckled; D – Snap-through; E – LCO about buckled deformation.

$h_s = 3h$ caused an increase of 208% in the non-dimensional pressure parameter for LCO with $\Delta T = 0$. An increase of 380% was seen for the *in vacuo* buckling temperature for the same case.

CHAPTER VII

CONCLUSIONS

The main contribution of this work resides in the numerical investigation of aerothermal and aerothermoelastic phenomena that are more complex than those previously studied in the context of hypersonic flow. Additionally, the numerical investigation was done by coupling state-of-the-art aerodynamic, structural elasticity, and thermal-diffusion codes. Aerothermal simulations were performed on a double-wedge airfoil with a rounded leading edge, which was simulated with several interior cooling profiles. Aerothermoelastic simulations were performed on a panel supported by stringers and subject to hypersonic flow. The following conclusions can be drawn from the computational studies presented herein:

1. The coupled aerothermoelastic solver provides accurate results for simulations involving fluid-structure interaction in hypersonic flows. Verification and validation studies showed that each of the methodologies used produces results in line with those previously published.

2. Variation in angle of attack caused a significant percentage increase in heat flux over most of the hypersonic wedge airfoil. A change in angle of attack from 0 to 5 degrees resulted in an increase of the local heat flux by more than 60%. This increase may limit the operational range of aircraft in hypersonic flight.

3. Flight envelopes can be generated for given wall temperatures and average heat flux. Although these average analyses do not provide information about

localized heating, they give an estimation of the cooling necessary to allow sustained flight at a given exterior wall temperature.

4. Augmented cooling near the leading edge of a hypersonic airfoil can adequately cool the entire airfoil, but creates large chordwise temperature gradients. Cooling with chordwise constant heat flux is inefficient because of localized high temperature at the leading edge. The reduction of both the maximum temperature and the temperature gradient requires more sophisticated cooling distributions than those presented herein.

5. Post-flutter dynamics of even a simple panel should be simulated without the assumption of symmetry. At high temperatures, an asymmetric mode exists which grows to dominate the flutter of the panel.

6. The addition of stringers modifies both the *in vacuo* and aeroelastic mode shapes and eigenvalues for both vibration and buckling. The change in the *in vacuo* mode provides a reasonable estimate of how the panel flutter regime boundaries of the panel will change, but the aeroelastic mode is needed for accurate rescaling.

7. Stringers are most effective when providing both bending stiffness and torsional rigidity to the panel. In other words, ideal stringer placement will prevent both out-of-plane displacement and local rotation of the panel.

8. Aeroelastic modes tend to have larger deformations downstream. In order to provide greater resistance to LCO, it may be beneficial to place either stronger or more stringers downstream to restrict the aeroelastic mode.

CHAPTER VIII

FUTURE WORK

The goal of this work was to advance the state of the art towards the simulation of aerothermoelastic phenomena for complete hypersonic aircraft. Future work would seek to expand upon the advances made in this work:

1. Aerothermal studies of airfoils and wedges should be performed using cooling conditions based on convective heat transfer, which would more accurately model internal coolant flow and allow for the design of cooling systems, including concepts such as coolant consumption per second of flight.

2. The addition of radiative heat transfer to the aerothermal simulation would increase the accuracy of the aerothermal simulations, especially as the Mach number and wall temperature are increased.

3. More detailed analysis of the post-flutter response of the panel should be performed, detailing the decomposition of the response into aerothermoelastic vibration modes. Investigation of multi-mode vibration and transition to chaos would provide a more precise view of the boundary between LCO and snap-through.

4. An expansion of the model to a panel containing a larger number of stringers and further expansion to a full wing could provide insight for design studies of hypersonic vehicles.

5. The coupling of long-term aerothermal analysis using active cooling with aerothermoelastic flutter simulation would facilitate the investigation of active cooling as a means of panel flutter suppression.

6. Navier–Stokes aerodynamics or the Eckert reference enthalpy method would enable more realistic heating patterns to be used in the simulation of panel flutter, which is especially important for situations in which the heating rate is high relative to the material thermal conductivity.

REFERENCES

[1] Hank, J. M., Murphy, J. S., and Mutzman, R. C., "The X-51A scramjet engine flight demonstration program," *15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2008-2540, AIAA, Dayton, OH, May 2008,
doi:10.2514/6.1991-1596.

[2] Jackson, K. R., Gruber, M. R., and Buccellato, S., "Mach 6–8+ hydrocarbon-fueled scramjet flight experiment: the HIFiRE flight 2 project," *Journal of Propulsion and Power*, Vol. 31, No. 1, 2014, pp. 36–53,
doi:10.2514/1.B35350.

[3] Quiroz, R., Embler, J., Jacobs, R., Tzong, G., and Liguore, S., "Air Vehicle Integration and Technology Research (AVIATR) Task Order 0023: predictive capability for hypersonic structural response and life prediction: phase II - detailed design of hypersonic cruise vehicle hot-structure," Tech. Rep. AFRL-RQ-WP-TR-2012-0265, AFRL, Dayton, OH, 2012.

[4] Kendall, R., Rindal, R., and Bartlett, E., "Thermochemical ablation," *AIAA Thermophysics Specialist Conference*, 65-642, AIAA, Monterey, CA, September 1965,
doi:10.2514/6.1965-642.

[5] Liu, S. and Zhang, B., "Effects of active cooling on the metal thermal protection systems," *Aerospace Science and Technology*, Vol. 15, No. 7, 2011, pp. 526–533, doi:10.1016/j.ast.2010.08.001.

[6] Stallings, J., R. L., "Experimentally determined local flow properties and drag coefficients for a family of blunt bodies at Mach numbers from 2.49 to 4.63," Tech. Rep. NASA-TR-R-274, NASA Langley Research Center, Hampton, VA, 1967.

[7] Korobkin, I., "Laminar heat transfer characteristics of a hemisphere for the Mach number range 1.9 to 4.9." Tech. Rep. 3841.10, US Naval Ordnance Laboratory, Silver Spring, MD, 1954.

[8] Fay, J. and Riddell, F., "Theory of stagnation point heat transfer in dissociated air," *Journal of the Aeronautical Sciences*, Vol. 25, No. 2, 1958, pp. 73–85,
doi:10.2514/8.7517.

[9] Swigart, R. J., "A theory of asymmetric hypersonic blunt-body flows," *AIAA Journal*, Vol. 1, No. 5, 1963, pp. 1034–1042,
doi:10.2514/3.1721.

[10] Dyke, M. V., "Second-order compressible boundary layer theory with application to blunt bodies in hypersonic flow," in "Hypersonic Flow Research," American Institute of Aeronautics and Astronautics (AIAA), Reston, VA, pp. 37–76, 1962,
doi:10.2514/5.9781600864810.0037.0076.

[11] Yakura, J. K., "Theory of entropy layers and nose bluntness in hypersonic flow," in "Hypersonic Flow Research," American Institute of Aeronautics and Astronautics (AIAA), Reston, VA, pp. 421–470, 1962,
doi:10.2514/5.9781600864810.0421.0470.

[12] Wittliff, C. E., "Heat transfer to slender cones in hypersonic air flow, including effects of yaw and nose bluntness," *Journal of the Aerospace Sciences*, Vol. 29, No. 7, 1962, pp. 761–774,
doi:10.2514/8.9603.

[13] Frazer, R., "Aeroplane Wing Flutter," *Aircraft Engineering and Aerospace Technology*, Vol. 1, No. 3, 1929, pp. 94–97,
doi:10.1108/eb029120.

[14] Glauert, H., "The force and moment on an oscillating aerofoil," in "Vortrage aus dem Gebiete der Aerodynamik und verwandter Gebiete," Springer, Heidelberg, Germany, pp. 88–95, 1930,
doi:10.1007/978-3-662-33791-2_16.

[15] Theodorsen, T., "Report No. 496, general theory of aerodynamic instability and the mechanism of flutter," *Journal of the Franklin Institute*, Vol. 219, No. 6, 1935, pp. 766–767,
doi:10.1016/s0016-0032(35)92022-1.

[16] von Karman, T., "Airfoil theory for non-uniform motion," *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, Vol. 5, No. 10, 1938, pp. 379–390,
doi:10.2514/8.674.

[17] Goland, M., "The flutter of a uniform cantilever wing," *Journal of applied mechanics – transactions of the ASME*, Vol. 12, No. 4, 1945, pp. A197–A208.

