A TIME-LINEARIZED NAVIER-STOKES SOLVER FOR ANNULAR GAS SEAL ROTORDYNAMIC ANALYSIS

A Dissertation

by

DAVID NOEL LILIEDAHL

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,	Thomas W. Strganac
	Diego Donzis
	Gerald L. Morrison
Department Head,	Rodney Bowersox

August 2017

Major Subject: Aerospace Engineering

ABSTRACT

A time-linearized CFD solver for analyzing rotordynamics of gas seals is presented offering an improvement over existing linearized CFD solvers. Previous linearized solvers required structured grids and axisymmetric domains, limiting the complexity of the geometries of the seals that could be analyzed. A preexisting, full-order, in-house CFD solver was available which operated on fully 3D and unstructured grids and was well suited for complex seal geometries. A linearized version of the in-house code is developed as a companion to the full-order solver, retaining its unstructured and fully 3D features. Furthermore, boundary conditions are developed for the linearized solver allowing it to take advantage of the geometric symmetries that were required by earlier linearized solvers without necessarily being limited to them. Additionally, a linearization procedure is presented which is general enough to be used for the many various features of the full-order solver. As the in-house code continues to be developed and new features are included, the same linearization procedure can be used to keep the companion code up to date. The full-order, in-house solver and the time-linearized companion code combine to become a powerful CFD-perturbation solver accessible to all complexities of seal geometries.

This dissertation also presents an analytical formula that describes features of cavity flow as it pertains to annular gas seals in order to progress the fundamental understanding of the flow physics of roughened seals. An existing semi-empirical analytical formula, developed to describe the cavity flow of aircraft bomb bays, is modified using the full-order, in-house CFD solver. The numeric model is validated against experimental measurements and used to adjust empirical parameters of the formula to match cavity flow conditions unique to annular seals. The modified analytical formula is able to predict features of cavity flows found in annular gas seals better than existing formulae.

Finally, the companion, time-linearized CFD solver is verified using two simple cases and the combined full-order and time-linearized CFD-perturbation solver is used to predict rotordynamic properties for two gas seal geometries. The first case used to verify the linearized solver is a channel flow with an oscillating back-pressure and the second is a stationary flow with an oscillating wall. The first gas seal case the combined CFD-perturbation solver is used for is a straight seal based on the High Pressure Oxidizer Turbopump (HPOTP) of the Space Shuttle Main Engine (SSME). The second is a stepped labyrinth seal. The rotordynamic predictions are compared with established bulk-flow models of the two cases and conclusions are presented.

DEDICATION

Wesley & Madeline

ACKNOWLEDGMENTS

I would like to offer a special thanks to Dr. Cizmas for his advice, guidance and patience. I would also like to thank Tom, Forrest, Will, Robert, Raymond, Brian, Neil and all the other fellow students near and far for their assistance and lively discussions throughout the years.

But mostly I would like to thank my wife Michelle for her love, support, patience and gentle pressure without which I would have never gotten to this point.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Paul G. A. Cizmas, advisor, and Professors Thomas W. Strganac and Diego Donzis of the Department of Aerospace Engineering and Professor Gerald L. Morrison of the Department of Mechanical Engineering.

The data analyzed for Section 4.1 were provided in part by Professor Dara Childs of the Turbomachinery Labrotory of Texas A&M University. The analyses depicted in Section 4.1.3 were conducted in part by Forrest Carpenter of the Department of Aerospace Engineering.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by the Turbomachinery Research Consortium.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

NOMENCLATURE

$\mathbf{A}$	flux Jacobian matrix
$\tilde{A}_{ij}$	Roe's matrix
c	cross coupled damping coefficient
C	direct damping coefficient
c_p	specific heat at constant pressure
c_o	speed of sound outside cavity
c_t	speed of sound inside cavity
c_v	specific heat at constant volume
$\mathbf{D}$	block diagonal matrix
e	internal energy
E	total energy
f	frequency
$\mathbf{F}^{\star}$	filter matrix
$\vec{F}_c$	convective flux vector
F_n	force in n-direction
$\vec{F}_v$	viscous flux vector
$\vec{G}_v$	body source
$\mathbf{G}_s$	surface source
h	internal enthalpy
H	total enthalpy
$\mathbf{I}$	identity tensor
J	Jacobian matrix
$\hat{k}$	turbulent kinetic energy
k	thermal conductivity
k	cross coupled stiffness coefficient
K	direct stiffness coefficient
k_x	wave number
L	length of cavity
$\mathbf{L}$	block lower triangular matrix
$\mathbf{L}$	primitive to conservative transformation matrix
M	Mach number
$\vec{m}$	momentum
$\hat{n}$	normal unit vector of surface
N_i	number of neighbors of node i
n_n	component of unit normal vector in n-direction
$\hat{n}_v$	direction from boundary to interior node

p	pressure
$p^\star$	total pressure
$\mathbf{P}_i$	least squares gradient matrix
Pr	Prandtl number
Pr_T	turbulent Prandtl number
q	general state variable
$\vec{Q}$	state vector
$\overline{q}$	mean of variable
q'	fluctuation of variable about mean
Δq	difference between variable
$\hat{q}$	Favre average of variable
q''	fluctuation of variable about Favre average
$\vec{Q}_c$	vector of conservative variables
$\vec{Q}_m$	master state vector
$\vec{Q}_p$	vector of primitive variables
$\vec{Q}_s$	slave state vector
R	specific gas constant
R	radius
$\Re$	real part of complex value
R^-	outgoing Riemann invariant
R^+	incoming Riemann invariant
s	entropy
dS	infinitesimal elements of $\partial\Omega$
S_{ij}	surface area of face ij
S_n	projection of volume in the n-direction
t	time
T	temperature
$T^\star$	total temperature
$\mathbf{T}$	eigenvector matrix
$\mathbf{T}_{ms}$	rotation matrix
U	free stream flow velocity
$\mathbf{U}$	block upper triangular matrix
$\mathbf{U}_{ms}$	periodicity matrix
$\vec{v}$	velocity vector
V	contravariant velocity
v_g	group velocity
v_n	normal flow velocity
v_t	tangential flow velocity
V_t	contravariant velocity of face
w	gradient weight coefficient
X	rotor position in radial x-direction
Y	rotor position in radial y-direction

α	adjustable coefficient
α	inlet flow angle
α_i	Runge-Kutta stage
β	inlet flow angle
β	adjustable weight coefficient
γ	ratio of specific heats
γ	empirical constant - phase shift between shear layer and pressure oscillations
ϵ	small value
θ	angle from z-axis
θ	rotation angle
θ	phase angle
θ_n	component of work in n-direction
κ	empirical constant - ratio of vorticity convection and freestream velocities
κ	diffusivity coefficient
λ	eigenvalue
$\mathbf{\Lambda}$	eigenvalue matrix
Λ_c	convective spectral radii
Λ_v	viscous spectral radii
μ	dynamic viscosity coefficient
ρ	density
σ	CFL number
τ	viscous stress tensor
τ_{ij}	component of stress tensor
$\vec{\phi}$	vector of characteristic variables
Φ	general conserved constant
Ψ_k	limiter function
ω	whirl frequency
Ω	finite region of domain
Ω_i	volume of cell i
$\partial\Omega$	surface of Ω

1. INTRODUCTION

1.1 Motivation and Background

Gas seals are used in turbines and compressors to isolate internal areas of turbomachinery. They are found in areas of the machinery where the rotor and the stator come in very close proximity to each other. Because of the relative motion between the rotor and stator, a small gap between the two must remain to permit vibrations. Unfortunately, leaving a gap also allows unwanted leakage of gas between otherwise isolated areas of machinery diminishing overall effectiveness of the compressor or turbine. The primary role of gas seals is to control this leakage flow.

Gas seals are able to control or limit unwanted leakage flow by creating a frictional flow path between the rotor and stator. As large pressure differences force gas through the gap, large viscous forces dissipate internal energy effectively slowing the flow of leaked gas. This forces the gas along the intended path within the machinery. The geometries of gas seals are designed in such a way as to maximize the viscous losses and act as a hindrance to leakage flow while simultaneously providing enough clearance for the rotor to travel freely.

The large pressure differences and small length scales seen by gas seals result in large aerodynamic forces. These forces serve as a major contribution to overall rotordynamic stability of the machinery leading to a secondary, however, just as important, role for gas seals. This secondary role is control of rotordynamic stability. Recent demand for increased power and efficiency has led to higher operating speeds and working pressures and has further increased the importance of the role of the gas seals. Consequently, this has lead to an increased interest in analysing the aerodynamic and rotordynamic properties of different gas seal designs.

Three general methods are used to predict the aerodynamic and rotordynamic properties of gas seals. The first method is based on simplifying bulk flow assumptions and produces simple, computationally inexpensive analytical equations. However, under certain operating and testing conditions, bulk flow models fail to accurately predict aerodynamic forces. Because of the bulk flow assumptions, little insight is provided to why these methods fail. The second method is based on computational fluid dynamic (CFD) methods with small perturbation approximations. These CFD-perturbation methods use only a time-linearized assumption and are applicable to a larger range of conditions than bulk flow methods. Solutions must be found numerically making this method more computationally expensive than the bulk flow method. However, all of the unsteady dynamics are confined to linear perturbations and only a single steady, non-linear, solution is needed. This makes CFD-perturbation methods much less expensive than the third method. This final method uses time-accurate CFD to generate time dependent flow fields without any linear or bulk flow assumptions. They are very computationally expensive due to the need to resolve every step in the time marching procedure as well as needing to rerun the entire simulation for different conditions. Taking all these arguments into consideration, a CFD-perturbation method based on time-linearization, based on its balance between efficiency and accuracy, is used herein to predict rotordynamic coefficients for annular gas seals.

CFD-perturbation methods were first used to predict rotordynamic coefficients by Diezen and Nordmann [1]. They developed a quasi-3D method which required axisymmetric domains and structured grids. The method was further developed by Kim and Rhode [2] and Xi and Rhode [3] to include more complex geometries. Their method, however, still required axisymmetric domains and structured grids. They also introduced an incompressibility assumption.

Similar CFD-perturbation methods have been used to analyze cascaded airfoil geometries. The method developed by Hall and Lorence [4] allowed for non-axisymmetric domains but assumed inviscid and incompressible flows. Viscous effects in the time-linearization method were introduced by Cizmas and Hall [5] using an interacting viscous-inviscid method. Clark and Hall [6] applied the time-linearization method to the RANS equations.

All previous research using CFD-perturbation methods for turbomachinery analysis required structured grids and nearly all required axisymmetric domains. These limitations greatly restrict the complexity of the geometries accessible to this method. The current methods need to be expanded so they may be used to study the rotordynamics of a greater variety gas seals.

The time-linearized, CFD-perturbation method developed herein is based on an unstructured finite volume method and is fully 3D. This allows for the analysis of complex seal geometries that were previously inaccessible. It is based on the Reynolds-averaged Navier-Stokes (RANS) equations and uses the Shear Stress Transport (SST) model to account for turbulent effects. This time-linearized RANS solver, being unstructured, fully 3D, and turbulent, can be used to predict rotordynamic coefficients for complex seal geometries that were inaccessible to previous CFD-perturbation and bulk flow methods.

Numerical simulation of annular seals typically produces good predictions for the seal rotordynamic coefficients and reasonable predictions for leakage. For flow conditions where the friction factor decreases with increasing Reynolds numbers, the numerical models work well, and the seal can be expected to produce a positive static direct stiffness through the Lomakin effect. The Lomakin effect, however, produces negative stiffness if the friction factor increases with increasing Reynolds number. There are reported cases investigated by Childs *et al.* [7, 8, 9] when an annular seal test rig became unstable in the transition regime where the friction factor increased with increasing Reynolds numbers because of negative direct stiffness.

Highly turbulent flow normally exists in annular seals with Reynolds numbers in the range of 20,000, well above the transition regime. Although the flow was already turbulent, in almost half of the tests at elevated Reynolds numbers, the friction factor experienced a sharp increase while increasing Reynolds numbers [10]. Unsteady pressure measurements showed that flow instabilities in the roughness patterns interferes with the through flow and causes this apparent increase in friction factor.

The flow/acoustics interaction in these seals seems to be the cause of the fluid instability that causes the abrupt increase of the friction factor. This phenomenon is similar to the acoustic instability that occurs in aircraft at certain Mach numbers when the bomb bay door is open. The sound radiating from the cavity is associated with shear layer instability over cavity length and with vortex roll-up in the shear layer. The vortex-acoustic interaction selectively amplifies certain sound tones.

In these cases analysis of the rotordynamic stability still requires a better understanding of the features of the flow phenomenon. Without a proper a rigorous analysis of the rotordynamic stability is impossible. Following the insights of experimental research a deeper study of the culprit cavity flow/acoustic interaction at conditions found in gas seals is needed. In short, we need to know what the cavity flow is doing in the gas seals.

The fluid instabilities in the seals are produced when the flow between the rotor shaft and stator passes over roughness patterns of the seals. The roughness act as cavities trapping pockets of recirculating fluid. The grazing flow over the opening of the cavities generates a flow-acoustic feed-back loop within each open cavity which in turn causes acoustic waves to emanate from the cavities.

The phenomenon of fluid instabilities caused by grazing flow past open cavities was first studied by Roshko [11] and Krishnamurty [12]. Formulae to predict the lock-in frequency of the instability were proposed by Rossiter [13] and Tam and Block [14]. Further modeling and experimental investigations were reported by several researchers [15, 16, 17, 18]. More recently computational approaches have been used to study flow past open cavities [19, 20, 21, 22]. Grace [23] has compiled an overview of several of these studies. Reduced order approaches, such as proper orthogonal decomposition, have been employed to extract the salient features of the flow [24, 25].

The majority of the past studies of cavity flows were concerned with open wheel wells and weapons bays on aircraft. They investigated single, two-dimensional cavities on an infinite wall with no opposing wall. Flow past annular gas seals, however, experiences several hundred cylindrical cavities with an opposing wall very near the opening of the cavities. Dougherty *et al.* [26] investigated flows in cavities with an opposing wall. However, the clearance between the cavity and wall was much larger than what would be seen in seals.

Rotordynamics is an area of the study of vibration and controls which concerns itself with the phenomena demonstrated by flexible rotors. Specifically, rotordynamics is focused on linear resonance phenomena, linear instabilities, parametric instabilities, and forced steady-state and transient nonlinear response. The rotordynamic system associated with gas seals most commonly takes the following form,

$$-\begin{pmatrix} F_x \\ F_y \end{pmatrix} = \begin{bmatrix} K & k \\ -k & K \end{bmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \begin{bmatrix} C & c \\ -c & C \end{bmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} \tag{1.1}$$

where X and Y are the rotor's position and F_x and F_y are forces on the rotor in the radial x and y directions, respectively. K and k are direct and cross coupled stiffness coefficients, and C and c are direct and cross coupled damping coefficients. Once the rotordynamic coefficients K, k, C and c are found, the complete dynamics of the system is known. With Eq. (1.1), all that is required to analyze the rotordynamics for a particular gas seal geometry at specified flow conditions, is to determine the values for the rotordynamic coefficients.