[18] Hedman, S. G., "Vortex lattice method for calculation of quasi steady state loadings on thin elastic wings in subsonic flow," Tech. Rep. FFA 105, Aeronautical Research Institute of Sweden, Stockholm, 1966.

[19] Rubbert, P. E., "Theoretical characteristics of arbitrary wings by a non-planar vortex lattice method," Tech. Rep. D6-9244, The Boeing Company, Seattle, WA, 1964.

[20] Albano, E. and Rodden, W. P., "A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows." *AIAA Journal*, Vol. 7, No. 2, 1969, pp. 279–285,
doi:10.2514/3.5086.

[21] Chushkin, P., "Numerical method of characteristics for three-dimensional supersonic flows," *Progress in Aerospace Sciences*, Vol. 9, 1968, pp. 41–122,
doi:10.1016/0376-0421(68)90004-3.

[22] Deiwert, G. S., "Numerical Simulation of High Reynolds Number Transonic Flows," *AIAA Journal*, Vol. 13, No. 10, 1975, pp. 1354–1359,
doi:10.2514/3.60547.

[23] Thomas, P. D. and Lombard, C. K., "Geometric conservation law and its application to flow computations on moving grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037,
doi:10.2514/3.61273.

[24] Mavriplis, D. J. and Nastase, C. R., "On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes," *Journal of Computational Physics*, Vol. 230, No. 11, 2011, pp. 4285–4300,
doi:10.1016/j.jcp.2011.01.022.

[25] Guillard, H. and Farhat, C., "On the significance of the geometric conservation law for flow computations on moving meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 11-12, 2000, pp. 1467–1482,
doi:10.1016/s0045-7825(00)00173-0.

[26] Mavriplis, D. and Nastase, C., "On the geometric conservation law for high-order discontinuous galerkin discretizations on dynamically deforming meshes," in "46th AIAA Aerospace Sciences Meeting and Exhibit," American Institute of Aeronautics and Astronautics, Reno, NV, 2008,
doi:10.2514/6.2008-778.

[27] Soheili, A. R. and Ameri, M. A., "Adaptive grid based on geometric conservation law level set method for time dependent PDE," *Numerical Methods for Partial Differential Equations*, Vol. 25, No. 3, 2009, pp. 582–597,
doi:10.1002/num.20360.

[28] Lesoinne, M. and Farhat, C., "Geometric conservation laws for aeroelastic computations using unstructured dynamic meshes," in "12th Computational Fluid Dynamics Conference," American Institute of Aeronautics and Astronautics, San Diego, CA, 1995,
doi:10.2514/6.1995-1709.

[29] Mei, C., Abdel-Motagaly, K., and Chen, R., "Review of nonlinear panel flutter at supersonic and hypersonic speeds," *Applied Mechanics Reviews*, Vol. 52, No. 10, 1999, p. 321,
doi:10.1115/1.3098919.

[30] Ashley, H., "Piston theory-a new aerodynamic tool for the aeroelastician," *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, Vol. 23, No. 12, 1956, pp. 1109–1118,
doi:10.2514/8.3740.

[31] Xue, D. Y. and Mei, C., "Finite element nonlinear panel flutter with arbitrary temperatures in supersonic flow," *AIAA Journal*, Vol. 31, No. 1, 1993, pp. 154–162,
doi:10.2514/3.11332.

[32] Dowell, E. H., "Nonlinear flutter of curved plates. II," *AIAA Journal*, Vol. 8, No. 2, 1970, pp. 259–261,
doi:10.2514/3.5653.

[33] Dugundji, J., "Theoretical considerations of panel flutter at high supersonic Mach numbers." *AIAA Journal*, Vol. 4, No. 7, 1966, pp. 1257–1266,
doi:10.2514/3.3657.

[34] Gordnier, R. and Visbal, M., "Development of a three-dimensional viscous aeroelastic solver for nonlinear panel flutter," *Journal of Fluids and Structures*, Vol. 16, No. 4, 2002, pp. 497–527,
doi:10.1006/jfls.2000.0434.

[35] Selvam, R., Visbal, M., and Morton, S., "Computation of nonlinear viscous panel flutter using a fully-implicit aeroelastic solver," in "39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit," American Institute of Aeronautics and Astronautics (AIAA), 1998,
doi:10.2514/6.1998-1844.

[36] Crowell, A., McNamara, J., and Miller, B., "Surrogate based reduced-order aerothermodynamic modeling for structural response prediction at high mach numbers," in "52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference," American Institute of Aeronautics and Astronautics (AIAA), Denver, CO, 2011,
doi:10.2514/6.2011-2014.

[37] Miller, B. A. and McNamara, J. J., "Time-marching considerations for response prediction of structures in hypersonic flows," *AIAA Journal*, Vol. 53, No. 10, 2015, pp. 3028–3038,
doi:10.2514/1.j053872.

[38] Dowell, E. H., "Generalized aerodynamic forces on a flexible plate undergoing transient motion in a shear flow with an application to panel flutter," *AIAA Journal*, Vol. 9, No. 5, 1971, pp. 834–841, doi:10.2514/3.6283.

[39] Dowell, E., Dugundji, J., and Perkin, B., "Subsonic flutter of panels on continuous elastic foundations," *AIAA Journal*, Vol. 1, No. 5, 1963, pp. 1146–1154, doi:10.2514/3.1738.

[40] Yang, T. Y. and Han, A. D., "Nonlinear panel flutter using high-order triangular finite elements," *AIAA Journal*, Vol. 21, No. 10, 1983, pp. 1453–1461, doi:10.2514/3.8267.

[41] Miles, J. W., "Supersonic flutter of a cylindrical shell - II," *Journal of the Aerospace Sciences*, Vol. 25, No. 5, 1958, pp. 312–316, doi:10.2514/8.7648.

[42] Holt, M. and Strack, S. L., "Supersonic panel flutter of a cylindrical shell of finite length," *Journal of the Aerospace Sciences*, Vol. 28, No. 3, 1961, pp. 197–208, doi:10.2514/8.8925.

[43] Dowell, E. H., "Panel flutter - a review of the aeroelastic stability of plates and shells," *AIAA Journal*, Vol. 8, No. 3, 1970, pp. 385–399, doi:10.2514/3.5680.

[44] Nydick, I., Friedmann, P., and Zhong, X., "Hypersonic panel flutter studies on curved panels," in "36th Structures, Structural Dynamics and Materials Conference," American Institute of Aeronautics and Astronautics, New Orleans, LA, 1995, doi:10.2514/6.1995-1485.

[45] Thornton, E. A., "Thermal buckling of plates and shells," *Applied Mechanics Reviews*, Vol. 46, No. 10, 1993, p. 485, doi:10.1115/1.3120310.

[46] Udrescu, R., "On the understanding of snap-through phenomenon in aerothermoelasticity of panels," in "41st Structures, Structural Dynamics, and Materials Conference and Exhibit," American Institute of Aeronautics and Astronautics (AIAA), Atlanta, GA, 2000, doi:10.2514/6.2000-1531.

[47] Crisfield, M., "A fast incremental/iterative solution procedure that handles snap-through," *Computers & Structures*, Vol. 13, No. 1-3, 1981, pp. 55–62, doi:10.1016/0045-7949(81)90108-5.

[48] Bein, T., Friedmann, P., Zhong, X., and Nydick, I., "Hypersonic flutter of a curved shallow panel with aerodynamic heating," in "34th Structures, Structural Dynamics and Materials Conference," American Institute of Aeronautics and Astronautics, La Jolla, CA, 1993,
doi:10.2514/6.1993-1318.

[49] Song, Z.-G. and Li, F.-M., "Aerothermoelastic analysis of nonlinear composite laminated panel with aerodynamic heating in hypersonic flow," *Composites Part B: Engineering*, Vol. 56, 2014, pp. 830–839,
doi:10.1016/j.compositesb.2013.09.019.

[50] Miller, B., *Loosely coupled time integration of fluid-thermal-structural interactions in hypersonic flows*, Ph.D. thesis, Ohio State University, 2015.

[51] Dechaumphai, P., Wieting, A. R., and Thornton, E. A., "Flow-thermal-structural study of aerodynamically heated leading edges," *Journal of Spacecraft and Rockets*, Vol. 26, No. 4, 1989, pp. 201–209,
doi:10.2514/3.26055.

[52] Thuruthimattam, B., Friedmann, P., Powell, K., and McNamara, J., "Aeroelasticity of a generic hypersonic vehicle," in "43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference," American Institute of Aeronautics and Astronautics (AIAA), Denver, CO, 2002,
doi:10.2514/6.2002-1209.

[53] Lamorte, N. and Friedmann, P. P., "Hypersonic aeroelastic and aerothermoelastic studies using computational fluid dynamics," *AIAA Journal*, Vol. 52, No. 9, 2014, pp. 2062–2078,
doi:10.2514/1.j053018.

[54] Culler, A. J. and McNamara, J. J., "Studies on fluid-thermal-structural coupling for aerothermoelasticity in hypersonic flow," *AIAA Journal*, Vol. 48, No. 8, 2010, pp. 1721–1738,
doi:10.2514/1.j050193.

[55] Eckert, E., "Engineering relations for heat transfer and friction in high-velocity laminar and turbulent boundary-layer flow over surfaces with constant pressure and temperature," *Transactions of the ASME*, Vol. 78, No. 6, 1956, pp. 1273–1283.

[56] McNamara, J. J. and Friedmann, P. P., "Aeroelastic and aerothermoelastic analysis in hypersonic flow: past, present, and future," *AIAA Journal*, Vol. 49, No. 6, 2011, pp. 1089–1122,
doi:10.2514/1.j050882.

[57] Felippa, C. A., Park, K., and Farhat, C., "Partitioned analysis of coupled mechanical systems," *Computer methods in applied mechanics and engineering*, Vol. 190, No. 24, 2001, pp. 3247–3270, doi:10.1016/S0045-7825(00)00391-1.

[58] Lai, W. M., Rubin, D., and Krempl, E., *Introduction to continuum mechanics*, Butterworth-Heinemann, Waltham, MA, 3rd ed., 2009.

[59] Menter, F. R., "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605, doi:10.2514/3.12149.

[60] Han, Z. and Cizmas, P. G. A., "Axial thrust load prediction of centrifugal compressors," *International Journal of Turbo & Jet-Engines*, Vol. 20, No. 1, 2003, pp. 1–16, doi:10.1515/TJJ.2003.20.1.1.

[61] Cizmas, P. G. A., Gargoloff, J. I., Strganac, T. W., and Beran, P. S., "Parallel multigrid algorithm for aeroelasticity simulations," *Journal of Aircraft*, Vol. 47, No. 1, 2010, pp. 53–63, doi:10.2514/1.40201.

[62] Barth, T. J., "A 3-D upwind Euler solver for unstructured meshes," *10th AIAA Computational Fluid Dynamics Conference*, 91-1548-CP, AIAA, Honolulu, HI, June 1991, doi:10.2514/6.1991-1548.

[63] Kim, S., Kim, C., Rho, O., and Hong, S. K., "Cures for the shock instability: development of a shock-stable Roe scheme," *Journal of Computational Physics*, Vol. 185, No. 2, 2003, pp. 342–374, doi:10.1016/S0021-9991(02)00037-2.

[64] Paige, C. C. and Saunders, M. A., "LSQR: an algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, Vol. 8, No. 1, 1982, pp. 43–71, doi:10.1145/355984.355989.

[65] Weiss, J. M., Maruszewski, J. P., and Smith, W. A., "Implicit solution of preconditioned Navier–Stokes equations using algebraic multigrid," *AIAA Journal*, Vol. 37, No. 1, 1999, pp. 29–36, doi:10.2514/2.689.

[66] Blazek, J., *Computational fluid dynamics: principles and applications*, Elsevier Science, Kidlington, Oxford, 1st ed., 2001.

[67] Courant, R., Friedrichs, K., and Lewy, H., "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, Vol. 100, No. 1, 1928, pp. 32–74,
doi:10.1007/BF01448839.

[68] Weiss, J. M. and Smith, W. A., "Preconditioning applied to variable and constant density flows," *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2050–2057,
doi:10.2514/3.12946.

[69] Ihm, S.-W. and Kim, C., "Computations of homogeneous-equilibrium two-phase flows with accurate and efficient shock-stable schemes," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3012–3037,
doi:10.2514/1.35097.

[70] Jameson, A. and Mavriplis, D., "Finite volume solution of the two-dimensional euler equations on a regular triangular mesh," *AIAA Journal*, Vol. 24, No. 4, 1986, pp. 611–618,
doi:10.2514/3.9315.

[71] Forum, M., "MPI: A Message-Passing Interface Standard Version 3.0." Tech. rep., University of Tennessee, Knoxville, TN, 2012.

[72] Reddy, J. N., *An introduction to the finite element method*, McGraw-Hill, New York, NY, pp. 318–321, 3rd ed., 2006.

[73] Giles, M., "Stability analysis of numerical interface conditions in fluid-structure thermal analysis," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 4, 1997, pp. 421–436,
doi:10.1002/(SICI)1097-0363(19970830)25:4⟨421::AID-FLD557⟩3.0.CO;2-J.

[74] Buhmann, M. D., "Radial basis functions," *Acta Numerica 2000*, Vol. 9, No. 0, 2000, pp. 1–38,
doi:10.1017/s0962492900000015.

[75] Beckert, A. and Wendland, H., "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerospace Science and Technology*, Vol. 5, No. 2, 2001, pp. 125–134,
doi:10.1016/s1270-9638(00)01087-7.

[76] Ding, L., Lu, Z., and Guo, T., "An efficient dynamic mesh generation method for complex multi-block structured grid," *Advances in Applied Mathematics and Mathematics*, Vol. 6, No. 1, 2014, pp. 120–134,
doi:10.4208/aamm.2013.m199.

[77] Illingworth, C., "Some solutions of the equations of flow of a viscous compressible fluid," *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 46, No. 3, 1950, pp. 469–478.

[78] Falkiewicz, N. J. and Cesnik, C. E. S., "Enhanced modal solutions for structural dynamics in aerothermoelastic analysis," *Proceedings 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2011-1963, AIAA, Denver, Colorado, April 2011, doi:10.2514/6.2011-1963.

[79] McNamara, J. J., *Aeroelastic and aerothermoelastic behavior of two and three dimensional lifting surfaces in hypersonic flow*, Ph.D. thesis, University of Michigan, 2005.

[80] Glass, C. E. and Hunt, L. R., "Aerothermal tests of spherical dome protuberances on a flat plate at a Mach number of 6.5," *NASA STI/Recon Technical Report N*, Vol. 87, 1986, p. 13664.

[81] Ostoich, C. M., Bodony, D. J., and Geubelle, P. H., "Fluid-thermal response of spherical dome under a Mach 6.59 laminar boundary layer," *AIAA Journal*, Vol. 50, No. 12, 2012, pp. 2791–2808, doi:10.2514/1.j051634.

[82] Vinokur, M., "On one-dimensional stretching functions for finite-difference calculations," *Journal of Computational Physics*, Vol. 50, No. 2, 1983, pp. 215–234, doi:10.1016/0021-9991(83)90065-7.

[83] Blevins, R. D., Bofilios, D., Holehouse, I., Hwa, V. W., Tratt, M. D., Laganelli, A. L., Pozefsky, P., and Pierucci, M., "Thermo-vibro-acoustic loads and fatigue of hypersonic flight vehicle structure," Tech. Rep. AFRL-RB-WP-TR-2009-3139, AFRL, Dayton, OH, 2009.

[84] Ko, W. L., "Thermal buckling analysis of rectangular panels subjected to humped temperature profile heating," Tech. Rep. NASA/TP-2004-212041, NASA Dryden Flight Research Center, Edwards, CA, 2004.

[85] Mei, C., Xue, D. Y., and Zhou, R. C., "Finite element time domain - modal formulation for nonlinear flutter of composite panels," *AIAA Journal*, Vol. 32, No. 10, 1994, pp. 2044–2052, doi:10.2514/3.12250.

[86] Cheng, G. and Mei, C., "Finite element modal formulation for hypersonic panel flutter analysis with thermal effects," *AIAA Journal*, Vol. 42, No. 4, 2004, pp. 687–695, doi:10.2514/1.9553.

[87] Anderson, J. D., *Modern compressible flow with historical perspective*, McGraw-Hill, New York, New York, 3rd ed., 2003.

APPENDIX A

REYNOLDS TRANSPORT THEOREM FOR A MOVING MESH

If we take a material volume $\Omega_m(t)$, and assume that it instantaneously coincides with a stationary control volume Ω_c, the Reynolds Transport Theorem states that for some conserved quantity $\aleph$:

$$\frac{D}{Dt}\int_{\Omega_m(t)} \aleph \, dV = \int_{\Omega_c} \frac{\partial}{\partial t}(\aleph) \, dV + \oint_{\partial\Omega_c} \aleph(\vec{u}\cdot\hat{n}) \, dS = \int_{\Omega_c} \vec{f_b} \, dV + \oint_{\partial\Omega_c} \vec{f_s} \, dS, \quad \text{(A.1)}$$

where $\vec{u}$ is the fluid velocity and $\vec{f_b}$ and $\vec{f_s}$ are the body and surface sources, respectively.