Fluid instabilities generated by the grazing flow past an open cavity occur due to the interaction of shear layer oscillations within the cavity, vortices within the shear layer and acoustic waves radiating from the cavity. As shown in Fig. 1.1, a boundary layer forms along the wall upstream of the cavity and separates from the wall as it reaches the leading edge of the cavity forming a shear layer across the top of the cavity. The faster grazing flow in the channel passing over the slower recirculation in the cavity causes Kelvin-Helmholtz oscillations in the shear layer. These oscillations cause vortices to be shed from the leading edge of the cavity and also propagate them along the shear layer. The vortices propagate downstream and impinge on the trailing edge. The interaction between the vortices and the shear layer warps the shear layer causing its reattachment point to momentarily move down the cavity wall below the lip. The shear layer stagnates just below the trailing edge causing a brief period of higher pressure. As the flow accelerates past the trailing edge of the cavity, an area of low pressure is momentarily formed along the wall just downstream of the cavity edge. As a result, an acoustic dipole is generated at the trailing edge, which radiates acoustic waves in all directions. The acoustic waves that propagate upstream excite the shear layer at the leading edge of the cavity, which in turn causes the shedding of additional vortices. This vortex-acoustic interaction forms a feed-back loop which selectively amplifies a dominant frequency.

Using this understanding of the feed-back loop mechanism described above, Rossiter [13]

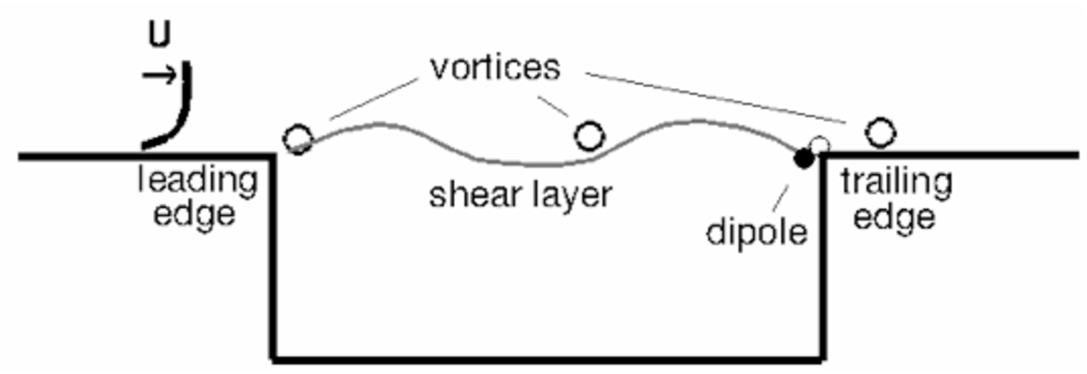


Figure 1.1: Features of cavity flow.

derived a semi-empirical equation that predicts the dominant frequency, f

$$f = \frac{U}{L} \cdot \frac{m - \gamma}{M\frac{c_0}{c_t} + 1/\kappa} \tag{1.2}$$

where U is the freestream flow velocity, L is the length of the cavity, M is the Mach number, c_o and c_t are the speed of sound outside and inside the cavity respectively, γ and κ are empirical terms, and m is the mode of the oscillation. Using experimental measurements, Rossiter determined that the empirical terms are $\gamma = 0.25$ and $\kappa = 0.66$. Subsequent studies have reported κ values as low as 0.57 while the value for γ has remained consistent [27].

It should be noted that Rossiter, as well the majority of researchers utilizing Rossiter's formula, studied cavities open to the freestream with no influence of an opposing wall. Rossiter's formula and values for the empirical constants were found using these flow conditions. For flow in gas seals the cavities are located within a channel where the opposing wall is situated in the proximity of the opening of the cavities. Consequently, Rossiter's formula must be modified to account for the presence of the opposing wall.

1.2 Objective and Scope

The objective of this research was to further the state of the art in analysis of the aerodynamic and rotordynamic stability of compressor and turbine annular gas seals. To achieve the goals, a fully 3D, Reynolds-averaged Navier-Stokes (RANS), unstructured, finite volume solver was developed. A companion fully linearized version of the solver was also developed and used as part of a CFD-perturbation solver based on the time-linearized approximation. The full-order RANS cose is also used to provide the mean flow conditions about which the linearization of the time-linearized code is performed. Together, the full-order and time-linearized codes are used to predict rotordynamic stability coefficients for gas seals.

For seal cases where cavity flow is present, a deeper understanding of the flow physics is necessary. An existing analytical formula, used to predict features of cavity flow, is modified such that it may be applied to gas seals. The full-order RANS code, validated with experimental results, provides insight for the modification.

1.3 Original Contributions of this Dissertation

The time-linearized CFD solver presented in this dissertation offers an improvement over existing linearized CFD solvers developed for turbomachinery. Previous linearized solvers required structured grids and axisymmetric domains, limiting the complexity of the geometries of the seals that could be analyzed. Updating these codes would require major modifications and is impractical. A full-order, in-house CFD code operated on fully 3D and unstructured grids and was already well suited for complex seal geometries. This dissertation develops a linearized version of the full-order, in-house code, retaining its unstructured and fully 3D features. Boundary conditions are also developed for the linearized code allowing it to take advantage of the geometric symmetries that were required by previous codes without necessarily being limited to them. The linearization procedure is also general enough to be used for the many various features of the full-order code that were developed to make it a powerful CFD tool. As the in-house code continues to be developed and new features are included, the same linearization procedure can be used to keep the linearized code up to date. The full-order, in-house RANS solver and the time-linearized companion code combine to become a powerful CFD-perturbation solver accessible to all complexities of seal geometries.

This dissertation also presents an analytical formula that describes features of cavity flow as it pertains to annular gas seals in order to progress the fundamental understanding of the flow physics of roughened seals. An existing semi-empirical analytical formula, developed to describe the cavity flow of aircraft bomb bays, is modified using the full-order, in-house RANS solver. The numeric model is validated against experimental measurements and used to adjust empirical parameters of the formula to match cavity flow conditions unique to annular seals. The unique conditions taken into consideration include the presence of an opposing wall very near the opening of the cavity and the interactions of multiple cavities. The modified analytical formula is able to predict features of cavity flows found in annular gas seals better than existing formulae.

1.4 Outline of Dissertation

The following dissertation describes the numerical methods used to develop the full order code and the time linearized code. It also discusses simulations run to study cavity flow for conditions found in gas seals and how results from those simulations were used to modify an analytical formula. The report also discusses results generated by the time-linearized code for simple validation simulations and the prediction of rotordynamic coefficients for gas seals.

Chapter 2 presents the Navier-Stokes equations, the Reynolds-averaged Navier-Stokes equations and the turbulence model used in the present work. Chapter 3 discusses numerical methods used by the full-order and time-linearized solver and is split into four main sections. The first section describes the spatial discretization of the Reynolds-averaged Navier-Stokes and the turbulence model. The temporal integration methods are discussed the second section. Details of the time-linearization assumptions are discussed in the third section, as well as details on how the assumptions are implemented. The fourth and final section of Chapter 3 covers boundary conditions. The first part of Chapter 4 presents the results of the cavity flow simulations and their applications to gas seals. The second part presents results generated by the CFD-perturbation code and predictions of gas seal rotordynamic coefficients as compared to established bulk flow methods and experimental measurements. Finally, a discussion of the results and the conclusions of the report are presented in Chapter 8 along with suggested directions for future research.

2. PHYSICAL MODEL

This chapter introduces the Reynolds-Averaged Navier-Stokes equations, describing the dynamics of the gas within the annular seals. Methods for solving the equations are presented in later chapters.

2.1 General Conservation Law

A conservation law can be formulated for any conserved fluid property. The general formulation presented here provides a framework that every particular conserved property follows. Following sections will apply this general from to the conservations of mass, momentum and energy.

Consider an arbitrary finite region Ω bounded by a closed surface $\partial\Omega$. The surface is composed of infinitesimal surface elements, dS, along with the associated unit normals, $\hat{n}$, conventionally defined pointing out of the region Ω. For a given conserved quantity ϕ, defined at all locations of the domain, a conservation law hold that variation of ϕ in time within region Ω,

$$\frac{\partial}{\partial t} \int_{\Omega} \phi \, d\Omega \tag{2.1}$$

is equal to the sum of the net flux of ϕ into Ω and any source of ϕ. The flux is composed of two types, convective and diffusive. The convective flux is defined by the amount of ϕ being convected into Ω due to the motion of fluid. It is formulated as

$$-\oint_{\partial\Omega} \phi \left(\vec{v} \cdot \hat{n}\right) dS \tag{2.2}$$

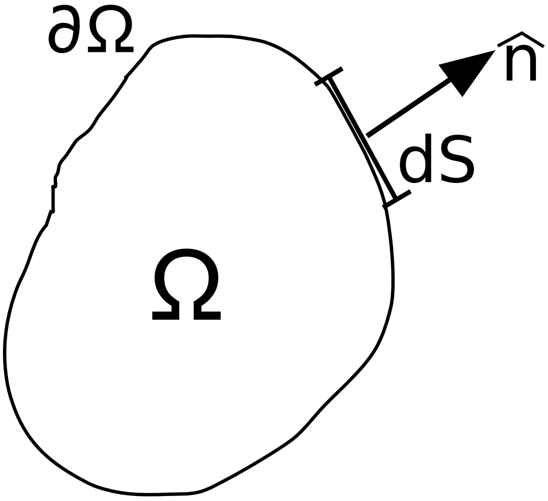


Figure 2.1: Definition of the geometric terms of an arbitrary control region.

where $\vec{v}$ is the velocity of the fluid.

Diffusive flux acts as a smoothing mechanism. It describes a quantity of ϕ entering Ω due to a diffusion of ϕ and is proportional to the gradient of ϕ. The diffusive flux is formulated using a generalized Fick's law as

$$\oint_{\partial\Omega} \kappa\rho \left(\nabla(U/\rho) \cdot \hat{n}\right) dS \tag{2.3}$$

where κ is the diffusivity coefficient and ρ is the fluid density. The derivation of the diffusive flux term is formulated for a fluid at rest.

Also considered are contributions due to body sources, $\vec{G}_v$, and surface sources, $\mathbf{G}_s$.

$$\int_{\Omega} \vec{G}_v \, d\Omega + \oint_{\partial\Omega} \left(\mathbf{G}_s \cdot \hat{n}\right) dS \tag{2.4}$$

Equating the time variation of ϕ within Ω with the sum of the convective and diffusive fluxes and body and surface sources gives the following general conservation law formulation for an arbitrary fluid property ϕ,

$$\frac{\partial}{\partial t}\int\limits_{\Omega} \phi\, d\Omega + \oint\limits_{\partial\Omega} \left[\phi(\vec{v}\cdot\hat{n}) - \kappa\rho(\nabla(U/\rho)\cdot\hat{n})\right] dS = \int\limits_{\Omega} \vec{G}_v\, d\Omega + \oint\limits_{\partial\Omega} (\mathbf{G}_s\cdot\hat{n})dS. \tag{2.5}$$

It is typical to write the flux terms on the same side as the time variation term.

With the formulation for a general conservation law, the following discussions on the conservations of mass, momentum and energy amount to deriving the convective and diffusive fluxes and any source terms as they apply to each specific fluid property.

2.2 Conservation of Mass (Continuity Equation)

The simplest of the three conservation laws described in this chapter is the conservation of mass. The conserved quantity for this case is density, ρ. Following the general formulation above, the time variation of total mass within region Ω is

$$\frac{\partial}{\partial t}\int\limits_{\Omega} \rho\, d\Omega \tag{2.6}$$

and the convective flux across the surface is

$$-\oint\limits_{\partial\Omega} \rho(\vec{v}\cdot\hat{n})dS. \tag{2.7}$$

For the conservation of mass there is no diffusive flux and, for cases within the scope of this paper, no body or surface sources.

With all terms accounted for the formulation for the conservation of mass is

$$\frac{\partial}{\partial t}\int\limits_{\Omega} \rho\, d\Omega + \oint\limits_{\partial\Omega} \rho(\vec{v}\cdot\hat{n})dS = 0. \tag{2.8}$$

2.3 Conservation of Momentum (Newton's 2nd Law)

The conserved quantity here, momentum, is a vector quantity defined as the product of density and velocity, $\vec{m} = \rho\vec{v}$.

The convective flux for momentum, $\rho\vec{v}$, is given by

$$-\oint_{\partial\Omega} \rho\vec{v}\,(\vec{v}\cdot\hat{n})dS. \tag{2.9}$$

No diffusive flux exists for the conservation of momentum.

Body sources can occur under certain conditions due to gravity, buoyancy, Coriolis and centrifugal effects. Most cases within the scope of this paper can neglect body sources so this term is assumed to be zero.

Surface source terms arise for the conservation of momentum due to forces on the finite region. The forces considered consist of isentropic pressure and viscous stress due to friction. With these two forces considered, the surface source term can be written as

$$\mathbf{G}_s = -p\mathbf{I} + \tau \tag{2.10}$$

where p is pressure, $\mathbf{I}$ is the identity tensor and τ is the viscous stress tensor. The surface source term is written as a tensor, as opposed to a vector as seen earlier, to incorporate the three components of momentum. If we wish, the surface source tensor may be separated into x, y, and z components,

$$\mathbf{G}_s = \begin{bmatrix} \vec{G}^T_{s,x} \\ \vec{G}^T_{s,y} \\ \vec{G}^T_{s,y} \end{bmatrix} \tag{2.11}$$

and applied to the x, y, and z components of momentum separately.

The viscous stress introduced above describes the force felt by the fluid within Ω due to friction between itself and neighboring fluids or surfaces. It is described by the tensor, τ, which in Cartesian coordinates, is written

$$\tau = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}. \tag{2.12}$$

By convention, τ_{ij} refers to the component of stress felt on a face of the defined finite region perpendicular to the i-axis and felt in the j-axis direction. This means, for $i = j$, τ_{ij} is felt as a normal force and for $i \neq j$, as a shear force.

For a Newtonian fluid the viscous stresses are proportional to the velocity gradient. The components of the viscous stress tensor, τ, is defined as

$$\begin{aligned}
\tau_{xx} &= \tfrac{2}{3}\mu\left(2\tfrac{\partial u}{\partial x} - \tfrac{\partial v}{\partial y} - \tfrac{\partial w}{\partial z}\right) \\
\tau_{yy} &= \tfrac{2}{3}\mu\left(2\tfrac{\partial v}{\partial y} - \tfrac{\partial w}{\partial z} - \tfrac{\partial u}{\partial x}\right) \\
\tau_{zz} &= \tfrac{2}{3}\mu\left(2\tfrac{\partial w}{\partial z} - \tfrac{\partial u}{\partial x} - \tfrac{\partial v}{\partial y}\right) \\
\tau_{xy} = \tau_{yx} &= \mu\left(\tfrac{\partial u}{\partial y} + \tfrac{\partial v}{\partial x}\right) \\
\tau_{xz} = \tau_{zx} &= \mu\left(\tfrac{\partial u}{\partial z} + \tfrac{\partial w}{\partial x}\right) \\
\tau_{yz} = \tau_{zy} &= \mu\left(\tfrac{\partial v}{\partial z} + \tfrac{\partial w}{\partial y}\right)
\end{aligned} \tag{2.13}$$

where μ is the dynamic viscosity coefficient where $\mu = 1.82 \times 10^{-5} \frac{kg}{m^2\,s}$. Note that τ is a symmetric matrix so $\tau_{ij} = \tau_{ji}$.