Similarly, if we define an arbitrarily moving volume $\Omega(t)$, also assuming that it coincides with our control volume at time t, we obtain:

$$\frac{d}{dt}\int_{\Omega(t)} \aleph \, dV = \int_{\Omega_c} \frac{\partial}{\partial t}(\aleph) \, dV + \oint_{\partial\Omega_c} \aleph(\vec{u}_g\cdot\hat{n}) \, dS, \quad \text{(A.2)}$$

where $\vec{u}_g$ is the velocity of the moving boundary.

Next, we solve for $\int_{\Omega_c} \frac{\partial}{\partial t}(\aleph) dV$ and substitute this into (A.1), obtaining:

$$\frac{d}{dt}\int_{\Omega(t)} \aleph \, dV - \oint_{\partial\Omega_c} \aleph(\vec{u}_g\cdot\hat{n}) \, dS + \oint_{\partial\Omega_c} \aleph(\vec{u}\cdot\hat{n}) \, dS = \int_{\Omega_c} \vec{f_b} \, dV + \oint_{\partial\Omega_c} \vec{f_s} \, dS \quad \text{(A.3)}$$

Finally, we recognize that for our moving mesh, the quantity of interest is the change in time of the conserved quantity, averaged over a cell in the mesh. Therefore, we solve (A.3) for $\frac{d}{dt}\int_{\Omega(t)} \aleph \, dV$, letting $\vec{u}_r$ be the relative velocity between the fluid and the grid, $\vec{u}_r = \vec{u} - \vec{u}_g$:

$$\frac{d}{dt}\int_{\Omega(t)} \aleph \, dV = -\oint_{\partial\Omega_c} \aleph(\vec{u}_r\cdot\hat{n}) \, dS + \int_{\Omega_c} \vec{f_b} \, dV + \oint_{\partial\Omega_c} \vec{f_s} \, dS \quad \text{(A.4)}$$

APPENDIX B

PISTON THEORY DERIVATION

Piston theory is based on the Method of Characteristics. Referring to [87, pp.285-299], we define characteristic lines for one-dimensional compressible flow as:

$$C_+ : \frac{dx}{dt} = u + a \qquad\qquad C_- : \frac{dx}{dt} = u - a \tag{B.1}$$

Along these lines, it can be shown that:

$$\text{along } C_\pm : du \pm \frac{dp}{\rho a} = 0 \tag{B.2}$$

Integration along $C_\pm$ yields:

$$J_\pm = \int \left(du \pm \frac{dp}{\rho a} \right) = u \pm \int \frac{dp}{\rho a} \tag{B.3}$$

Using the definition of the speed of sound $a = \sqrt{\frac{\gamma p}{\rho}}$, along with the assumption of isentropic flow, we define dp and ρ as functions of a as follows:

$$p = c_1 T^{\frac{\gamma}{\gamma-1}} = c_2 a^{\frac{2\gamma}{\gamma-1}}$$

$$dp = c_2 \frac{2\gamma}{\gamma-1} a^{\left(\frac{2\gamma}{\gamma-1}-1\right)} da$$

$$\rho = \frac{\gamma p}{a^2} = c_2 \gamma a^{\left(\frac{2\gamma}{\gamma-1}-2\right)}$$

We can now define $\frac{dp}{\rho a}$ in terms of a and integrate:

$$\int \frac{dp}{\rho a} = \int \frac{2da}{\gamma - 1} = \frac{2a}{\gamma - 1}$$

This gives us a simple relationship for $J_\pm$:

$$J_\pm = u \pm \frac{2a}{\gamma - 1}$$

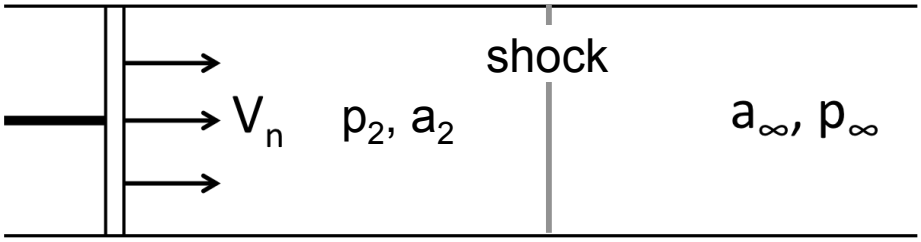


Figure B.1: Alternate domain used in piston theory.

Piston theory compares a hypersonic flowfield to the flow on a piston moving at some speed V_n. This is the situation shown in Figure B.1. p_∞ and a_∞ are the pressure and speed of sound of the non-moving air, while p_2 and a_2 are the pressure and speed of sound behind the shock (or expansion) caused by the moving piston.

Letting $J_\pm$ be constant through the shock, we find that:

$$\begin{aligned} V_n \pm \frac{2a_2}{\gamma-1} &= 0 \pm \frac{2a_\infty}{\gamma-1} \\ a_2 &= a_\infty \mp \frac{\gamma-1}{2} V_n \\ \frac{a_2}{a_\infty} &= 1 \mp \frac{\gamma-1}{2}\frac{V_n}{a_\infty} \\ \frac{T_2}{T_\infty} = \left(\frac{a_2}{a_\infty}\right)^2 &= \left(1 \mp \frac{\gamma-1}{2}\frac{V_n}{a_\infty}\right)^2 \\ \frac{p_2}{p_\infty} = \left(\frac{T_2}{T_\infty}\right)^{\frac{\gamma}{\gamma-1}} &= \left(1 \mp \frac{\gamma-1}{2}\frac{V_n}{a_\infty}\right)^{\frac{2\gamma}{\gamma-1}} \end{aligned}$$

Piston theory gives the pressure on the piston surface. For this reason, we are interested in the left running wave, or C_-. Therefore, we arrive at the following equation for full-order piston theory:

$$\frac{p_2}{p_\infty} = \left(1 + \frac{\gamma-1}{2}\frac{V_n}{a_\infty}\right)^{\frac{2\gamma}{\gamma-1}} \tag{B.4}$$

l theorem:

$$(1+y)^r \approx \sum_{k=0}^{N} \binom{r}{k} y^k$$
$$\approx 1 + ry + \frac{r(r-1)}{2!}y^2 + \frac{r(r-1)(r-2)}{3!}y^3 \ldots,$$

where N is the order of approximation. Note that this approximation is only valid for $y << 1$.

Application of the binomial theorem to Equation B.4 as a third order approximation is performed by substituting $y = \frac{\gamma-1}{2}\frac{V_n}{a_\infty}$ and $r = \frac{2\gamma}{\gamma-1}$:

$$\frac{p_2}{p_\infty} \approx 1 + \frac{2\gamma}{\gamma-1}\frac{\gamma-1}{2}\frac{V_n}{a_\infty} + \frac{1}{2}\frac{2\gamma}{\gamma-1}\left(\frac{2\gamma}{\gamma-1}-1\right)\left(\frac{\gamma-1}{2}\frac{V_n}{a_\infty}\right)^2$$
$$+ \frac{1}{6}\frac{2\gamma}{\gamma-1}\left(\frac{2\gamma}{\gamma-1}-1\right)\left(\frac{2\gamma}{\gamma-1}-2\right)\left(\frac{\gamma-1}{2}\frac{V_n}{a_\infty}\right)^3$$

Simplification yields the expected coefficients:

$$\frac{p_2}{p_\infty} \approx 1 + \gamma\frac{V_n}{a_\infty} + \frac{\gamma(\gamma+1)}{4}\left(\frac{V_n}{a_\infty}\right)^2 + \frac{\gamma(\gamma+1)}{12}\left(\frac{V_n}{a_\infty}\right)^3$$
$$p_2 - p_\infty \approx \gamma p_\infty \frac{V_n}{a_\infty}\left(1 + \frac{\gamma+1}{4}\left(\frac{V_n}{a_\infty}\right) + \frac{\gamma+1}{12}\left(\frac{V_n}{a_\infty}\right)^2\right)$$
$$\Delta p \approx \frac{\rho_\infty U_\infty^2}{M_\infty}\frac{V_n}{U_\infty}\left(1 + \frac{\gamma+1}{4}\left(\frac{V_n}{U_\infty}M_\infty\right) + \frac{\gamma+1}{12}\left(\frac{V_n}{U_\infty}M_\infty\right)^2\right)$$

The final equation is in the same form as Equation 2.7.