2.4 Conservation of Energy (1st Law of Thermodynamics)

The conserved quantity for the conservation of energy is the total energy per unit volume and is written as

$$\rho E = \rho \left(e + \frac{|v|^2}{2} \right) \tag{2.14}$$

where e and $|v|^2/2$ are the internal and kinetic energy per unit mass, respectively. The internal energy, e, for a calorically perfect gas is expressed as

$$e = c_v T \tag{2.15}$$

where c_v is the specific heat at constant volume and T is the temperature of the fluid. The convective flux, then, is written

$$-\oint_{\partial\Omega} \rho E \, (\vec{v} \cdot \hat{n}) dS. \tag{2.16}$$

Unlike the conservations of mass and momentum, the conservation of energy has a diffusive flux. As stated earlier, this term is defined for a fluid at rest so it is proportional to the gradient of only the internal energy without influence from the kinetic energy. The contribution, then, to the variation of energy in a finite region due to diffusive flux can be written as

$$\oint_{\partial\Omega} \gamma \rho \kappa \, (\nabla e \cdot \hat{n}) dS \tag{2.17}$$

where $\gamma = c_p/c_v$ is the ratio of specific heat ($\gamma = 1.4$ for air), and κ is the thermal diffusivity coefficient.

A more common form for the previous term may be chosen if, instead of the gradient of internal energy, ∇e, the gradient of total temperature, ∇T, is used. It can then be written as Fourier's law of heat transfer,

$$\oint\limits_{\partial\Omega} k\,(\nabla T \cdot \hat{n})dS \tag{2.18}$$

where k is thermal conductivity and T is total temperature.

For the cases considered in this paper, no body sources, $\vec{G}_v$ are included for the conservation of energy. Generally, body sources can be included due to radiation, chemical reactions along with the body forces outlined in the previous section. None of these effects, however, are included here.

The last term considered is the surface source, $\mathbf{G}_s$. It is written as,

$$\vec{G}_s = -p\vec{v} + \tau \cdot \vec{v}. \tag{2.19}$$

Combining all terms together, the conservation of energy is formulated,

$$\frac{\partial}{\partial t}\int\limits_{\Omega} \rho E\, d\Omega + \oint\limits_{\partial\Omega} \rho E(\vec{v}\cdot\hat{n})dS = \oint\limits_{\partial\Omega} k\,(\nabla T \cdot \hat{n}) - p(\vec{v}\cdot\hat{n}) + (\tau\cdot\vec{v})\hat{n}\, dS \tag{2.20}$$

A more common form of the equation can be written with the total enthalpy, H, defined as

$$H = \left(h + \frac{|\vec{v}|^2}{2}\right) = E + \frac{p}{\rho} \tag{2.21}$$

and allows for the consolidation of the convective flux and pressure surface source term. The internal enthalpy, h, is defined for a calorically perfet gas is expressed as

$$h = c_p T. \tag{2.22}$$

where c_p is the specific heat at constant pressure.

Using the total enthalpy, H, the conservation of energy can be written as

$$\frac{\partial}{\partial t}\int\limits_{\Omega} \rho E \, d\Omega + \oint\limits_{\partial\Omega} \rho H(\vec{v}\cdot\hat{n})dS = \oint\limits_{\partial\Omega} k\left(\nabla T\cdot\hat{n}\right) + (\tau\cdot\vec{v})\hat{n}dS \tag{2.23}$$

These equations, conservation of mass, momentum and energy, are collectively referred to as the Navier-Stokes equations.

2.5 Integral Versus Differential Forms

The Navier-Stokes equations have been derived in the previous sections in their integral, or weak, form. They may also be presented in their differential, or strong form. This section compares the two forms for the Navier-Stokes equations.

The integral form of the equations may be transformed into the differential form by shrinking the control volume Ω to an infinitesimal size and applying the divergence theorem. The divergence theorem is written,

$$\oint\limits_{\partial\Omega} \left((\vec{\bullet})\cdot\hat{n}\right) dS = \int\limits_{\Omega} \nabla\cdot(\vec{\bullet})\, d\Omega. \tag{2.24}$$

Starting from the final form of the conservation of mass equation, Eq.(2.8), and assuming the control volume is fixed in space, gives

$$\int\limits_{\Omega} \frac{\partial\rho}{\partial t} d\Omega + \oint\limits_{\partial\Omega} \rho(\vec{v}\cdot\hat{n})dS = 0 \tag{2.25}$$

Applying the divergence theorem to the surface integral leads to,

$$\int\limits_{\Omega} \frac{\partial\rho}{\partial t} d\Omega + \int\limits_{\Omega} \nabla\cdot(\rho\vec{v})\, d\Omega = 0 \tag{2.26}$$

or rearranged,

$$\int_{\Omega} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) \right] d\Omega = 0. \tag{2.27}$$

Since the control volume, Ω, is arbitrarily defined, to ensure Eq.(2.27) is satisfied for any general volume, the bracketed term must always equal zero. This leads to,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{2.28}$$

which is the differential form of the conservation of mass equation. Similarly, a differential form can be found for the conservation of momentum and conservation of energy equations.

The integral form of the equations are derived directly from the conservation assumptions. Because of this, the integral form holds even across solution discontinuities, such as shock waves or shear layers, satisfying the Rankine-Hugoniot relations. The differential form requires modifications to remain accurate.

2.6 Turbulence Models

For many engineering applications the flow is in the turbulent regime. Turbulence is a flow characterized by small scale random fluctuations. For most practical applications, capturing both the large scale fluid features important in engineering applications and the small scale fluctuations of turbulent flows requires computing power which is not yet available. Direct Numerical Simulation (DNS) is a numeric method that uses Navier-Stokes simulations to directly capture the turbulent fluctuations. Currently, it is only possible to simulate very simple flows at very low Reynolds' numbers using DNS. It is used for such things as understanding turbulent fluid structures, transition from laminar (non-turbulent) to turbulent flow and the development of turbulence models.

The first level of turbulence model is the Large-Eddy Simulation (LES) method. LES is based on the observation that the small scales of turbulent flow possess common characteristics that can be approximated by a carefully chosen model while larger scales of the flow work to transport the turbulent energy around. Therefore, only the large scales of the Navier-Stokes equations are captured while the small scale turbulent effects are modeled. This allows LES to be less computationally expensive than DNS. However, given current computational capabilities, LES is still very demanding. An overview of LES applications can be found in [28].

The next level of approximation is called the Reynolds-Averaged Navier-Stokes equations (RANS). This is the method of turbulence modeling employed here. The remainder of this section will be used to explain RANS.

The methodology of RANS is based on the decomposition of the flow variables into a mean and a fluctuating part. Applied to a general variable, q, Reynolds averaging appears as

$$q = \overline{q} + q' \tag{2.29}$$

where $\overline{q}$ is the mean value of q and q' represents the fluctuations. The mean value will be defined here as the average value of q over a large time.

$$\overline{q} = \lim_{T\to\infty} \frac{1}{T} \int\limits_{t}^{t+T} q\, dt \tag{2.30}$$

The large time, $T \to \infty$, refers, in a practical sense, to a time period much longer that the time scale of the fluctuations.

When working with the Navier-Stokes equations it is common to encounter terms of the form ρq where a given variable is multiplied by density, ρ. For these terms, Favre averaging gives simpler results than Reynolds averaging. Favre averaging can be thought of as a density weighted version of Reynolds averaging and can be written

$$q = \widehat{q} + q'' \tag{2.31}$$

where

$$\widehat{q} = \frac{1}{\overline{\rho}} \lim_{T \to \infty} \frac{1}{T} \int\limits_{t}^{t+T} \rho q \, dt. \tag{2.32}$$

For both Reynolds and Favre averaging the average of the fluctuating part for a single quantity is zero.

$$\begin{aligned} &\overline{q'} = 0 \\ &\widehat{q''} = \overline{\rho q''} = 0 \end{aligned} \tag{2.33}$$

However, the average of the product of two fluctuating quantities is generally not zero.

$$\begin{aligned} &\overline{q'q'} \neq 0 \quad \overline{q'r'} \neq 0 \\ &\widehat{q''q''} \neq 0 \quad \widehat{q''r''} \neq 0. \end{aligned} \tag{2.34}$$

Here r is a general variable different from q.

This section will mix Reynolds and Favre averaging. It is therefore useful to point out the following relationships,

$$\begin{aligned} &\widehat{\rho q} = \overline{\rho} \widehat{q} \\ &\overline{\rho q''} = 0 \\ &\overline{q''} \neq 0 \end{aligned} \tag{2.35}$$

Specifically, components of velocity, u, v, w, total enthalpy, H, and total energy,

E, appear as products multipied by density, ρ, and are Favre averaged. The other variables, density and pressure, p, are Reynolds' averaged. The selective use of Favre and Reynolds' averaging allows the resultant equations to avoid density fluctuation terms. The specified averaging applied to the differential form of the Navier-Stokes equations yields,

$$\begin{aligned}
\frac{\partial\overline{\rho}}{\partial t} + \frac{\partial}{\partial x_i}\left(\overline{\rho}\widehat{v}_i\right) &= 0 \\
\frac{\partial}{\partial t}\left(\overline{\rho}\widehat{v}_i\right) + \frac{\partial}{\partial x_i}\left(\overline{\rho}\widehat{v}_j\widehat{v}_i\right) &= -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\widehat{\tau_{ij}} - \overline{\rho}\widehat{v_i''v_j''}\right) \\
\frac{\partial}{\partial t}\left(\overline{\rho}\widehat{E}\right) + \frac{\partial}{\partial x_j}\left(\overline{\rho}\widehat{v}_j\widehat{H}\right) &= \frac{\partial}{\partial x_j}\left(k\frac{\partial\widehat{T}}{\partial x_j} - \overline{\rho}\widehat{v_j''h''} + \widehat{\tau_{ij}v_i''} - \overline{\rho}\widehat{v_j''K}\right) \\
&\quad +\frac{\partial}{\partial x_j}\left[\widehat{v}_i\left(\widehat{\tau_{ij}} - \overline{\rho}\widehat{v_i''v_j''}\right)\right]
\end{aligned} \tag{2.36}$$

where repeated indices imply summation. The viscous stress tensor, τ_{ij}, in the conservation of momentum and energy equations is extended by the Favre-averaged Reynolds stress tensor,

$$\tau_{ij}^F = -\overline{\rho}\widehat{v_i''v_j''}. \tag{2.37}$$

In Eq.(2.36), the Favre-averaged turbulent kinetic energy term, $\widehat{K}$, has been added. It is defined as

$$\widehat{K} = \frac{1}{2}\widehat{v_i''v_j''}. \tag{2.38}$$

The Favre averaged total energy, $\widehat{E}$, and total enthalpy, $\widehat{H}$, use the kinetic energy term as a result of the averaging.

$$\widehat{E} = \widehat{e} + \frac{1}{2}\widehat{v}_i\widehat{v}_i + \frac{1}{2}\widehat{v_i''v_i''} = \tilde{E} + \widehat{K} \tag{2.39}$$

and

$$\widehat{H} = \widehat{h} + \frac{1}{2}\widehat{v}_i\widehat{v}_i + \frac{1}{2}\widehat{v_i'' v_i''} = \tilde{H} + \widehat{K}. \tag{2.40}$$

The individual parts of the RANS equations (2.36) can be physically understood using the following interpretations [29],

$$\begin{array}{rcl}
\frac{\partial}{\partial x_j}\left(k\frac{\partial \widehat{T}}{\partial x_j}\right) & - & \text{molecular diffusion of heat} \\
\frac{\partial}{\partial x_j}\left(\overline{\rho}\widehat{v_j'' h''}\right) & - & \text{turbulent transport of heat} \\
\frac{\partial}{\partial x_j}\left(\widehat{\tau_{ij} v_i''}\right) & - & \text{molecular diffusion of } \widehat{K} \\
\frac{\partial}{\partial x_j}\left(\overline{\rho}\widehat{v_j'' K}\right) & - & \text{turbulent transport of } \widehat{K} \\
\frac{\partial}{\partial x_j}\left(\widehat{v}_i\widehat{\tau_{ij}}\right) & - & \text{work done by the molecular stresses} \\
\frac{\partial}{\partial x_j}\left(\widehat{v}_i\tau_{ij}^F\right) & - & \text{work done by the Favre-averaged Reynolds stresses}
\end{array} . \tag{2.41}$$

To close the RANS equations, six components of the Favre-averaged Reynolds stress tensor, τ_{ij}^F, and three components of the turbulent heat flux vector must be supplied.

Using the Boussinesq approximation, the Reynolds stress can be linked to the mean flow taking a form similar to the viscous stress tensor from Eq.(2.13). It is assumed that the Reynolds stress is related to the gradient of the mean velocity by

$$-\overline{\rho v_i'' v_j''} = \mu_T\left(\frac{\partial \widehat{v}_i}{\partial x_j} + \frac{\partial \widehat{v}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \widehat{v}_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\overline{\rho}\widehat{K}\delta_{ij} \tag{2.42}$$

where μ_T is the turbulent eddy viscosity as determined by an accompanying turbulence model. The Reynolds heat flux is related to the gradient of the mean temperature by

$$\overline{\rho h'' v_j''} = -\frac{c_p \mu_T}{Pr_T} \frac{\partial \overline{T}}{\partial x_j} \tag{2.43}$$

where Pr_T is the turbulent Prandtl number. For air at standard conditions, $Pr_T = 0.9$.

Eq.(2.42) and Eq.(2.43) are substituted into Eq.(2.36) giving,

$$\begin{aligned}
\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial \overline{\rho} \widehat{v}_j}{\partial x_j} &= 0 \\
\frac{\partial \overline{\rho} \widehat{v}_i}{\partial t} + \frac{\partial \overline{\rho} \widehat{v}_i \widehat{v}_j}{\partial x_j} &= -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \overline{\tau}_{ij}}{\partial x_j} \\
\frac{\partial \overline{\rho} \tilde{E}}{\partial t} + \frac{\partial \overline{\rho} \tilde{H} \widehat{v}_j}{\partial x_j} &= \frac{\partial}{\partial x_j} \left(\widehat{v}_i \overline{\tau}_{ij} - \overline{q}_j \right)
\end{aligned} \tag{2.44}$$

where

$$\overline{\tau}_{ij} = (\mu + \mu_T) \left(\frac{\partial \widehat{v}_i}{\partial x_j} + \frac{\partial \widehat{v}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \widehat{v}_k}{\partial x_k} \delta_{ij} \right), \tag{2.45}$$

and

$$\overline{q}_j = c_p \left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \frac{\partial \overline{T}}{\partial x_j}. \tag{2.46}$$

The effect of the turbulence is modeled with the two-equation eddy-viscosity Shear Stress Transport (SST) model proposed by Menter [30].

2.7 Formulation for a Perfect Gas

The working gas of the turbomachinery is assumed to behave like a calorically perfect gas. This section discusses the relationships between flow quantities that arise due to this assumption. The equation of state for perfect gases takes the form [31],

$$p = \rho R T \tag{2.47}$$

where R represents the specific gas constant and equals $287.16\frac{m^2}{s^2}K$. It is defined as the difference between the specific heat coefficients for a constant pressure, c_p, and a constant volume, c_v,

$$R = c_p - c_v. \tag{2.48}$$

Recalling the definitions for enthalpy, h, and total enthalpy, H, from Eq.(2.22) and Eq.(2.21), respectively, and using the definition for the specific gas constant, R, along with the following definition,

$$\gamma = \frac{c_p}{c_v} \tag{2.49}$$

a formulation for pressure, p, can be obtained using only conserved properties. This formulation has the form,

$$p = (\gamma - 1)\rho \left[E - \frac{u^2 + v^2 + w^2}{2} \right]. \tag{2.50}$$

The equation of state, Eq.(2.47), can then be used to define the temperature, T. The coefficient of the dynamic viscosity, μ, is strongly dependent on temperature and only weakly dependent on pressure. To determine μ, the Sutherland formula is used. It is written as,

$$\mu = \frac{1.45\, T^{3/2}}{T + 110} \cdot 10^{-6} \tag{2.51}$$

where the temperature, T, is in degrees Kelvin. The temperature dependence of the thermal conductivity coefficient, k, is very near that of μ. The following relationship is used for air,

$$k = c_p \frac{\mu}{Pr} \tag{2.52}$$

where the Prandtl number, Pr, is assumed constant throughout the flow field. For air, the Prandtl number has the value Pr=0.72.