APPENDIX C

GEOMETRIC CONSERVATION LAW AND IMPLEMENTATION IN BACKWARDS DIFFERENCE FORMULA

Starting from the semi-discrete Navier-Stokes equations derived for a moving mesh (3.1):

$$\frac{d}{dt}\left(\Omega_i \mathbf{Q}_i\right) = -\sum_{j\in N(i)} \oint_{\partial\Omega_{ij}} \left[\mathbf{F}_p + \mathbf{F}_v - \mathbf{Q}\left(\vec{u}_r \cdot \hat{n}\right)\right] dS \tag{3.1}$$

A state of constant flow is assumed: $\mathbf{Q} = \mathbf{Q}^*$. The viscous forces are dependent on flow gradients, and therefore drop out, resulting in the following:

$$\frac{d}{dt}\left(\Omega_i \mathbf{Q}^*\right) = -\sum_{j\in N(i)} \oint_{\partial\Omega_{ij}} \left[\mathbf{F}_p^* - \mathbf{Q}^*\left(\left(\vec{u}^* - \vec{u}_g\right) \cdot \hat{n}\right)\right] dS \tag{C.1}$$

Recognizing that $\mathbf{F}_p^*$ and $\vec{u}^*$ are constant and are integrated over a closed surface, all of the remaining terms are multiplied by $\mathbf{Q}^*$, which can be divided out:

$$\frac{d}{dt}\Omega_i = \sum_{j\in N(i)} \oint_{\partial\Omega_{ij}} \left(\vec{u}_g \cdot \hat{n}\right) dS \tag{C.2}$$

Equation C.2 is a statement of the Geometric Conservation Law (GCL). In order to satisfy the GCL, the face velocities must be defined in terms of $\frac{d}{dt}\Omega_i$. To find this, we set $\mathbf{Q}^i = \mathbf{Q}^*$ for the second-order backwards difference formula (3.7):

$$\frac{3}{2\Delta t}\Omega^{n+1}\mathbf{Q}^* - \frac{2}{\Delta t}\Omega^n\mathbf{Q}^* + \frac{1}{2\Delta t}\Omega^{n-1}\mathbf{Q}^* = \mathbf{Q}^* \int_{t^n}^{t^{n+1}} \sum_{j\in N(i)} \int_{\partial\Omega_{ij}} \left(\vec{u}_g \cdot \hat{n}\right) dS \; dt \tag{C.3}$$

With full temporal and spatial discretization, integration over the closed surface $\partial\Omega_{ij}$ is performed at time t^{i+1}. This results in:

$$\frac{1}{2\Delta t}\left(3(\Omega^{n+1}-\Omega^{n})-(\Omega^{n}-\Omega^{n+1})\right)=\sum_{j\in N(i)}(\vec{u}_g\cdot\hat{n})^{(n+1)}S_{ij}^{n+1}, \tag{C.4}$$

where $(\vec{u}_g\cdot\hat{n})^{(n+1)}$ is the grid velocity that should be applied when calculating the convective flux through the face.

Equation (C.4) relates the change in volume to the face velocity. This relationship becomes clearer if we split the cell volume changes into their contribution from various faces.

$$\Omega^{n+1}-\Omega^{n}=\sum_{j\in N(i)}\partial\Omega_{ij}^{(n+1)} \tag{C.5}$$

where $\partial\Omega_{ij}^{(k)}$ is the volume swept by face $\partial\Omega_{ij}$ between t^{i} and t^{i+1}. This substitution yields:

$$\frac{1}{2\Delta t}\sum_{j\in N(i)}\left(3\partial\Omega_{ij}^{(n+1)}-\partial\Omega_{ij}^{(n)}\right)=\sum_{j\in N(i)}(\vec{u}_g\cdot\hat{n})^{(n+1)}S_{ij}^{n+1} \tag{C.6}$$

In order to satisfy (C.6), it is sufficient to equate each element of the left-hand side to the corresponding element of the right-hand side:

$$(\vec{u}_g\cdot\hat{n}_{ij})^{(n+1)}=\frac{3\partial\Omega_{ij}^{(n+1)}-\partial\Omega_{ij}^{(n)}}{2\Delta tS_{ij}^{n+1}} \tag{C.7}$$

This value can now be used to calculate fluxes within the moving grid.

APPENDIX D

MEDIAN DUAL-MESH GEOMETRY

A median dual-mesh approach was used to discretize the domain for the RANS solver. One of the primary benefits of the dual-mesh is that it allows the values used for computation to be stored at the nodes of the original mesh. Figure D.1 shows a 2D projection of the median dual-mesh alongside the original mesh. In two dimensions, a dual-cell for a given node is defined by the median points of edges and the centroids of cells neighboring that node. In three-dimensions, a dual-cell for a given node is defined by the median points of edges, centroids of faces, and centroids of cells neighboring that node.

As can be seen in Figure D.1, the median-dual methodology described above is sufficient to handle the transition between quadrilateral and triangular cells. This is the primary benefit of the median-dual cell mesh. However, for large volume ratios between cells (such as the top and bottom of the triangular part of the mesh), the node defining the dual-cell begins to deviate from the actual location of the dual-cell centroid.

This has the potential to interfere with the linear reconstruction used for second-order calculations. For the sake of consistency, the value assigned to a node should be the average value of the solution over the dual-cell defined by the node. However, if the node is not the centroid of the dual-cell, then the average value of the linear reconstruction is not identically equal to the value at the node, which is an inconsistent formulation. The same argument can be made for the quadrature points of the

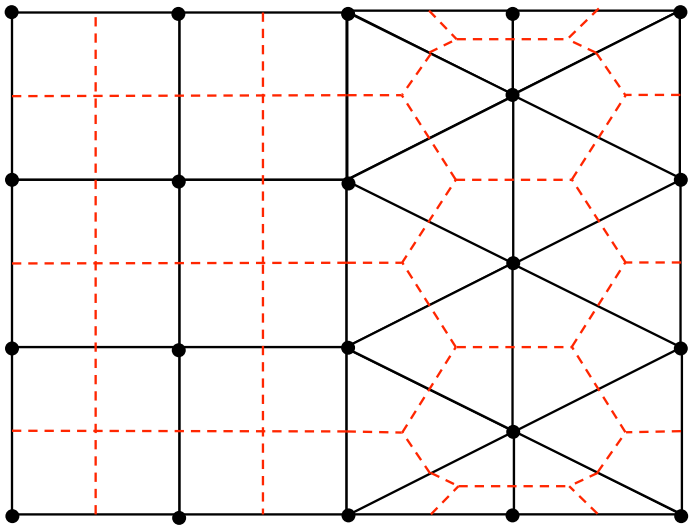


Figure D.1: 2D projection of median dual-mesh (red dashed lines) shown with original mesh (black lines).

faces. For this reason, the volume ratio between cells should be kept to a minimum.

APPENDIX E

MOVING REGRESSION ANALYSIS

A polynomial regression analysis of a dataset $(\mathbf{x}, \mathbf{y})$, where $\mathbf{y} \approx c_0 + c_1\mathbf{x} + \ldots + c_{m-1}\mathbf{x}^{m-1} + c_m\mathbf{x}^m$ can be simplified to a least squares problem using the polynomial basis $\mathbf{P}$. The data and polynomial parameters are defined in the current analysis as:

$$\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{Bmatrix} \qquad \mathbf{y} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{Bmatrix} \qquad \mathbf{c} = \begin{Bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \\ c_m \end{Bmatrix},$$

where n is the size of the dataset, and

$$\mathbf{P} = \begin{Bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \ldots & x_{n-1}^m \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{Bmatrix},$$

where m is the order of the polynomial regression.

This can be written as a system of n dependent linear equations as $\mathbf{y} \approx \mathbf{Pc}$ and simplified to a system of m independent linear equations as $\mathbf{P^T y} \approx \mathbf{P^T P c}$. This equation is then solved for the polynomial coefficients $\mathbf{c}$.

The moving polynomial regression uses only points in the local neighborhood

of the data point in question. This means that to approximate a value y_i, only the points $(x, y)_k, k = i - \ell ... i + \ell$ are used in the polynomial regression. This allows a simple polynomial to form a reasonable approximation of sections of the very complex data created by panel flutter simulations.