3. NUMERICAL METHOD

3.1 Spatial Discretization

This chapter discusses the derivation of the discretized form of the Navier-Stokes equations. The governing equations are discretized so they may be solved using numerical methods. This chapter outlines the discretization of the domain, the projection of the governing equations onto the domain and the upwinding, gradient calculation, and reconstruction methods required for a stable and accurate solution.

3.1.1 Navier-Stokes Equations

The Navier-Stokes equations were derived in the previous chapter. This section will present those equations in a useful form that eases the mathematical manipulations required for the implementation of numerical methods presented later in this chapter. Generally, this is done by storing the flow variables as a single vector and operating on the vector as a whole.

We begin by gathering the Navier-Stokes equations introduced in the previous chapter. The conserved quantities associated with each equation are stored in a single vector called the state vector, $\vec{Q}$. It is written as

$$\vec{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} . \tag{3.1}$$

We choose to present the components of $\vec{Q}$ defined using the primitive variables so the following discussion can be as conceptually simple to follow as possible. It could just as easily be written in terms of the conserved variables, or any other complete set of flow variables, and retain the same rigor. The primitive variables provide an intuitive comfort other flow variables do not.

Storing the conserved quantities as a single vector allows the Navier-Stokes equations to collectively be considered as a system of equations. The system of equations, then, take the form

$$\frac{\partial}{\partial t}\int\limits_{\Omega}\vec{Q}\,d\Omega + \oint\limits_{\partial\Omega}(\vec{F}_c - \vec{F}_v)dS = \int\limits_{\Omega}\vec{G}\,d\Omega \tag{3.2}$$

The remainder of this subsection will be used to define the terms of Eq.(3.2).

The vector of convective fluxes, $\vec{F}_c$, is the collection of the convective flux terms. For convenience, the source terms containing pressure, p, found in the conservation of momentum and energy equations are also included. The convective flux vector, $\vec{F}_c$, can be written

$$\vec{F}_c = \begin{pmatrix} \rho V \\ \rho uV + n_x p \\ \rho vV + n_y p \\ \rho wV + n_z p \\ \rho EV + Vp \end{pmatrix} = \begin{pmatrix} \rho V \\ \rho uV + n_x p \\ \rho vV + n_y p \\ \rho wV + n_z p \\ \rho HV \end{pmatrix} \tag{3.3}$$

Here, n_x, n_y, and n_z represent the components of the direction normal to the infinitesimal surface element dS. The term V represents the component of velocity normal to dS, called the contravariant velocity. It is defined in three dimensions as

$$V = (\vec{v}\cdot\hat{n}) = n_x u + n_y v + n_z w. \tag{3.4}$$

The vector of viscous fluxes, $\vec{F}_v$, contains the viscous stress terms found in the surface source terms in the previous chapter. Also, the heat diffusion term is included in the conservation of energy equation. The viscous flux vector is written

$$\vec{F}_v = \begin{pmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\Theta_x + n_y\Theta_y + n_z\Theta_z \end{pmatrix} \tag{3.5}$$

where the viscous stress tensor, τ, is defined in Eq.(2.3) and

$$\begin{aligned} \Theta_x &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\tfrac{\partial T}{\partial x} \\ \Theta_y &= u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\tfrac{\partial T}{\partial y} \\ \Theta_z &= u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\tfrac{\partial T}{\partial z} \end{aligned} \tag{3.6}$$

are terms describing work done by the viscous stresses and the heat conduction.

The vector, $\vec{G}$, represents any source terms that haven't already been included in $\vec{F}_c$ or $\vec{F}_v$. This may include the effects of gravity, buoyancy, Coriolis forces, etc. For the cases covered here the source vector, $\vec{G}$, is equal to zero.

3.1.2 Methods for Spatial Discretization

This subsection presents the first preprocessing step before solving the Navier-Stokes equations, discretizing the equations in space. There are many established methodologies for spatial discretization. The three most common are finite difference, finite element and finite volume.

The finite difference method defines the state variables at distinct points throughout the domain. It operates on the differential form of the governing equations and uses Taylor series to approximate the gradient terms. In this way, relationships are defined between the state variables stored at each point.

The finite element method, however, does operate on the integral form of the governing equations. The equations are projected onto a set of discrete spatial elements chosen so they fully cover the domain without overlapping or leaving gaps. Higher order approximations are achieved by using several higher order basis functions on each element.

The finite volumes method, like the finite element method, operates on the integral form of the governing equations. Similar to the finite element method, the equations are projected onto a set of discrete spatial, in this case, control volumes which fully span the domain without overlapping and without leaving gaps. With the finite volumes method a premium is placed on complexity so each control volume can be efficiently handled by the computer allowing for the use of a very large number of volumes. The simplicity of the method also eases the development of the more complicated procedures that are often required when working with the Navier-Stokes equations such as up-winding and preconditioning. The finite volume method is the most common procedure used for computational fluid dynamics and is the method used in this work. The remainder of this subsection will be used to describe it in more detail.

As was mentioned above, the finite volume method operates on a set of volumes that fill the domain. This set is referred to as the grid or mesh. Creating quality meshes can be a very difficult task and has developed into a field of its own. For the development of the finite volume method presented here, it will be assumed a mesh already exists.

We introduce a single control volume located somewhere in the domain. The Navier-Stokes equations consist of two types of terms, the volume integral and the surface integral. They are written in a general form, respectively, as

$$\int_{\Omega} a \, d\Omega \tag{3.7}$$

and

$$\oint_{\partial\Omega} (\vec{b} \cdot \hat{n}) \, dS. \tag{3.8}$$

Assuming that a is constant within the volume gives the following discrete approximation.

$$\int_{\Omega} a \, d\Omega = a_i \Omega_i \tag{3.9}$$

where Ω_i on the right hand side represents the area (volume for 3D domains) of the finite volume.

For the surface integral the assumption is made that $\vec{b}$ is piecewise constant along the boundary of the control volume. This leads to the discrete approximation

$$\oint_{\partial\Omega} (\vec{b} \cdot \hat{n}) \, dS = \sum_{j=1}^{N_i} (\vec{b}_{ij} \cdot \hat{n}_{ij}) S_{ij} \tag{3.10}$$

where each discrete face on the boundary is shared with a neighboring control volume. Here, j represents a neighbor of volume i, and $\hat{n}_{ij}$ and S_{ij} represent the normal direction and length (area in 3D) of the edge connecting control volumes i and j. N_i is the set of neighbors of control volume i.

Substituting the volume (3.9) and surface (3.10) approximations into the Navier-Stokes equations, Eq.(3.2), gives

$$\frac{\partial}{\partial t}(\vec{Q}_i \, \Omega_i) + \sum_{j=1}^{N_i} (\vec{F}_c - \vec{F}_v)_{ij} \, S_{ij} = \Omega_i \, G_i \tag{3.11}$$

Assuming the control volume is fixed and unchanging in time, Ω_i is independent of time, Eq.(3.11) can be rearranged as

$$\frac{\partial \vec{Q}_i}{\partial t}\Omega_i = \vec{R}_i \tag{3.12}$$

where $\vec{R}_i$ is the residual of control volume i and is defined as

$$\vec{R}_i = -\sum_{j=1}^{N}(\vec{F}_c - \vec{F}_v)_{ij}\, S_{ij} - \Omega_i\, G_i \tag{3.13}$$

The relationship in Eq.(3.12) is defined for each control volume of the mesh and, collectively, form a system of ordinary differential equations in time. A solution is achieved by advancing this system forward in time. The procedure used to advance the system is covered in the next chapter and discussions on boundary conditions will be presented in Chapter 6.

3.1.3 Geometric Quantities

Several terms in the discretized governing equations (3.11) are purely geometric. They depend on the definition of the mesh but not on the current state of the flow field. These terms include Ω_i, S_{ij} and n_{ij}. This subsection will describe how these geometric quantities are calculated.

There are generally two methods for discretizing and storing the mesh, structured and unstructured. With structured meshes each node is uniquely defined by indices i, j, k and the corresponding Cartesian coordinates $x_{i,j,k}$, $y_{i,j,k}$, and $z_{i,j,k}$. Neighbors of the nodes can be found by simply adding or subtracting an integer value to and from the node index. For unstructured meshes, no such index system exists. The set of neighboring nodes must be explicitly stored for every node.

Despite the apparent advantages using a structured mesh, generating the ordered index system for complex geometries becomes very difficult [32]. To allow more flexibility in the shape of the domain, unstructured meshes are used. This allows for arbitrarily shaped control volumes and domains that can be fitted more easily to complex geometries. Because of the added flexibility, unstructured meshes are used herein.

When using unstructured meshes, one of two strategies is used to define the control volumes and decide where to store the state variables. The two methods are cell-centered and cell-vertex schemes. Cell-centered schemes define the control volume identical to the cells defined by the mesh. The state variables are stored at the cell centroid. Cell-vertex schemes store the state variables at the nodes (vertices) of the mesh. The control volume can either be defined as the union of all cells adjacent to the node, creating an overlap between neighboring volumes (overlapping) or defined by connecting the cell centroids of the cells connected to the node (centroid dual Fig. 3.1), preventing the volume overlap. Alternatively, the volume may be defined by the edge midpoints and boundary face centroids in addition to the cell centroids (median dual Fig. 3.2).

The computational cost of the numeric method is closely correlated with the number of control volumes or nodes. For a typical mesh composed of tetrahedral cells, the ratio of cells to nodes ranges approximately from 5 to 6. Because cell-vertex schemes have control volumes defined by the vertices, they are computationally less costly than the cell-centered schemes. For this reason, a cell-vertex scheme with median dual meshes are used here.

The boundaries of a 2D median dual control volume runs from the centroid of a cell to the centroid of a neighboring cell, and pass through the midpoint of the edge connecting the two. To extend the approach to 3D, a control volume is defined using the centroids of the cells, the centroids of the faces between the cells, and the

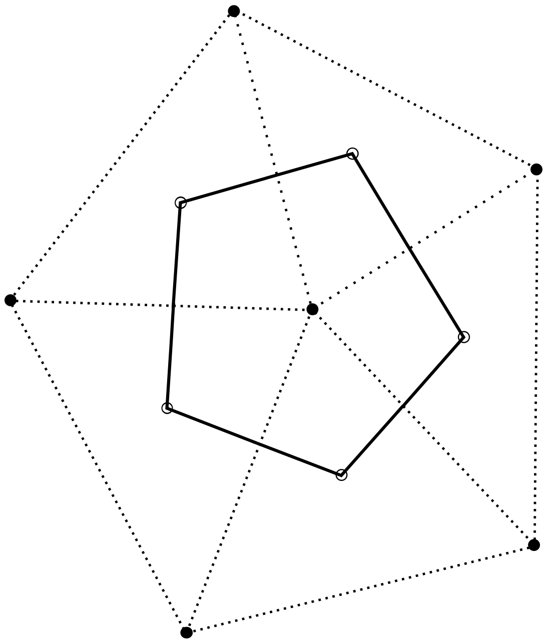


Figure 3.1: Centroid dual cell. Bold lines indicate the control volume.

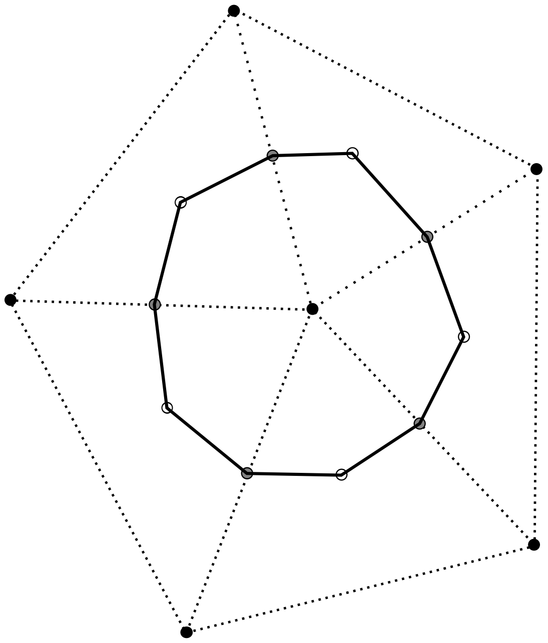


Figure 3.2: Median dual cell. Bold lines indicate the control volume.

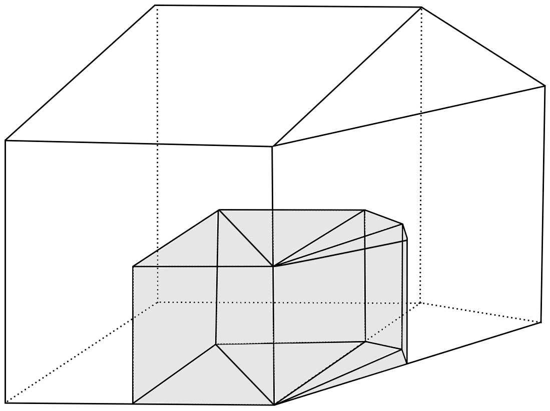


Figure 3.3: 3D dual cell. Shaded region indicates the control volume.

midpoints of the edges adjacent to a given node. Figures 3.2-3.3 show examples of 2D and 3D control volumes generated using the median dual approach. Median dual control volumes can be defined using cells of arbitrary shape. This flexibility makes this method well suited for mixed element meshes [33].

The volume of a control volume and the areas of its boundary faces are computed by summing the contributions from the cells adjacent to the node. The following pseudocode describes the algorithm used to calculate these volumes and areas. The following connectivity data is provided by the mesh; cells to faces, faces to edges, and edges to nodes.

```
$area = 0$          ! Area Vector !
$volume = 0$        ! Control Volume !
do $c = 1, cells$
    do $f = 1, faces(c)$
        do $e = 1, edges(c)$
            $v_1 = vector(me, cc)$        ! Mid-Edge to Cell Centroid !
            $v_2 = vector(me, fc)$        ! Mid-Edge to Face Centroid !
            $tri\_area = cross\_product(v_1, v_2)$
            $area(e) = area(e) + tri\_area$                            (3.14)
            $n_1 = edge\_to\_node(1)$
            $n_2 = edge\_to\_node(2)$
            $volume(n_1) = tri\_area * edge\_length(e)\ /\ 2$
            $volume(n_2) = tri\_area * edge\_length(e)\ /\ 2$
        end do
    end do
end do
```

Cross products are used to calculate the area of triangles defined by a cell centroid, a face centroid, and the midpoint of an edge. Generally, the triangles that make up a face are not coplanar. The areas are therefore stored as spatial vectors where the magnitude of the vector is equal to the area and the direction is normal to the face. The positive direction is defined to be from the first node to the second node as is defined by the mesh connectivity. Unit normals, $\hat{n}$, are defined for each face by normalising the area-vectors. In this way, the geometric terms Ω, S, and $\hat{n}$ are all uniquely and systematically defined by the mesh.

3.1.4 Convective Flux

With the geometric terms defined in the previous subsection, we return to the discretized governing equations, Eq.(3.11). Still remaining to be defined are the convective flux, $\vec{F}_c$, and viscous flux, $\vec{F}_v$. This subsection will address $\vec{F}_c$ while following subsections will address $\vec{F}_v$ and associated terms.

In the convective flux term in Eq.(3.11),

$$\sum_{j=1}^{N_i} \vec{F}_{c,ij}\, S_{ij} \tag{3.15}$$

the vector $\vec{F}_c$ is an edge-based term. Values are stored at the edges of the mesh. To find the edge-based convective fluxes, values are first found at the nodes using the node-based state vector, $\vec{Q}_i$. The nodal fluxes, then, are used to define the edge-based fluxes. The most straight forward method would be to average the two nodal values from either end of an edge. This is called the central scheme and is written as

$$\vec{F}_{c,ij} = \frac{1}{2}(\vec{F}_{c,i} + \vec{F}_{c,j}) \tag{3.16}$$

where $(\bullet)_{ij}$ refers to a term stored at the edge connecting nodes i and j.

Unfortunately, this method destabilizes the solution. To prevent this, one of two general approaches are used to calculate the convective flux. The first starts with the central scheme presented above and adds artificial dissipation term to stabilize the scheme. The most common implementation, JST [34], uses a combination of the 2nd and 4th-order differences to define the dissipation term. This scheme is computationally inexpensive but can numerically smooth discontinuities in the flow field due to the added dissipation.

The second group of schemes takes into account the physical properties of the convective flux to achieve stability. They are able to distinguish between the upstream and downstream influence to ensure that flow information travels in the direction dictated by the physics. Because of this, these schemes are called upwind schemes. They are more computationally expensive than central schemes but are able to capture shock waves and boundary layers more accurately. Because of their accuracy, upwind schemes are used in this work. The remainder of the subsection will be used to describe the implementation of an upwind scheme to discretize the convective flux.

Several versions of upwind schemes are commonly used. They include Flux-Vector splitting schemes such as Van Leer's[35, 36, 37], AUSM[38, 39, 40], and CUSP[41, 42, 43] as well as Flux-Difference splitting schemes such as Roe[44] and Osher[45]. An overview of the different upwind schemes can be found in [28]. The current work uses a flux difference scheme created by Roe which has been shown to have high accuracy in the boundary layer and good resolution of the shocks.

Roe's scheme, as with all flux-difference splitting schemes, attempts to solve the Riemann shock tube problem at each volume boundary for every time step. This approach was first proposed by Godunov and can be expensive to use. Roe negated much of the expense by proposing an approximate solution based on the linearization of the system. Roe's Riemann solver is based on 1-D approximation of the Riemann problem and can be easily implemented by defining a 1-D line connecting two neighboring nodes. Roe's Riemann solver has been shown to be accurate with a relatively low computational cost.

The convective flux discretized using Roe's Riemann scheme is defined as

$$\vec{F}_{c,ij} = \frac{1}{2}(\vec{F}_{c,i} + \vec{F}_{c,j} - |\tilde{A}|\,\Delta\vec{Q}_{ij}) \tag{3.17}$$

where $(\bullet)_i$ and $(\bullet)_j$ refer to terms associated with the nodes on the left and right sides of a given edge, respectively. $(\bullet)_{ij}$ refers to the flux vector associated with the edge ij connecting nodes i and j. $|\tilde{A}|$ is a modified version of the flux Jacobian with respect to the conserved state variables, and $\Delta\vec{Q}_{ij} = \vec{Q}_j - \vec{Q}_i$ is the difference between left and right states.

The modified flux Jacobian, $|\tilde{A}|$, is defined using Roe's density weighted averages of the state variables. These averages are defined as

$$\begin{aligned}
\tilde{\rho} &= \sqrt{\rho_i \rho_j} \\
\tilde{u} &= (u_i\sqrt{\rho_i} + u_j\sqrt{\rho_j}) \,/\, (\sqrt{\rho_i} + \sqrt{\rho_j}) \\
\tilde{v} &= (v_i\sqrt{\rho_i} + v_j\sqrt{\rho_j}) \,/\, (\sqrt{\rho_i} + \sqrt{\rho_j}) \\
\tilde{w} &= (w_i\sqrt{\rho_i} + w_j\sqrt{\rho_j}) \,/\, (\sqrt{\rho_i} + \sqrt{\rho_j}) \\
\tilde{H} &= (H_i\sqrt{\rho_i} + H_j\sqrt{\rho_j}) \,/\, (\sqrt{\rho_i} + \sqrt{\rho_j}) \\
\tilde{V} &= \tilde{u}n_x + \tilde{v}n_y + \tilde{w}n_z \\
\tilde{q}^2 &= \tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2 \\
\tilde{c} &= \sqrt{(\gamma - 1)(\tilde{H} - \tilde{q}^2/2)}.
\end{aligned} \tag{3.18}$$

The dissipative term, $|\tilde{A}|\,\Delta\vec{Q}_{ij}$, can be pre-multiplied to improve computational efficiency. The result is written as,

$$|\tilde{A}|\,\Delta\vec{Q}_{i,j} = |\Delta\vec{F}_1| + |\Delta\vec{F}_{2,3,4}| + |\Delta\vec{F}_5| \tag{3.19}$$

where

$$|\Delta \vec{F}_1| = |\tilde{V} - \tilde{c}| \left(\frac{\Delta p - \tilde{\rho}\tilde{c}\, \Delta V}{2\tilde{c}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} - \tilde{c}\, n_x \\ \tilde{v} - \tilde{c}\, n_y \\ \tilde{w} - \tilde{c}\, n_z \\ \tilde{H} - \tilde{c}\, \tilde{V} \end{bmatrix}, \tag{3.20}$$

$$\begin{aligned} |\Delta \vec{F}_{2,3,4}| &= |\tilde{V}| \left(\Delta\rho - \tfrac{\Delta p}{\tilde{c}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{q}^2/2 \end{bmatrix} \\ &+ |\tilde{V}|\tilde{\rho} \begin{bmatrix} 0 \\ \Delta u - \Delta V\, n_x \\ \Delta v - \Delta V\, n_y \\ \Delta w - \Delta V\, n_z \\ \tilde{u}\, \Delta u + \tilde{v}\, \Delta v + \tilde{w}\, \Delta w - \tilde{V}\, \Delta V \end{bmatrix}, \end{aligned} \tag{3.21}$$

and

$$|\Delta \vec{F}_5| = |\tilde{V} + \tilde{c}| \left(\frac{\Delta p + \tilde{\rho}\tilde{c}\, \Delta V}{2\tilde{c}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} + \tilde{c}\, n_x \\ \tilde{v} + \tilde{c}\, n_y \\ \tilde{w} + \tilde{c}\, n_z \\ \tilde{H} + \tilde{c}\, \tilde{V} \end{bmatrix}. \tag{3.22}$$

As noted in [46, 47, 48, 49], numeric stability issues occur near the sonic points where the eigenvalue, $|\Lambda_c| = |\tilde{V} \pm \tilde{c}|$ become zero. To avoid this issue, Harten[50] suggested the following entropy correction.

$$|\Lambda_c| = \begin{cases} |\Lambda_c| & \text{if } |\Lambda_c| > \delta \\ \frac{\Lambda_c^2+\delta^2}{2\delta} & \text{if } |\Lambda_c| \leq \delta \end{cases} \tag{3.23}$$

where δ is a small number chosen as a fraction of the speed of sound.

3.1.5 Viscous Flux

This subsection outlines the discretization of the viscous flux vector, $\vec{F}_v$, of the discretized governing equations (3.2). The viscous flux is defined with the gradients of the state variables as opposed to the state variables themselves. Like the convective flux, the viscous flux vector is stored at the edges while the state variables and their gradients are stored at the nodes. The simplest method to define the edge-based gradients is a central scheme. However, this scheme can lead to a decoupling between local terms and edge-based gradients severely restricting the overall accuracy [51]. To prevent this decoupling, additional terms are included. The modified edge-based gradients take the form,

$$\nabla\phi_{ij} = \frac{1}{2}\left(\nabla\phi_i + \nabla\phi_j\right) - \left[\frac{1}{2}\left(\nabla\phi_i + \nabla\phi_j\right) \cdot \hat{e}_{ij}\right]\hat{e}_{ij} + \frac{\phi_j - \phi_i}{|\vec{x}_j - \vec{x}_i|}\hat{e}_{ij} \tag{3.24}$$

where $\hat{e}_{ij}$ is the unit direction vector pointing from node i to node j. With the gradient terms defined at the nodes, $\vec{F}_v$ is calculated using Eq.(3.5).

3.1.6 Gradient Calculations

Gradients of the state variables are required to calculate the viscous fluxes while gradients of k and ω are required for the turbulence model. Gradients are also used for second-order spatial discretization, which will be discussed later. This subsection discusses the computation of the gradient of a general field variable, stored at the nodes, which can be applied to any particular variable.

Two common methods to compute gradients on unstructured meshes are Green-Gauss and least-squares. Both methods are pointwise methods that can be efficiently implemented for a dual-volume mesh using element-based operations. This subsection will discuss both methods.

The Green-Gauss method is based on Gauss' Theorem, which is written for a general, continuously differentiable, scalar variable, ϕ, as

$$\int\limits_{\Omega} \nabla\phi \, d\Omega = \oint\limits_{\partial\Omega} \phi \, dS. \tag{3.25}$$

When Gauss' Theorem is applied to a control volume and discretized in space, Eq.(3.25) becomes,

$$(\nabla\phi_i)\,\Omega_i = \sum_{j=1}^{N_i} \phi_{ij}\,\hat{n}_{ij}\,S_{ij} \tag{3.26}$$

where the gradient vector $\nabla\phi_i = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right)^T$ is associated with node i and ϕ_{ij} is associated with the edge connecting node i and neighboring node j. The term ϕ_{ij} is edge based and is related to the node-based variables with the central scheme.

$$\phi_{ij} = \frac{1}{2}\left(\phi_i + \phi_j\right). \tag{3.27}$$

The least-squares method is based on a linear Taylor series expansion of a general variable, ϕ, about a given node i. The expansion is written

$$\phi(\vec{x}) = \phi_i + (\vec{x} - \vec{x}_i)\cdot\nabla\phi_i \tag{3.28}$$

where $\vec{x}_i = (x_i, y_i, z_i)^T$ is the location of node i and $\vec{x}$ is the location of $\phi(\vec{x})$. If the linear Taylor expansion is carried out for every neighbor of node i the following system of equations is formed,

$$\begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix} [\nabla \phi_i] = \begin{bmatrix} \phi_{j_1} - \phi_i \\ \phi_{j_2} - \phi_i \\ \vdots \\ \phi_{j_N} - \phi_i \end{bmatrix} \tag{3.29}$$

where $(j_1, j_2, \ldots, j_N)$ represent the set of neighbors of node i. The gradients, $\nabla \phi_i$, are found by solving the system above.

If node i has a greater number of neighbors than dimensions, which is likely the case, then Eq.(3.29) is over defined. The best that can be done is to find the best fit solution using least-squares.

Least-squares is formulated by multiplying both sides of Eq.(3.29) by the transpose of the system matrix.

$$\begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix}^T \begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix} [\nabla \phi_i] = \begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix}^T \begin{bmatrix} \phi_{j_1} - \phi_i \\ \phi_{j_2} - \phi_i \\ \vdots \\ \phi_{j_N} - \phi_i \end{bmatrix} \tag{3.30}$$

This operation creates a well-defined system the solution which minimizes the L2 norm of the system error.

The system matrix in Eq.(3.29) contains strictly geometric terms and can be computed once for each mesh and stored. Additionaly, because Eq.(3.30) is relatively small, the inverse of the system matrix may be symbolically computed producing a solution of the form

$$\nabla\phi_i = \mathbf{P}_i \begin{bmatrix} \phi_{j_1} - \phi_i \\ \phi_{j_2} - \phi_i \\ \vdots \\ \phi_{j_N} - \phi_i \end{bmatrix} \tag{3.31}$$

where $\mathbf{P}_i$ is a matrix of geometric terms pre-computed for every node. The $\mathbf{P}_i$ matrix is written

$$\mathbf{P}_i = \left(\begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix}^T \begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix} \right)^{-1} \begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix}^T \tag{3.32}$$

or, more explicitly

$$\mathbf{P}_i = \frac{1}{a} \begin{bmatrix} \ell_{yy}\ell_{zz} - \ell_{yz}\ell_{yz} & \ell_{xz}\ell_{xz} - \ell_{xx}\ell_{zz} & \ell_{xy}\ell_{yz} - \ell_{xz}\ell_{yy} \\ \ell_{xz}\ell_{yz} - \ell_{xy}\ell_{zz} & \ell_{xx}\ell_{zz} - \ell_{xz}\ell_{xz} & \ell_{xz}\ell_{xy} - \ell_{xx}\ell_{yz} \\ \ell_{xy}\ell_{yz} - \ell_{xz}\ell_{yy} & \ell_{xz}\ell_{xy} - \ell_{xx}\ell_{yz} & \ell_{xx}\ell_{yy} - \ell_{xy}\ell_{xy} \end{bmatrix} \begin{bmatrix} \vec{x}_{j_1} - \vec{x}_i \\ \vec{x}_{j_2} - \vec{x}_i \\ \vdots \\ \vec{x}_{j_N} - \vec{x}_i \end{bmatrix}^T \tag{3.33}$$

where

$$a = \ell_{xx}\ell_{yy}\ell_{zz} + 2\ell_{xy}\ell_{yz}\ell_{xz} - \ell_{xx}\ell_{yz}\ell_{yz} - \ell_{yy}\ell_{xz}\ell_{xz} - \ell_{zz}\ell_{xy}\ell_{xy} \tag{3.34}$$

and

$$\ell_{rs} = \left\{ \begin{array}{c} r_{j_1} - r_i \\ r_{j_2} - r_i \\ \vdots \\ r_{j_N} - r_i \end{array} \right\} \cdot \left\{ \begin{array}{c} s_{j_1} - s_i \\ s_{j_2} - s_i \\ \vdots \\ s_{j_N} - s_i \end{array} \right\}. \tag{3.35}$$

To generalize the least-squares method, weighting coefficients are added to Eq.(3.29) giving,

$$\begin{bmatrix} w_{i_1}\left(\vec{x}_{j_1} - \vec{x}_i\right) \\ w_{i_2}\left(\vec{x}_{j_2} - \vec{x}_i\right) \\ \vdots \\ w_{i_N}\left(\vec{x}_{j_N} - \vec{x}_i\right) \end{bmatrix} [\nabla\phi_i] = \begin{bmatrix} w_{i_1}\left(\phi_{j_1} - \phi_i\right) \\ w_{i_1}\left(\phi_{j_2} - \phi_i\right) \\ \vdots \\ w_{i_1}\left(\phi_{j_N} - \phi_i\right) \end{bmatrix}. \tag{3.36}$$

The weights, w_{i_k}, can be defined as functions of the geometry, the current flow field or a combination of the two [52]. If the definitions are purely geometric, matrix $\mathbf{P}_i$ can be pre-computed. In this work, the weights are defined geometrically as

$$w_{i_k} = |\vec{x}_k - \vec{x}_i|^{-1}. \tag{3.37}$$

For extremely stretched meshes the system in Eq.(3.36) can be singular or nearly singular. To solve these systems, QR decomposition or singular value decomposition can be used [53, 54].

3.1.7 Higher-Order Spatial Discretization

The spatial discretization presented in this chapter has been first-order accurate. It has become standard to expect at least second-order spatial accuracy from computational methods. To achieve second-order accuracy using the finite volume method, the state variables are assumed to vary linearly across the control volume.