APPENDIX F

PANEL RESPONSE TO A PRESSURE LOAD

In this appendix, the response to a pressure load is investigated for panels with substructures A and B. In particular, the underlying relationships between the stringer thickness and the translational and rotational stiffness of the stringers is determined.

F.1. Panel with Substructure A

The panel with substructure A used stringers which modified the torsional rigidity of the panel along its edges. In ABAQUS, a 1000 Pa load was applied to the entire panel. The maximum panel displacement Δz and maximum rotation on the stringer θ were noted. These data were used to determine the effective panel stiffness and stringer torsional rigidity in Section VI.2.3.

Table F.1 shows the raw displacement and rotation data from ABAQUS, the final relationship based on a power law analysis, and the intermediate data used for the analysis. The idea behind the power law analysis is that the various quantities follow a relationship governed by the equation $A_h = A_0 + (A_1 - A_0)(h_s/h)^k$, where A_h is the quantity of interest, A_0 is the value provided by the panel, and $A_1 - A_0$ is the incremental increase provided by a stringer with thickness $h_s = h$.

No clear relationship exists between the stringer thickness and the panel stiffness, except that it is sublinear. As discussed in VI.2.3, the panel is asymptotically approaching the case with rigid stringers, and this is difficult to capture with a power

Table F.1: ABAQUS analysis for panel with substructure A.

h_s/h	Δz [μm]	$K_{p,h}/K_{p,0}$	$\frac{K_{p,h}-K_{p,0}}{K_{p,1}-K_{p,0}}$	θ_s [rad]	JG_h/JG_0	$\frac{JG_h-JG_0}{JG_1-JG_0}$	k_{JG}
0	111.9	1.000	0.000	7.436E-04	1.000	0.000	—
1	66.82	1.675	1.000	3.605E-04	2.062	1.000	—
2	44.44	2.518	2.248	1.743E-04	4.267	3.075	1.621
3	35.55	3.148	3.181	9.962E-05	7.465	6.085	1.644
5	29.30	3.819	4.176	4.681E-05	15.88	14.01	1.640
10	25.76	4.344	4.953	1.677E-05	44.34	40.80	1.611

law analysis.

The relationship between the stringer thickness and the torsional rigidity is shown to be approximately as follows:

$$JG_h = JG_0 + (JG_1 - JG_0)(h_s/h)^{1.63},$$

where JG_h is the torsional rigidity of the panel and stringer, JG_1 is the rotational stiffness for $h_s/h = 1$, and JG_0 is the torsional rigidity from the panel alone. Based on the power of the exponent, the relationship between the stringer thickness and torsional rigidity is more than linear and less than quadratic.

F.2. Panel with Substructure B

The panel with substructure B used stringers which modified the bending stiffness of the panel through reduction of the effective panel width and the addition of torsional rigidity along its stringers. In ABAQUS, a 1000 Pa load was applied to the entire panel. The maximum panel displacement Δz, the maximum stringer displacement Δz_s, and maximum rotation on the stringer θ were noted. These data

were used to determine the effective panel stiffness, stringer bending stiffness, and stringer torsional rigidity in Section VI.3.3.

Table F.2 shows the raw displacement and rotation data from ABAQUS, the final relationship based on a power law analysis, and the intermediate data used for the analysis.

Table F.2: ABAQUS analysis for panel with substructure B.

h_s/h	Δz_p [μm]	Δz_s [μm]	$\frac{K_{s,h}-K_{s,0}}{K_{s,1}-K_{s,0}}$	k_{Ks}	θ_s [rad]	$\frac{JG_h-JG_0}{JG_1-JG_0}$	k_{JG}
0	1782.5	1289.2	0.000	—	3.85E-3	0.000	—
1	665.78	450.70	1.000	—	1.46E-3	1.000	—
3	150.88	86.231	7.498	1.834	2.73E-4	7.999	1.893
5	69.109	28.129	24.10	1.977	8.94E-5	25.71	2.017
10	34.650	5.3818	128.2	2.108	1.85E-5	126.5	2.102

The panel displacement Δz_p is similar to the stringer displacement Δz_s, but diverges at higher stringer thicknesses. As discussed in VI.2.3, the panel is asymptotically approaching the case with rigid stringers, and this is difficult to capture with a power law analysis.

The relationship between the stringer thickness and both the stringer bending stiffness K_s and torsional rigidity JG_s was approximately quadratic.

APPENDIX G

CASE GENERATION PROCEDURE

In order to run an aerothermoelastic simulation in the solver discussed in Chapter III, the following data is needed:

- Beta mesh files
  - Volume meshes (matching for structural and thermal solvers)
  - Surface meshes for Neumann boundary conditions
- UNS3D mesh files
- Files containing coupling information (`cfd2fea.dat` and `fea2cfd.dat`).
- Input files describing individual run

The following sections will detail how each of these files is generated.

Beta Mesh Files

Beta requires a volume mesh for any problem, and additionally requires surface meshes for any surfaces that are coupled or have Neumann boundary conditions.

Both the Beta volume mesh file and the Beta surface mesh file have the following format:

- Header, consisting of:

  ```
  #nodes, #elements, #dimensions
  ```

- Spatial coordinates for each node:

```
node # (0 based), xi, yi, zi
```

- Cell to node connectivity for each element:

```
element # (0 based), # nodes/element, node 0, node 1, ..., node n-1
```

In practice, the mesh is generated in UGRID format in a grid generation program such as Pointwise. The UGRID format allows inclusion of boundary conditions on the surfaces of the volume, which can be used to define which surfaces are needed for the boundary conditions.

The UGRID file is converted via a utility program named BetaPrep (short for Beta Preprocessor). Given a UGRID file and a short input file to specify filenames, BetaPrep can generate the volume and surface mesh using either 8-node brick and 4-node surface elements or 20-node brick and 8-node surface elements (the undeformed edges are assumed to vary linearly in space). The source code for BetaPrep is given in a separate appendix.

UNS3D Mesh Files

UNS3D also requires a volume mesh, and also requires a separate file with the cell to node connectivity information, which is used primarily for output to Tecplot format.

The UNS3D volume mesh has the following format:

- Header information, organized as follows:

```
#elements, #nodes, #edges, #faces, #boundary faces
```

- Spatial coordinates for each node:

  ```
  xi, yi, zi
  ```

- Boundary face to node information and boundary condition:

  ```
  #points on boundary face, node 1, node 2 ..., node n, boundary condition
  ID #
  ```

- Cell to face information

  ```
  #faces in element face 1, face 2, ..., face n
  ```

- Face to edge information

  ```
  #edges in face edge 1, edge 2, ..., edge n
  ```

- Edge to node information

  ```
  node 1, node 2
  ```

The cell to node connectivity file has the following format:

- Header information, organized as follows:

Similar to the Beta mesh file generation, the UNS3D mesh file is generated first in UGRID format using a grid generation program such as Pointwise. This UGRID file is given to a preprocessing program known as Prep along with a short input file to specify file names. Prep then generates the volume mesh file and the cell connectivity file. The source code for Prep is given in a separate appendix.

Once the mesh has been generated, it must still be split for parallelization using domain decomposition. This is done using a utility called SplitMesh. Given a

volume mesh file and a cell connectivity file, SplitMesh gives volume mesh and cell connectivity files for each processor along with a file that contains the local node number, processor, and global node number for every node in the mesh (named "`loc2glob.dat_###`", where `###` is the number of processors for the split). The "`loc2glob`" file has the following format:

- Header information, organized as follows:

  ```
  # of total nodes (includes some repeats), # of global nodes, maximum
  # of nodes on a single processor, # of processors
  ```

- For each node on every processor (may be repeats of nodes in a global sense):

  ```
  Local node #, Processor #, Global node #
  ```

Coupling Information Files

There are two files that contain the interpolation data between UNS3D and Beta. The first, titled "cfd2fea.dat" contains the interpolation from the UNS3D surface mesh to the Beta surface mesh. The second, titled "fea2cfd.dat" contains the interpolation from the Beta surface mesh to the UNS3D surface mesh. It has the following format:

- For each destination node:

  ```
  Destination node #, Source flag, Number of source nodes, Source node
  1, Source weight 1, ..., Source node n, Source weight n
  ```

The source flag indicates whether the node for the interpolation is based on a node in the UNS3D mesh (a value of 1) or a node in the Beta mesh (a value of 0).

The coupling file is generated in a two step process. First, the desired surface mesh is extracted from the UNS3D volume mesh. Typically, the entire viscous wall is used to generate the surface mesh, though a subset could be used. This is done with a simple program named SurfaceMesh.

The UNS3D and Beta surface meshes are then compared in a program call BetaInterp, which once again has a simple input file containing the names of the surface files and the output interpolation files. BetaInterp uses an octree sort to match nodes between the two surfaces, which has more efficient scaling than a brute force search.

At this point, BetaInterp is designed only for matching interfaces (nodes on each surface are located at the same point in space). It could easily be expanded to non-matching interfaces using Moving Least Squares or another interpolation methodology. Again, the source code is given in a separate appendix.

Run Input Files

This section contains the input files which control how each solver runs. Additionally, any additional required input files are defined for each solver.

UNS3D Input

UNS3D requires an input file in the following format:

```
&cardf
title        = "Spherical Protuberance"  ! Title of Run
tecplot_name = "plt/SP_io7.plt"          ! Name of Tecplot format file
dump_tecplot = T                         ! Logical for Tecplot output
case_name    = "yp/SP_io7"               ! Name of surface output file
dump_yplus   = T                         ! Logical for surface output
rsdfile      = "dat/io7.dat"             ! Text output file
```

```
relative_V   = F                          ! Output option for velocities
!                                         !  (removes frame rotation)
/
&cardg
gridfile     = "mesh/vol.mesh"  ! Filename for volume mesh (base file)
c2nfile      = "mesh/c2n.def"   ! Filename for connectivity
/
&cardh
filedin      = "out/sp6.out"    ! Filename for restart input
filedout     = "out/sp7.out"    ! Filename for restart output
tempin       = "out/vol.q12"    ! Filename for unsteady input
/
&cardi
fileturin    = "out/none3.tur"  ! Filename for turbulence restart input
fileturout   = "out/none4.tur"  ! Filename for turbulence restart output
/
&cardj
profin = "prof.in"      ! Filename for inlet profile
use_profin = .false.    ! Option for profile or constant vel. inlet
/
&cardk
/
&card0
noblade      = 1    ! Number of domains for annular periodicity
igeom        = 1    ! Annular/linear periodicity, if applicable
inbc         = 6    ! Inlet boundary condition type
ioutbc       = 3    ! Outlet boundary condition type
ispet        = 0    ! Wall boundary condition (0 - adiabatic,
!                   !    1 - isothermal with twall, 2 - coupled to beta_t)
iwall        = 3    ! Inviscid wall boundary condition
/
&card1
istep        = 1         ! 1 for restart, 0 for initial case
readq12      = F         ! Option for reading unsteady restart input
iturm        = 1         ! 1 for turbulent restart,
!                        ! 0 for initial turbulence
npseudotimesteps = 3000  ! Number of pseudo-time steps for each time step
mtime        = 200       ! Number of time steps in real time
/
&card2
preconditioned = .true.  ! Option for preconditioning of inviscid fluxes
iorder       = 2         ! Spatial order of accuracy specification
use_limiter  = .true.    ! Option for limiter use
invis        = F         ! Option for inviscid/viscous flows
lamin        = F         ! Option for laminar/turbulent flows
```

```
mstg         = 4         ! Number of stages in Runge-Kutta integration
irhsm        = 0         ! Number of implicit residual smoothing iterations
lsgg         = 3         ! Type of gradient calculation (3 -> WENO)
steady       = F         ! Option for steady/unsteady flows
typlim       = 5         ! Type of limiter calculation (5 -> Flux Limiter)
fluxtype     = 3         ! Type of flux calculation (3 -> RoeM)
/
&card2a
imp          = .false.   ! Disable linearized implicit calculation
/
&card3
cfl          = 0.5d0     ! Courant-Friedrichs-Lewy number (pseudo-time step)
dtimedim     = 0.001d0   ! Real-time step
/
&card4
tintens      = 0.002d0   ! Turbulent intensity
tlength      = 0.001d0   ! Turbulent length scale
u0           = 0.035d0   ! Initial Mach number (and inlet for supersonic)
alfaz        = 0.0d0     ! Rotation of inlet velocity toward z-axis (first)
alfax        = 0.0d0     ! Rotation of inlet velocity about x-axis (second)
/
&card5
ptot         = 101413.d0 ! Total pressure at inlet
ttot         = 228.22d0  ! Total temperature at inlet
pback        = 101325d0  ! Back pressure at outlet
twall        = 300.0d0   ! Wall temperature for ispet = 1
/
&card6
pref         = 101325d0  ! Reference pressure for nondimensionalization
tref         = 288.15d0  ! Reference temperature for nondimensionalization
scale        =      1.0  ! Nondimensionalization spatial scale
/
&card7
/
&card8
omegax       = 0.0d0     ! Rotation speed (in RPM)
/
&card9
echo         = F         ! Option for output verbosity
debug        = F         ! Option for output verbosity
intev_freq   = 1         ! Frequency of integration output in real-time steps
intev_freq_pt= 3000      ! Frequency of integration output in pseudo-time steps
res_freq     = 1         ! Frequency of residual output in real-time steps
res_freq_pt  = 1000      ! Frequency of residual output in real-time steps
itersave     = 25        ! Frequency of restart output in real-time steps
```

```
q_corrctn_limit = 1.d-12 ! Limit for residuals before iteration stops
q_corrctn_ratio = 7.0d0  ! Limit for residual ratio to initial residual before
!                        !   iteration stops
MonitorMaxMach     = F   ! Option for Mach number monitoring
MonitorMaxTemp     = F   ! Option for temperature monitoring
MaxMachThreshold   = 2.5 ! Threshold for Mach number monitoring
reset_iter_counter = F   ! Option for restart file
Force_l      = T         ! Option for force integration and reporting
makemovie = T            ! Option for movie generation
mov_start_num = 1        ! Start number for movie generation
movie_freq = 25          ! Frequency of frame recording for movie generation
!                        !   in real-time steps
/
&card10
/
&cardforced
def_type = "beta"     ! Type of deformation, can be:
!                     !  "none" -- no deformation
!                     !  "beta" -- deformation specified by Beta
!                     !  "rbf_plunge" -- linear motion of airfoil
!                     !  "rbf_pitch" -- angular motion of airfoil
!                     !  "fromfile" -- Deformation specified in file
def_wave = "cos"      ! Time variation of forced deformation
amp = 1.0             ! Amplitude of forced deformation
forced_freq = 1.0     ! Frequency of forced deformation
sparse_rbf = .true.   ! Option specifying use of MUMPS for radial
!                     !     basis function interpolation
pitch_rrat = 0.1d0    ! Size of support for RBF interpolation
l2gfile = "mesh/loc2glob.dat" ! "loc2glob" filename
forced_file = "mesh/vol.def"  ! Filename for deformed mesh
ea(1) = 0.0           ! Elastic axis for pitch deformation
ea(2) = 0.0           ! Elastic axis for pitch deformation
axisdir = 3           ! Axis direction for pitch deformation
/
&cardrom
/
&cardprecon
eps_minur = 0.035d0   ! Minimum relative velocity for preconditioning
/
&vortex
/
&beta
UNS3D_GROUP_KEY = 0   ! Communication key for coupling
BETAS_GROUP_KEY = 1   ! Communication key for coupling
BETAT_GROUP_KEY = 2   ! Communication key for coupling
```

```
flux_factor = 1.0d0   ! Factor to multiply heat flux
subit_freq = 50       ! Number of pseudo-time steps between each
!                     !   subiteration in aeroelastic coupling
/
```