In addition to the state variables, gradient vectors are stored in the control volumes. Together, the state variables and their gradients fully define a piecewise, linearly varying approximation to the continuous flow field. Computation of the gradients is covered in the previous section.

The assumption of linear variation affects both the convective and viscous fluxes. As was shown earlier in Eq.(3.17), the computation of the flux vectors require left and right state variables. To account for the linear variation, the left and right state variables are reconstructed using the following,

$$\vec{Q}_L = \vec{Q}_i + \frac{1}{2}\Psi_i \left\{ \nabla\vec{Q}_i \cdot (\vec{x}_j - \vec{x}_i) \right\}, \tag{3.38}$$

and

$$\vec{Q}_R = \vec{Q}_j - \frac{1}{2}\Psi_j \left\{ \nabla\vec{Q}_j \cdot (\vec{x}_j - \vec{x}_i) \right\}, \tag{3.39}$$

where $\vec{Q}_i$ and $\vec{Q}_j$ refer to the state vector associated with nodes on the left and right sides of the edge ij, $\vec{x}_k = (x_k, y_k, z_k)^T$ is the position vector of node k and $\Psi_k \in [0,1]$ acts as a limiter function.

This reconstruction method was first proposed by Barth & Jesperson [55]. In areas of high gradients, this reconstruction method may produce oscillations and spurious solutions. The limiter functions provide a method to prevent these oscillations[52, 56] by enforcing a monotonicity preserving scheme. That is, the local maxima must not increase, the local minima must not decrease and no new extrema may be created. Two popular limiter functions for unstructured meshes are those of Barth & Jespersen [55] and Venkatakrishnan [57, 58].

3.2 Temporal Discretization

Discussions in the previous section focused on formulating the governing equations, originally introduced in Chapter 2, as a set of ordinary differential equations in time. This was achieved by introducing spatial discretization. The next step is to solve the ODE's by integrating them forward in time. That is the focus of this section.

Spatial discretization has left a system of ordinary differential equations in time of the form

$$\frac{\partial}{\partial t}(\vec{Q}_i\,\Omega_i) = -\sum_{j=1}^{N_i}(\vec{F}_c - \vec{F}_v)_{ij}\,S_{ij} + G_i\,\Omega_i \tag{3.40}$$

where the terms $\vec{Q}_i$, Ω_i, S_{ij}, $\vec{F}_c$, $\vec{F}_v$ and $\vec{G}_i$ have all been defined in the previous section. Recalling the definition of the residual, $\vec{R}_i$ from Eq.(3.13), Eq.(3.40) can be written in the form

$$\frac{\partial}{\partial t}(\vec{Q}_i\,\Omega_i) = \vec{R}_i \tag{3.41}$$

If we assume the mesh is fixed and the control volumes do not vary in time, then Ω_i is constant and Eq.(3.41) can be written

$$\frac{\partial}{\partial t}\vec{Q}_i = \vec{R}_i/\Omega_i \tag{3.42}$$

This form of the governing equation, with nearly all spatial consideration lumped into $\vec{R}_i$, will be used in the remainder of this section to discuss temporal discretization.

Equation (3.42) describes a time accurate evolution of the flow field. For cases that are unsteady in nature, a time accurate integration method is required. For these cases, Eq.(3.42) remains unchanged.

For cases where a steady solution is desired, *i.e.*, a flow field unchanging in time, the time derivative term is set to zero and a solution is sought. For a general code written to handle both unsteady and steady cases, it is simplest and most straightforward to reuse the numerical time integration tools required for the unsteady cases to solve the steady cases. This can be done by leaving the time derivative term and integrating the flow field forward in time until the field is unchanging.

The two classes of numerical schemes used to integrate the governing equations are explicit and implicit. The explicit schemes are computationally cheap and conceptually straightforward. They, however, suffer from numeric stability limitations and require a large number of time steps to converge to a solution. Implicit schemes, however, mitigate these stability concerns, offering a quicker route to the solution by requiring many fewer timesteps. However, they require additional computational resources and are more complicated to implement. It should be pointed out that, because of the nonlinearity of the Navier-Stokes and Euler equations, no truly implicit methods are available. However, as will be shown, approximate implicit methods are available that offer much of the savings that pure implicit methods promise.

The remainder of this section is split into three subsection. The first will outline explicit time integration while the second will outline implicit. Both will cover the implementation of the schemes for steady and unsteady cases. The final subsection will cover dual-time stepping, a more robust method for solving unsteady flows.

3.2.1 Explicit Time Integration

The formulation of the explicit time integration scheme starts by using a forward, first-order, finite difference approximation for the time derivative in Eq.(3.42). This is written

$$\frac{\partial}{\partial t}\vec{Q}_i = \frac{\Delta \vec{Q}_i^n}{\Delta t_i^n} = \frac{\vec{R}_i}{\Omega_i} \tag{3.43}$$

where the superscript n is introduced as an index for time and $\Delta(\bullet)^n = (\bullet)^{n+1} - (\bullet)^n$ represents the change of a variable between time steps. Using the definition for $\Delta(\bullet)$, the Eq.(3.43) can be rearranged as

$$\vec{Q}_i^{n+1} = \vec{Q}_i^n + \frac{\Delta t_i^n}{\Omega_i}\vec{R}_i^n. \tag{3.44}$$

Equation (3.44) provides a method to update the state vector, $\vec{Q}_i$, at every time step $n+1$ using only information provided in time step n. The method to compute $\vec{R}_i^n$ is provided in the previous section. Finding an adequate time step size, Δt_i^n, will be covered later in this subsection. In this way, once an initial value is provided, the solution for $\vec{Q}_i$ can be found for all time.

The method shown above offers only solutions that are first-order accurate in time. Higher-order solutions can be obtained using Runge-Kutta methods. Runge-Kutta methods include an entire family of schemes. The specific methods used in this work are a 3-step and 4-step that have been derived to allow for simple implementation.

The 3-step Runge-Kutta method is written as

$$\begin{aligned}
\vec{Q}_i^{(0)} &= \vec{Q}_i^n \\
\vec{Q}_i^{(1)} &= \vec{Q}_i^{(0)} + \alpha_1\, \Delta t_i\, \vec{R}_i\left(\vec{Q}_i^{(0)}\right)/\Omega_i \\
\vec{Q}_i^{(2)} &= \vec{Q}_i^{(0)} + \alpha_2\, \Delta t_i\, \vec{R}_i\left(\vec{Q}_i^{(1)}\right)/\Omega_i \\
\vec{Q}_i^{(3)} &= \vec{Q}_i^{(0)} + \alpha_3\, \Delta t_i\, \vec{R}_i\left(\vec{Q}_i^{(2)}\right)/\Omega_i \\
\vec{Q}_i^{n+1} &= \vec{Q}_i^{(3)}
\end{aligned} \tag{3.45}$$

where the stage coefficients are

$$
\begin{aligned}
\alpha_1 &= 0.1918 \\
\alpha_2 &= 0.4929 \\
\alpha_3 &= 1.0000.
\end{aligned}
\tag{3.46}
$$

The 4-step Runge-Kutta method is written

$$
\begin{aligned}
\vec{Q}_i^{(0)} &= \vec{Q}_i^{n} \\
\vec{Q}_i^{(1)} &= \vec{Q}_i^{(0)} + \alpha_1 \, \Delta t_i \, \vec{R}_i \left(\vec{Q}_i^{(0)} \right) / \Omega_i \\
\vec{Q}_i^{(2)} &= \vec{Q}_i^{(0)} + \alpha_2 \, \Delta t_i \, \vec{R}_i \left(\vec{Q}_i^{(1)} \right) / \Omega_i \\
\vec{Q}_i^{(3)} &= \vec{Q}_i^{(0)} + \alpha_3 \, \Delta t_i \, \vec{R}_i \left(\vec{Q}_i^{(2)} \right) / \Omega_i \\
\vec{Q}_i^{(4)} &= \vec{Q}_i^{(0)} + \alpha_4 \, \Delta t_i \, \vec{R}_i \left(\vec{Q}_i^{(3)} \right) / \Omega_i \\
\vec{Q}_i^{n+1} &= \vec{Q}_i^{(4)}
\end{aligned}
\tag{3.47}
$$

where the stage coefficients are

$$
\begin{aligned}
\alpha_1 &= 0.1084 \\
\alpha_2 &= 0.2602 \\
\alpha_3 &= 0.5052 \\
\alpha_4 &= 1.0000.
\end{aligned}
\tag{3.48}
$$

The three and four step Runge-Kutta methods defined above provide second-order accurate time integration schemes with minimal memory requirements.

The explicit scheme remains numerically stable only up to some maximum time step, Δt. This maximum time step is chosen such that it satisfies the CFL condition. The CFL condition, formulated by Courant, Friedrichs and Lewy [59], states that the domain of dependence of the numerical scheme must include the domain of dependence of the governing equations. The domain of dependence for the governing equations lies on the characteristic lines $\frac{dx}{dt} = \lambda_k$, where λ_k are the eigenvalues of the flux Jacobian matrix and quantify the speed at which information flows within the system. Consequently, the domain of dependence for the governing equations at point x_i includes points $x_k^* = x_i - \lambda_k \, \Delta t$. Therefore, to satisfy the CFL condition, the domain of dependence for the numerical scheme, defined as the discretization stencil, must enclose all of the points x_k^*.

From the CFL condition, a relationship can be introduced between the time step, Δt, and the distance between neighboring nodes, Δx, that must hold for the explicit scheme to be stable. The eigenvalue, λ_k, with the largest absolute value is the most restrictive and is therefore used in the definition.

$$\lambda_k \, \Delta t \leq \Delta x \tag{3.49}$$

Because Δx is prescribed by the mesh and λ_k is dependent on current flow conditions, the time step Δt is adjusted to fufil Eq.(3.49). A user defined coefficient σ, referred to as the CFL number, is introduced to offer a simple means to adjust the time step

$$\Delta t = \sigma \frac{\Delta x}{|\lambda_k|} \tag{3.50}$$

A further discussion of the CFL condition can be found in [28, 60].

Combining (3.50) and the explicit time integration scheme (3.42), along with the spatial discretization leads to the following condition [28],

$$\Delta t_i^n = \sigma \frac{\Omega_i}{(\Lambda_c^x + \Lambda_c^y + \Lambda_c^z)_i^n + 4(\Lambda_v^x + \Lambda_v^y + \Lambda_v^z)_i^n} \tag{3.51}$$

where Δt_i^n is a stable time step for node i and time step n. The convective spectral radii [61], $\Lambda_c^x, \Lambda_c^y, \Lambda_c^z$, are defined as

$$\begin{aligned}\Lambda_c^x &= (|u| + c)\, S^x \\ \Lambda_c^y &= (|u| + c)\, S^y \\ \Lambda_c^z &= (|u| + c)\, S^z\end{aligned} \tag{3.52}$$

and the viscous spectral radii [28], $\Lambda_v^x, \Lambda_v^y, \Lambda_v^z$, are defined as

$$\begin{aligned}\Lambda_v^x &= \max\left(\tfrac{4}{3\rho}, \tfrac{\gamma}{\rho}\right)\left(\tfrac{\mu}{\mathrm{Pr}} + \tfrac{\mu_T}{\mathrm{Pr}_T}\right)\tfrac{(S^x)^2}{\Omega} \\ \Lambda_v^y &= \max\left(\tfrac{4}{3\rho}, \tfrac{\gamma}{\rho}\right)\left(\tfrac{\mu}{\mathrm{Pr}} + \tfrac{\mu_T}{\mathrm{Pr}_T}\right)\tfrac{(S^y)^2}{\Omega} \\ \Lambda_v^z &= \max\left(\tfrac{4}{3\rho}, \tfrac{\gamma}{\rho}\right)\left(\tfrac{\mu}{\mathrm{Pr}} + \tfrac{\mu_T}{\mathrm{Pr}_T}\right)\tfrac{(S^z)^2}{\Omega}\end{aligned} \tag{3.53}$$

Laminar and turbulent viscosity, μ and μ_T respectively, as well as laminar and turbulent Prandtl numbers, Pr and Pr_T respectively, were defined in Chapter 2. The variables S^x, S^y, and S^z are the projections of the volume Ω on the $y - z$, $z - x$, and $x - y$ planes. They are defined as

$$\begin{aligned}S_i^x &= \tfrac{1}{2}\textstyle\sum_{j=1}^{N_i} |n_{x,ij}\; S_{ij}| \\ S_i^y &= \tfrac{1}{2}\textstyle\sum_{j=1}^{N_i} |n_{y,ij}\; S_{ij}| \\ S_i^z &= \tfrac{1}{2}\textstyle\sum_{j=1}^{N_i} |n_{z,ij}\; S_{ij}|\end{aligned} \tag{3.54}$$

where N_i is the number of neighbors of node i.

Following the process above, a stable time step Δt_i^n can be found for each node and time step. If the case being simulated is unsteady then a time accurate solution is required. To enforce time accuracy, the time step for every node in the domain must be the same. To ensure the computation remains stable everywhere, a time step is calculated for each node and the smallest time step is then enforced everywhere. If the case, however, is steady, there is no concern for time accuracy. It is much quicker to allow each node to use their own local largest stable time step. A steady solution can then be found with minimal computational effort.

It should be noted that the CFL condition is derived assuming information travels at a constant rate. This is not generally the case. The arguments above are taken as an approximation and used only as a guide. Often it is found that, for the CFL coefficient, a value less than one is required to keep the solution stable. An appropriate CFL coefficient must be found for each case.

3.2.2 Implicit Time Integration

To avoid the constraint on the size of the time step inherent in explicit schemes, an implicit time integration scheme may be used. To begin the formulation of an implicit scheme we return to the forward approximation of Eq.(3.43).

$$\frac{\Delta \vec{Q}_i^n}{\Delta t_i^n} = \left(\beta \vec{R}_i^{n+1} + (1-\beta) \vec{R}_i^n \right) / \, \Omega_i \tag{3.55}$$

but replace residual, $\vec{R}_i^n$, with an average of residuals between time steps $n+1$ and n. Coefficient β may be adjusted to weight the average.

The solution of Eq.(3.55) requires the evaluation of the residual term, $\vec{R}^{n+1}$. This cannot be done directly because it depends on the, as of yet, unknown state vector, $\vec{Q}^{n+1}$. An approximation is introduced for $\vec{R}^{n+1}$ by linearizing about the current time step. This is written

$$\vec{R}^{n+1} \approx \vec{R}^n + \left[\frac{\partial \vec{R}}{\partial \vec{Q}} \right]^n \Delta \vec{Q}^n \tag{3.56}$$

where $\frac{\partial \vec{R}}{\partial \vec{Q}}$ is the Jacobian matrix. Using Eq.(3.56), Eq.(3.55) is rearranged to give the implicit scheme.

$$\left(\frac{\Omega_i}{\Delta t^n} [I] - \beta \left[\frac{\partial \vec{R}}{\partial \vec{Q}} \right]^n \right) \Delta \vec{Q}^n = \vec{R}^n \tag{3.57}$$

The term in the square brackets is referred to as the implicit operator. When the entire computational domain is taken into consideration, the implicit operator becomes a large, sparse, non-symmetric, square, block matrix of size 5×number of nodes. The system must be solved for $\Delta\vec{Q}_i^n$ at every step. Directly inverting the full implicit operator is impractical so various methods have been suggested to iteratively invert it. These methods attempt to take advantage of the sparse, block structure of the operator to construct an efficient method. Many practical applications require very large number of nodes and can cause memory issues for many standard matrix inversion approaches.

Two general categories of methods are used to iteratively invert the implicit operator. The first category is based on Krylov-subspace methods. Their similarity to Newton methods leads to them being called Newton-Krylov methods. Several methods of this type exist, such as Conjugate Gradient Squared [62], Bi-Conjugate Gradient Stabilized [63], Transpose-Free Quasi-Minimum Residual [64], and the most popular, Generalised Minimum Residual [65, 66, 67, 68, 69]. The second solution category is based on decomposing the implicit operator into several parts, each easier to invert than the whole. Various decomposition methods include Alternating Direction Implicit (ADI), Lower-Upper Symmetric Gauss-Seidel [70, 71, 72] and Block Lower-Upper Symmetric Gauss-Seidel [73]. The work here utilizes LU-SGS and BLU-SGS methods because of their simple implementation and their low memory requirements. The remainder of this subsection details their implementations for unstructured meshes.

The derivations of the LU-SGS and BLU-SGS methods both begin with the formulation of the global Jacobian term, $\frac{\partial\vec{R}}{\partial\vec{Q}}$, in Eq. (3.57). The global Jacobian can be written as a sparse matrix composed of 5×5 blocks of local Jacobians. The diagonal blocks come from local Jacobians, $\frac{\partial\vec{R}_i}{\partial\vec{Q}_i}$, relating the residual at node i, $\vec{R}_i$, with the state variables at node i, $\vec{Q}_i$. The off-diagonal blocks come from local

Jacobians, $\frac{\partial \vec{R}_i}{\partial \vec{Q}_j}$, relating the residual at node i, $\vec{R}_i$, with the state variables at a neighboring node, $\vec{Q}_j$. To derive the local Jacobian terms, we first return to Eq. 3.13 for the definition of the residual at node i

$$\vec{R}_i^n = -\sum_{j=1}^{N_i} \left(\vec{F}_{c,ij} - \vec{F}_{v,ij}\right)^n S_{ij}. \tag{3.58}$$

The source term, G_i, is omitted for simplicity. To derive the local Jacobians on the diagonal, the derivative of Eq. (3.58) is taken with respect to the state vector at node i, $\vec{Q}_i$.

$$\left[\frac{\partial \vec{R}_i}{\partial \vec{Q}_i}\right]^n = \sum_{j=1}^{N_i} \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_i} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_i}\right)^n S_{ij} \tag{3.59}$$

The convective and viscous flux Jacobians, $\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_i}$ and $\frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_i}$ are discussed below. Similarly, to derive the local Jacobians off of the diagonal, the derivative of Eq. (3.58) is taken with respect to the state vector at a neighboring node j, $\vec{Q}_j$.

$$\left[\frac{\partial \vec{R}_i}{\partial \vec{Q}_j}\right]^n = \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_j} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_j}\right)^n S_{ij} \tag{3.60}$$

The two convective flux Jacobian terms, $\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_i}$ and $\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_j}$, can be evaluated by considering Roe's Riemann upwind scheme presented in Eq. (3.17). The definition for the convective flux vector is

$$\vec{F}_{c,ij}^n = \frac{1}{2}\left[\vec{F}_{c,i} + \vec{F}_{c,j} - |\tilde{A}_{ij}|(\vec{Q}_j - \vec{Q}_i)\right]^n \tag{3.61}$$

where $\vec{F}_{c,k}$ is the flux vector evaluated at node k. It follows that the convective flux Jacobian terms can be written

$$\left[\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_i}\right]^n = \frac{1}{2}\left[\frac{\partial \vec{F}_{c,i}}{\partial \vec{Q}_i} + |\tilde{A}_{ij}|\right]^n \tag{3.62}$$

and

$$\left[\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_j}\right]^n = \frac{1}{2}\left[\frac{\partial \vec{F}_{c,i}}{\partial \vec{Q}_j} - |\tilde{A}_{ij}|\right]^n . \tag{3.63}$$

Roe's matrix, $|\tilde{A}_{ij}|$, is defined in the spatial discretization section. The local convective flux Jacobian is defined as [33]

$$\frac{\partial \vec{F}_c}{\partial \vec{Q}} = \begin{bmatrix} 0 & n_x & n_y & n_z & 0 \\ n_x\phi - uV & V - a_3 n_x u & n_y u - a_2 n_x v & n_z u - a_2 n_x w & a_2 n_x \\ n_y\phi - vV & n_x v - a_2 n_y u & V - a_3 n_y v & n_z v - a_2 n_y w & a_2 n_y \\ n_z\phi - wV & n_x w - a_2 n_z u & n_y w - a_2 n_z v & V - a_3 n_z w & a_2 n_z \\ V(\phi - a_1) & n_x a_1 - a_2 uV & n_y a_1 - a_2 vV & n_x a_1 - a_2 wV & \gamma V \end{bmatrix} \tag{3.64}$$

with the definitions

$$\begin{aligned} a_1 &= \gamma E - \phi \\ a_2 &= \gamma - 1 \\ a_3 &= \gamma - 2 \\ V &= n_x u + n_y v + n_z w \\ \phi &= \tfrac{1}{2}(\gamma - 1)\left(u^2 + v^2 + w^2\right). \end{aligned} \tag{3.65}$$

Recall that fluxes are associated with a unit direction which, in the unstructured finite volume formulation, is equal to the unit normal vector of the edge, $\hat{n}_{ij}$.

The BLU-SGS method retains the flux Jacobians as 5 × 5 matrices while the LU-SGS method, to lower the total number of calculations required, approximates the matrix as a diagonal matrix of eigenvalues. This approximation, while quicker per iteration, will limit the maximum stable CFL value and can end up being slower by requiring more total iterations to converge.

The viscous flux Jacobian, $\frac{\partial \vec{F_v}}{\partial \vec{Q}}$, is known to be difficult to derive because of its strong dependency on gradients. Several methods were implemented to calculate the Jacobian and a switch was added to allow the user to decide which method to use. All methods are described below.

The first method, referred to in this dissertation as a diagonalized Jacobian, is proposed in Kim and Kwon [74]. Similar to how LU-SGS treats the convective flux Jacobian, only a diagonal matrix is used for the viscous flux Jacobian. The motivation, besides greatly reducing the numeric expense, is to enhance the diagonal dominance of the viscous flux Jacobian resulting in a more stable scheme.

The diagonalized Jacobian consists of the eigenvalues of the full Jacobian. The smallest eigenvalue is used. The value is $\frac{\mu}{\rho}\frac{1}{\Delta x_{ij}}$ where Δx_{ij} is the distance between nodes i and j.

For some cases, however, the diagonalized Jacobian destabilized the code to the point that the implicit scheme gave no advantage over the explicit. What was needed was a method to calculate the viscous flux Jacobian which has the minimal expensive of the diagonalized method while retaining the accuracy of the full matrix.

It was determined that, by employing the least squares gradient calculations in the derivation of the viscous flux Jacobian, a direct viscous flux Jacobian can be derived that depended directly on the state vector instead of its gradient. The least squares gradient calculations are inexpensive once the geometric weighting coefficients are calculated. Because these coefficients are already being used and don't need to be recalculated, the additional expense is minimal.

Also, for a third option, it was found that the code could be run accurately by eliminating the viscous flux Jacobian altogether and use only the convective flux Jacobians in the implicit term. This, fortunately, provides a last case scenario option that can be employed for all cases.

Substituting the definitions for the convective and viscous flux Jacobians into (3.57) gives

$$\left[\frac{\Omega_i}{\Delta t_i} I + \sum_{j=1}^{N_i} \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_i} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_i} \right) \right]^n \Delta \vec{Q}_i^n = \vec{R}_i^n + \left[\sum_{j=1}^{N_i} \left\{ \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_j} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_j} \right) \Delta \vec{Q}_j^n \right\} \right]^n . \tag{3.66}$$

Equation (3.66) is written out for the local node i. It can be thought of as a block of rows in the complete implicit matrix. The bracketed term on the left becomes a 5×5 block on the diagonal of the implicit matrix and the bracketed term on the right becomes the off-diagonal blocks.

The global implicit matrix can factored into three parts for LU-SGS and BLU-SGS. They are defined as

$$\mathbf{D} = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_M \end{bmatrix} \tag{3.67}$$

where

$$D_i^n = \left[\frac{\Omega_i}{\Delta t_i} I + \sum_{j=1}^{N_i} \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_i} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_i} \right) \right]^n , \tag{3.68}$$

$$\mathbf{U} = \begin{bmatrix} 0 & U_{1,2} & U_{1,3} & \ldots & U_{1,M} \\ & 0 & U_{2,3} & \ldots & U_{2,M} \\ & & 0 & \ddots & \vdots \\ & & & \ddots & U_{M-1,M} \\ & & & & 0 \end{bmatrix} \tag{3.69}$$

where

$$U_{i,j} = \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_j} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_j} \right) \;, \;\; i < j, \tag{3.70}$$

$$\mathbf{L} = \begin{bmatrix} 0 & & & & \\ L_{2,1} & 0 & & & \\ L_{3,1} & L_{3,2} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ L_{M,1} & L_{M,2} & \ldots & L_{M,M-1} & 0 \end{bmatrix} \tag{3.71}$$

with

$$L_{i,j} = \left(\frac{\partial \vec{F}_{c,ij}}{\partial \vec{Q}_j} - \frac{\partial \vec{F}_{v,ij}}{\partial \vec{Q}_j} \right) \;, \;\; i > j, \tag{3.72}$$

where, for all above, M is the total number of nodes in the system. The matrix D is made up of the blocks on the diagonal of the global implicit matrix, matrix U is made up of blocks above the diagonal, and matrix L, of blocks below the diagonal.

The complete system can then be written as

$$[\mathbf{D} + \mathbf{U} + \mathbf{L}]^n \; \Delta\vec{Q}^n = \vec{R}^n, \tag{3.73}$$

where $\vec{Q}^n$ and $\vec{R}^n$ are conglomerations of the state vector, $\vec{Q}_i^n$, and the residual vector, $\vec{R}_i^n$, respectively.

Inversion of the implicit operator using LU-SGS or BLU-SGS leads to an iterative process outlined with the following pseudocode.

```
$\Delta\vec{Q}^k = \Delta\vec{Q}^{n-1}$
do $k = 1, k_{max}$
    ! Forward Sweep
    do $i = 1, M$
        $\Delta\vec{Q}_i^{k+1/2} = [D_i^n]^{-1} \left\{ \vec{R}_i^n + \sum_{j=1}^{N_i} (U_{i,j^n}\, \Delta\vec{Q}_j^k + L_{i,j}^n\, \Delta\vec{Q}_j^{k+1/2}) \right\}$
    end do
    ! Backward Sweep                                                    (3.74)
    do $i = M, 1, -1$
        $\Delta\vec{Q}_i^{k+1} = [D_i^n]^{-1} \left\{ \vec{R}_i^n + \sum_{j=1}^{N_i} (U_{i,j}^n\, \Delta\vec{Q}_j^{k+1} + L_{i,j}^n\, \Delta\vec{Q}_j^{k+1/2}) \right\}$
    end do
    Check for Convergence
end do
$\Delta\vec{Q}^n = \Delta\vec{Q}^{k_{max}}$
```

In the pseudo-code, k_{max} is the maximum number of iterations, M is the number of nodes in the domain, and N_i is the set of neighbors of node i.

The process consists of a forward sweep through the nodes solving for $\Delta Q^{k+1/2}$ and a backward sweep solving for ΔQ^{k+1}. These forward and backward sweeps give the symmetric flavor to the algorithm. Also, consistent with Gauss-Seidel methods, the most up-to-date value for ΔQ_j is used for both sweeps due to matrix U containing only terms with $i > j$ and matrix L containing only terms with $i < j$.

This algorithm reduces the global matrix inversion to a series of local 5×5 matrix inversions which are more easily handled. For LU-SGS, the local matrix D_i^n is diagonal and the inversion is straightforward. For BLU-SGS, D_i^n is fully populated. Because of its small size, it is best to use a direct solver for the inversion of D_i^n.

The algorithm is iterated upon until convergence is reached or the maximum number of iterations is reached. The stopping conditions for the LU-SGS or BLU-SGS should be adjusted depending on the case being simulated. Approximately, convergence can be declared if the implicit residual drops 2-3 orders of magnitude with a maximum iteration limit set to 10.

3.2.3 Dual-Time Stepping

Previous subsections covered explicit and implicit time integration schemes. Both of these methods work well for finding steady solutions and unsteady solutions in which the time scale prescribed by the physics is near the time scale prescribed by the numerics. For unsteady cases, however, where the time scale of the physics can be much larger than that of the numerics, the CFL condition severely limits the allowable time step size leading to a large required number of time steps. Implicit schemes loosen these restrictions. To completely remove the time scale limitations dictated by numeric stability, and allow time steps more appropriate for the given physics, dual-time stepping must be used.

Development of the dual-time stepping scheme begins by discretizing the time derivative term in the semi-discrete Eq.(3.42) with a second-order backward difference approximation,

$$\frac{3\,\vec{Q}_i^{n+1} - 4\,\vec{Q}_i^{n} + \vec{Q}_i^{n-1}}{2\,\Delta t}\Omega_i = \vec{R}_i^{n+1} \tag{3.75}$$

where Δt is the global physical time step prescribed by the user and is chosen so the physical features in the flow are captured. No consideration is given at this point for numerical stability.

The dual-time stepping scheme then updates the flow field by solving Eq.(3.75) for $\vec{Q}_i^{n+1}$ at every node i. The method used to solve the system dictates the details of the method. Because Eq.(3.75) is non-linear and a non-linear solver has already been developed to solve the steady system, it is convenient to use the same method here. Essentially, we cast each physical time step in the dual-time stepping scheme as a unique steady problem and solve them one at a time using the converged solution from one time step as the initial condition for the next.

Casting the system in Eq.(3.75) as a steady problem results in the following form,

$$\frac{\partial}{\partial t^*}\left(\vec{Q}_i^*\right)\,\Omega_i = \vec{R}_i^* \tag{3.76}$$

where $\vec{Q}_i^*$ is an approximation to $\vec{Q}_i^{n+1}$ and t^* is pseudo-time variable added to solve the equation. The pseudo-time residual, $\vec{R}_i^*$, is defined as

$$\vec{R}_i^* = \vec{R}_i - \left(\frac{3}{2\,\Delta t}\Omega_i\,\vec{Q}_i^* - \vec{G}_i^*\right) \tag{3.77}$$

where $\vec{R}_i$ is the standard residual introduced earlier (3.13). All terms that are constant during the current dual-time step are lumped into a pseudo-time source term, $\vec{G}_i^*$. It takes the following form,

$$\vec{G}_i^* = \frac{2}{\Delta t}\Omega_i\,\vec{Q}_i^n - \frac{1}{2\,\Delta t}\Omega_i\,\vec{Q}_i^{n-1}. \tag{3.78}$$

Either the explicit or implicit time marching schemes developed earlier in this section may be used to solve Eq.(3.76). Once the equation is converged, the pseudo-time derivative, $\frac{\partial}{\partial t^*}(\vec{Q}_i^*)$, drops out, Eq.(3.75) is satisfied, and the state vector is updated, $\vec{Q}_i^{n+1} = \vec{Q}_i^*$. The system is then stepped forward in time and a new steady problem is set up and solved. The remainder of this subsection will develop the dual-time stepping method for the explicit and implicit schemes.