Beta Structural Input

Beta-S requires an input file in the following format:

```
createModel
TransientElasticityImplicitLinearModel // Defines structural model
exitcreateModel

ReadTitle
simple.beta  // Title of run

setVerboseFlag                  0   // Determines how much is written to screen

SolverSettings
SetSolverVerboseFlag    0           // Determines how much is written to screen
setStorageMethod        MKLpardiso  // Setting for how matrices are stored
setReorderingScheme     PAR         // Reordering scheme
UseMultiCoreSolver      10          // How many cores may be used (OpenMP)
ExitSolverSettings

BasicElement::SetAnalysisType
1                                   // Geometrically non-linear analysis

CreateElements
968                                 // Number of elements in mesh
TransientElasticityElement3D        // Type of element
all                                 // Uniform element type
-1
exitCreateElements

openFile Orthotropic.matlib ReadMaterials  // Opens another file to read materials

openFile 3Dplate.b3d ReadMesh              // Opens another file to read mesh

setNumDofPerNode                           // Set dimension of run to 3
3 all
-1

SetElementProperty                         // Set material for elements
```

```
selectElementMaterial
all 1            // All elements set to material 1
-1

SetElementProperty                          // Integration order for elements
setIntegrationOrder
all 3
-1 0 0

DefineCurve                                 // Define a piecewise linear curve
// Curve 0: Time Vs Load
// Used for traction in x and z directions
0.0     0.0
120.0   0.0
exitDefineCurve

DefineCurve                                 // Define a piecewise linear curve
// Curve 1: Time Vs Load
// Used for traction in y direction
0.0     0.0d0
0.01   -1.0d0
0.02    0.0d0
200.0   0.0d0
exitDefineCurve

ReadLoads                                   // Set up Neumann conditions
SurfaceTraction
3Dplate.b2d CFDMPI 1 0     // Couple surface in input file to CFD
//  Surface File -- Traction type -- Local Comm Key -- UNS3D Comm key
  end
SurfaceTraction
3Dplate_bot.b2d 0 1 0   // Create constant load based on curves
     //    x, z -- Curve 0
     //       y -- Curve 1
  end
exitReadLoads

CreateNodeGroup                      // Set up nodes for Dirichlet conditions
               // Nodes on a line:  (x1, y1, z1)  to  (x2, y2, z2)
NodeSet1 OrderedNodesOnLine -0.25 0.0 -0.25  0.25 0.0 -0.25
NodeSet1 OrderedNodesOnLine -0.25 0.0  0.25  0.25 0.0  0.25
NodeSet2 OrderedNodesOnLine -0.25 0.0 -0.25 -0.25 0.0  0.25
NodeSet2 OrderedNodesOnLine  0.25 0.0 -0.25  0.25 0.0  0.25
NodeSet3 NodesOnPlane 1 0.0   // All nodes on plane x = 0
NodeSet3 NodesOnPlane 1 0.5
```

```
NodeSet4 NodesOnPlane 3 0.0
NodeSet4 NodesOnPlane 3 0.5
exitCreateNodeGroup

readConstraints                     // Define Dirichlet conditions
    NodeSet1 1                      // Restrain NodeSet1 in x direction
    NodeSet1 2                      // Restrain NodeSet1 in y direction
    NodeSet1 3                      // Restrain NodeSet1 in z direction
    NodeSet2 1
    NodeSet2 2
    NodeSet2 3
exitreadConstraints

SetElementProperty
SetElementTemperature               // Define initial temperature of elements
all 0.0
-1 0 0

ReadTransientAnalysisParameters
InitialVelocityAllNodes 0.0        // Sets initial velocity of nodes
SaveRestartData ate_restart.bin    // Saves data for restart
// LoadRestartData                 // Loads restart data (commented for first run)
SetTimeStep     0.0001             // Time step
SetNumTimeStep  50000              // Number of time steps
SetOptionalOutTimeStepSize 0.0001  // Frequency of output
SetMPIGroupKey 1                   // Communication key of Structural
CoupleToThermalwithMPI 2           // Communication key of Thermal
SetdeltaThermalUpdateT 0.0001      // Update frequency with Thermal
CoupleToCFDwithMPI 0 4223          // Communication key of UNS3D and
//                                 //   number of nodes in communication
SetNewmarkBeta    0.25             // Parameters for time-stepping
SetNewmarkGamma   0.5              // Parameters for time-stepping
IterateWithCFD 5 1.0e-6            // Subiteration procedure with UNS3D
exitReadTransientAnalysisParameters

ReadOptionalOutput          // Additional output
displacements 10            // Write the displacements every 10 iterations
exitReadOptionalOutput

DoAnalysis                  // Run the analysis

end
```

The structural material file (named "Orthotropic.matlib" in the input above) has the following format:

```
ElasticMaterialWithDensity          // Type of material
1 "Soft Orthotropic Material"       // Reference Number and
readModuli
200.0e9  200.0e9  200.0e9           // Young's Modulus
0.3      0.3      0.3               // Poisson's Ratio
76.923e9 76.923e9 76.923e9          // Shear Modulus
7850                                // Density
readThermalExpansionCoefficients
3.5e-006 3.5e-006 3.5e-006          // Coefficient of Thermal Expansion
exitElasticMaterial

exitReadMaterials
```

Beta Thermal Input

Beta-T requires an input file in the following format:

```
createModel
TransientHeatTransferModel  // Defines thermal model
exitcreateModel

ReadTitle
simple.beta   // Title of run

setVerboseFlag  0    // Determines how much is written to screen

SolverSettings
setSolverVerboseFlag          0          // Determines amount written to screen
setStorageMethod              mklpardiso // Setting for how matrices are stored
setSparseSolverMaxIterations  100000     // Maximum solver iterations
setSparseSolverTolerance      1e-8       // Tolerance for solver
replaceZeroDiagonal           1.0        // Avoid singular matrix
SetReorderingScheme           PAR        // Reordering scheme for matrix
setUsePreviousFillInOrdering  1          // Option for fill-in ordering
UseMultiCoreSolver            10         // How many cores may be used (OpenMP)
ExitSolverSettings

UseMultiCoreAssembly max

BasicElement::SetAnalysisType
0                                        // Linear analysis

maxResidual
1e-5                                     // Residual required to advance
```

```
CreateElements
968                                     // Number of elements in mesh
TransientHeatTransferElement3D          // Type of element
all                                     // Uniform element type
-1
exitCreateElements

openFile ThermalMaterial.mat ReadMaterials  // Opens another file to read materials

openFile 3Dplate.b3d ReadMesh               // Opens another file to read mesh

setNumDofPerNode                            // Only one temperature per node
1 all
-1

SetElementProperty                        // Set material for elements
selectElementMaterial
all 1            // All elements set to material 1
-1

SetElementProperty                         // Integration order for elements
setIntegrationOrder
all 3
-1 0 0


CreateNodeGroup                           // Set nodes for Dirichlet BC
NodeSet1 NodesOnPlane 1 0.0
NodeSet1 NodesOnPlane 1 0.5
NodeSet2 NodesOnPlane 3 0.0
NodeSet2 NodesOnPlane 3 0.5
exitCreateNodeGroup

readConstraints                           // Define Dirichlet BC (if applicable)
exitreadConstraints

ReadLoads                                   // Set up Neumann conditions
SurfaceFlux
3Dplate.b2d CFDMPI 2 0     // Couple surface in input file to CFD
//  Surface File -- Traction type -- Local Comm Key -- UNS3D Comm key
  end
exitReadLoads

setTimeIntegrationAlpha 0.0  // Alpha for integration method (set to Euler)
setTimeUnit seconds          // Time unit (only affects output)
```

```
ReadOptionalOutput
concentration 1      // Output (concentration gives temperature)
exitReadOptionalOutput

ReadTransientAnalysisParameters
InitialConditions         0.0d0            // Set initial Temp to 0
SaveRestartData           ate_restart.bin  // Restart filename
// LoadRestartData                         // Loads restart data (commented for first run)
SetTimeStep               0.0001           // Time step
SetNumTimeStep            50000            // Number of time steps
SetMPIGroupKey            2                // Communication key of Thermal
CoupleToMechanicalWithMPI 1                // Communication key of Structural
SetdeltaThermalUpdateT    0.0001           // Frequency of output
CoupleToCFDWithMPI        0 4223           // Communication key of UNS3D and
//                                         //   number of nodes in communication
exitReadTransientAnalysisParameters


DoAnalysis                  // Run the analysis

end
```

The thermal material file (named "ThermalMaterial.mat" in the input above) has the following format:

```
TransientHeatTransferMaterial
1 "Aluminium Heat Transfer Material"   // Reference Number and Name
ReadDensity
2.7e03                                 // Density of Material
ReadSpecificHeat
902.0                                  // Specific Heat of Material
ReadConductivity
237.0 237.0 237.0                      // Thermal Conductivity of Material
ExitTransientHeatTransferMaterial
ExitReadMaterials
```

G.1. Codes Included as Separate Files

The following FORTRAN 90/95 codes are included as separate files:

- BetaInterp (Interpolation generator between Beta and UNS3D)
- BetaPrep (Conversion code from .ugrid file to Beta mesh file)
- Prep (Conversion code from .ugrid file to UNS3D mesh file)
- SplitMesh (Parallelization preprocessing code)
- SurfaceMesh (Surface extraction code for UNS3D mesh files)

Each code is compressed as a tarball (.tgz) file. Inside the tarball, the source code and a makefile are included, which should be enough to generate an executable with a suitable FORTRAN 90/95 compiler, such as gfortran.

Also included are the subroutines which were modified or added to the RANS solver. These are included as a separate tarball named RANS_subs.tgz.