A Runge-Kutta explicit time stepping scheme can be used to solve Eq.(3.76). The Runge-Kutta stages take the following form,

$$\begin{array}{rcl}
\vec{Q}_i^{(0)} & = & \vec{Q}_i^{(*,k)} \\
\vec{Q}_i^{(1)} & = & \vec{Q}_i^{(0)} - \alpha_1 \frac{\Delta t_i^*}{\Omega_i} \vec{R}_i^*\Big(\vec{Q}_i^{(0)}\Big) \\
\vec{Q}_i^{(2)} & = & \vec{Q}_i^{(0)} - \alpha_2 \frac{\Delta t_i^*}{\Omega_i} \vec{R}_i^*\Big(\vec{Q}_i^{(1)}\Big) \\
 & \vdots & \\
\vec{Q}_i^{(m)} & = & \vec{Q}_i^{(0)} - \alpha_m \frac{\Delta t_i^*}{\Omega_i} \vec{R}_i^*\Big(\vec{Q}_i^{(m-1)}\Big) \\
\vec{Q}_i^{(*,k+1)} & = & \vec{Q}_i^{m}
\end{array} \tag{3.79}$$

where k denotes the current and $(k+1)$ the new pseudo-time iteration. The initial guess for the pseudo-time marching is $\vec{Q}_i^n$ for every node i. The marching is typically continued until $\vec{R}_i^*$ is reduced by four orders of magnitude.

3.2.3.1 Explicit Dual-Time Stepping

As was pointed out by Arnone et al. [75], the explicit scheme (3.79) becomes unstable when the physical time step, Δt, is of the same order as the pseudo-time step, Δt^*. Melson et al. [76] showed that the instability is due to the term $\frac{3}{2\,\Delta t}\Omega_i\,\vec{Q}_i^*$ from Eq.(3.77). To avoid this instability, the culprit term can be treated implicitly. The explicit scheme in Eq.(3.79) is modified to take the following form,

$$\vec{Q}_i^{(m)} = \vec{Q}_i^{(0)} - \alpha_m \frac{\Delta t_i^*}{\Omega_i} \left[1 + \frac{3}{2\Delta t} \alpha_m \Delta t_i^* \beta \right]^{-1} \cdot \left[\vec{R}_i^* \Big(\vec{Q}^{(m-1)} \Big) - \frac{3}{2\Delta t} \Omega_i \, \beta \, \vec{Q}_i^{(m-1)} \right] \tag{3.80}$$

where β can be adjusted to stabilize the scheme. Literature suggest that choosing $\beta = 2$ provides sufficient stability [58, 77]. Additional development of the dual-time stepping scheme with explicit time marching can be found in Refs. [75, 78, 79, 80, 81, 82].

3.2.3.2 Implicit Dual-Time Stepping

Using the implicit scheme in dual-time stepping begins by formulating Eq.(3.76) as

$$\frac{\partial}{\partial t^*} \left(\vec{Q}_i^* \right) \Omega_i = \vec{R}_i^{(*,k+1)} \tag{3.81}$$

with $(k+1)$ being the new pseudo-time iteration. The pseudo-time residual, $\vec{R}_i^*$, can be linearized in pseudo-time as

$$\vec{R}^{(*,k+1)} \approx \vec{R}^{(*,k)} + \frac{\partial \vec{R}^*}{\partial \vec{Q}^*} \Delta \vec{Q}^*. \tag{3.82}$$

The pseudo-time Jacobian, $\frac{\partial \vec{R}^*}{\partial \vec{Q}^*}$, takes the form

$$\frac{\partial \vec{R}^*}{\partial \vec{Q}^*} = \frac{\partial \vec{R}}{\partial \vec{Q}} + \frac{3}{2\,\Delta t} \Omega_i \tag{3.83}$$

where $\frac{\partial \vec{R}}{\partial \vec{Q}}$ is the Jacobian. This leads to the following form for the implicit dual-time stepping scheme,

$$\left[\left(\frac{1}{\Delta t_i^*}+\frac{3}{2\,\Delta t}\right)\,\Omega_i+\left(\frac{\partial\vec{R}}{\partial\vec{Q}}\right)_i\right]\Delta\vec{Q}_i^*=-\vec{R}_i^{(*,k)}. \tag{3.84}$$

Equation (3.84) can be solved using the LU-SGS or BLU-SGS methods. Further discussion on dual-time stepping schemes using implicit time marching can be found in Refs. [83, 84, 85].

3.3 Time-Linearization

This section introduces the time-linearization approximation that is used to calculate rotordynamic coefficients. It outlines the time-linearization approximation and how it manifests itself in the discretized Navier-Stokes equations that were derived in previous sections. Additional solution methods are discussed that are needed to handle the additional terms generated by the time-linearization approximation. Finally, techniques are introduced to generate a time-linearized mesh upon which the time-linearized Navier-Stokes equations operate.

3.3.1 Small Perturbation Theory

The time-linearization approximation is based on small perturbation theory. Small perturbation theory is a powerful approach that is used to find approximate solutions to difficult to solve problems [86]. It is especially useful for problems containing multiple time and spatial scales. For instance, two spatial scales are considered for the gas seals in this work. The larger scale is associated with the clearance distance between the rotor and the stator while the smaller scale is associated with the amplitude of the rotor whirl. This suggests splitting the solution into two parts as

$$\phi = \phi_0 + \epsilon\phi_1 \tag{3.85}$$

with larger scale ϕ_0 and smaller scale ϕ_1 added together. Here, ϵ is a small variable, usually some parameter of the problem, that represents the ratio between the scales. ϕ_1 is called a small perturbation of ϕ_0. Additional scales and perturbations may be considered by extending Eq.(3.85) and forming a power series in ϵ,

$$\phi = \phi_0 + \epsilon\phi_1 + \epsilon^2\phi_2 + \dots \tag{3.86}$$

When the power series is used in place of the original term in a function, a new power series is formed. For example, using the simple function $f(\phi) = \phi^2$ results in the following power series,

$$f(\phi) = \phi_0^2 + \epsilon\,(2\phi_0\phi_1) + \epsilon^2\left(\phi_1^2 + 2\phi_0\phi_2\right) + \dots \tag{3.87}$$

It can be seen how the lower order terms of ϕ appear in the higher order terms of f, but the higher order terms do not appear in lower order terms. This generates a cascade of influence from low order to high. Approximations may be made to f to within a given ϵ order by truncating the power series.

3.3.2 Linearization of Spatial Coordinates

The derivation of the time-linearized Navier-Stokes equations begins with addressing the linearized motion of the mesh and how it effects the governing equations. Recall that the motion of the domain is caused by the harmonic whirl of the rotor. The amplitude of the whirl is very small compared to the seal clearance allowing for the linearization approximation to hold. The position and motion of the center of the rotor can be described with

$$\vec{x}_r(t) = \vec{x}_{r,0} + \epsilon\,\Re\big[\vec{x}_{r,1}e^{j\omega t}\big] \tag{3.88}$$

where $\vec{x}_r(t)$ is the position as a function of time, $\vec{x}_{r,0}$ is the unperturbed position, $\vec{x}_{r,1}$ is a complex number that contains information on the phase-shift, $\arg\vec{x}_{r,1}$, and amplitude, $\|\vec{x}_{r,1}\|$, of the whirl, $j = \sqrt{-1}$, ω is the whirl frequency, and $\Re\,[\bullet]$ is the real part of the term. Also, ϵ is a small value that's included to scale the amplitude and enforce the small perturbation assumption. Figure 3.4 shows the motion of the center of a whirling rotor.

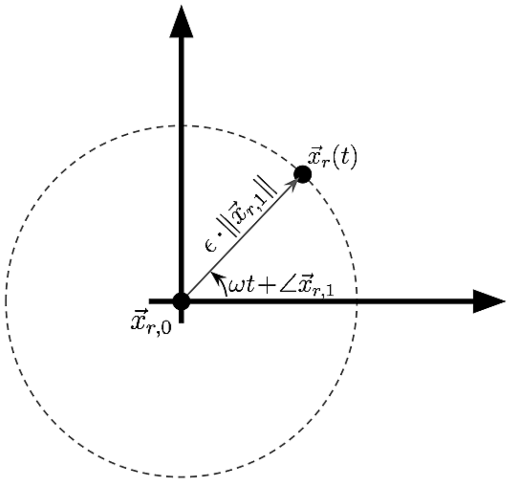

$\vec{x}_r(t)$
$\epsilon \cdot \|\vec{x}_{r,1}\|$
$\omega t + \angle \vec{x}_{r,1}$
$\vec{x}_{r,0}$

Figure 3.4: Motion of the center of a whirling rotor.

To account for the whirl motion of the rotor, the computational domain must deform with a matching harmonic motion. Figure 3.5 shows an example of a deforming mesh. The moving, physical coordinates, (x, y, z, t), therefore, are related to the steady, zeroth-order coordinates, (x_0, y_0, z_0, t_0), by,

$$\begin{aligned} x(x_0, y_0, z_0, t_0) &= x_0 + \epsilon\, x_1(x_0, y_0, z_0)e^{j\omega t} \\ y(x_0, y_0, z_0, t_0) &= y_0 + \epsilon\, y_1(x_0, y_0, z_0)e^{j\omega t} \\ z(x_0, y_0, z_0, t_0) &= z_0 + \epsilon\, z_1(x_0, y_0, z_0)e^{j\omega t} \\ t(x_0, y_0, z_0, t_0) &= t_0 \end{aligned} \tag{3.89}$$

where x_1, y_1, and z_1, are perturbations of the mesh. Amplitude, ϵ, and frequency, ω are the same as Eq. (3.88) and remain constant throughout the simulation.

To project the governing equations on to the moving mesh, it is necessary to construct a relationship between derivative operators in the physical and computational coordinate frame. This relationship is given by

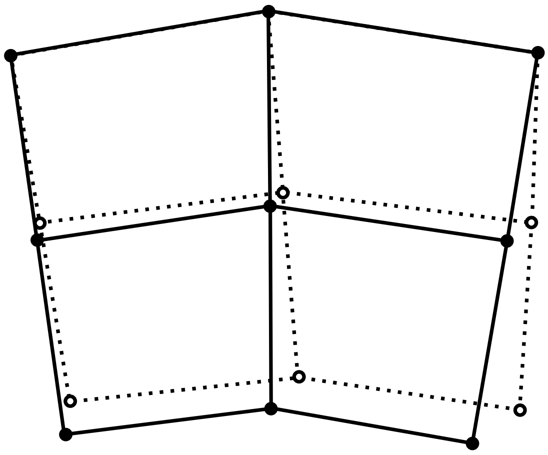


Figure 3.5: Example of an undeformed (solid) and deformed (dashed) mesh.

$$\begin{pmatrix} \frac{\partial}{\partial x_0} \\ \frac{\partial}{\partial y_0} \\ \frac{\partial}{\partial z_0} \\ \frac{\partial}{\partial t_0} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial y}{\partial x_0} & \frac{\partial z}{\partial x_0} & \frac{\partial t}{\partial x_0} \\ \frac{\partial x}{\partial y_0} & \frac{\partial y}{\partial y_0} & \frac{\partial z}{\partial y_0} & \frac{\partial t}{\partial y_0} \\ \frac{\partial x}{\partial z_0} & \frac{\partial y}{\partial z_0} & \frac{\partial z}{\partial z_0} & \frac{\partial t}{\partial z_0} \\ \frac{\partial x}{\partial t_0} & \frac{\partial y}{\partial t_0} & \frac{\partial z}{\partial t_0} & \frac{\partial t}{\partial t_0} \end{bmatrix} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \\ \frac{\partial}{\partial t} \end{pmatrix} = [J] \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \\ \frac{\partial}{\partial t} \end{pmatrix} \tag{3.90}$$

where J is the Jacobian matrix associated with the transformation from physical to computational frames. The inverse relationship is

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \\ \frac{\partial}{\partial t} \end{pmatrix} = [J]^{-1} \begin{pmatrix} \frac{\partial}{\partial x_0} \\ \frac{\partial}{\partial y_0} \\ \frac{\partial}{\partial z_0} \\ \frac{\partial}{\partial t_0} \end{pmatrix} \tag{3.91}$$

Using the relationships defined in Eq.(3.89), the matrix J^{-1} is

$$[J]^{-1} = \begin{bmatrix} 1-\epsilon\frac{\partial x_1}{\partial x_0} & -\epsilon\frac{\partial y_1}{\partial x_0} & -\epsilon\frac{\partial z_1}{\partial x_0} & 0 \\ -\epsilon\frac{\partial x_1}{\partial y_0} & 1-\epsilon\frac{\partial y_1}{\partial y_0} & -\epsilon\frac{\partial z_1}{\partial y_0} & 0 \\ -\epsilon\frac{\partial x_1}{\partial z_0} & -\epsilon\frac{\partial y_1}{\partial z_0} & 1-\epsilon\frac{\partial z_1}{\partial z_0} & 0 \\ -j\omega x_1 & -j\omega y_1 & -j\omega z_1 & 1 \end{bmatrix} \tag{3.92}$$

Neglecting higher order terms, the derivatives in the physical coordinates can be expressed in terms of the computational coordinates with the following relationships

$$\begin{aligned} \frac{\partial}{\partial x} &= \frac{\partial}{\partial x_0} - \epsilon\left(\frac{\partial x_1}{\partial x_0}\frac{\partial}{\partial x_0} + \frac{\partial x_1}{\partial y_0}\frac{\partial}{\partial y_0} + \frac{\partial x_1}{\partial z_0}\frac{\partial}{\partial z_0}\right) \\ \frac{\partial}{\partial y} &= \frac{\partial}{\partial y_0} - \epsilon\left(\frac{\partial y_1}{\partial x_0}\frac{\partial}{\partial x_0} + \frac{\partial y_1}{\partial y_0}\frac{\partial}{\partial y_0} + \frac{\partial y_1}{\partial z_0}\frac{\partial}{\partial z_0}\right) \\ \frac{\partial}{\partial z} &= \frac{\partial}{\partial z_0} - \epsilon\left(\frac{\partial z_1}{\partial x_0}\frac{\partial}{\partial x_0} + \frac{\partial z_1}{\partial y_0}\frac{\partial}{\partial y_0} + \frac{\partial z_1}{\partial z_0}\frac{\partial}{\partial z_0}\right) \\ \frac{\partial}{\partial t} &= \frac{\partial}{\partial t_0} - \epsilon j\omega\left(x_1\frac{\partial}{\partial x_0} + y_1\frac{\partial}{\partial y_0} + z_1\frac{\partial}{\partial z_0}\right). \end{aligned} \tag{3.93}$$

3.3.3 Linearization of the Flow

Unsteadiness in the flow field is being excited by the harmonic whirling of the rotor. This suggest splitting the state variables into the familiar linearized form,

$$\vec{Q}(x_0, y_0, z_0, t_0) = \vec{Q}_0(x_0, y_0, z_0) + \epsilon\,\vec{Q}_1(x_0, y_0, z_0)e^{j\omega t} \tag{3.94}$$

where the vector $\vec{Q}_0$ represents the zeroth-order or mean flow field and vector $\vec{Q}_1$ represents the first-order perturbation. Here, the zeroth-order terms, $\vec{Q}_0$, correspond to a rotor exhibiting no whirling motion. The variations from the mean field caused by the whirling motion of the rotor are small and harmonic in time and are accounted for by the first-order terms, $\vec{Q}_1$, and are written with complex numbers to account for not only the magnitude of the perturbation, but also the phase differences between different parts of the domain.

The time-linearized approximations are carried through to every other term using a robust, if not sometimes tedious, method described below. To clarify this concept, a simple example is given where we linearize the first term in the convective flux vector, $\vec{F}_c$. This example is worked out in 2D to limit the number of terms. The term to be linearized, ϕ, is given as

$$\phi = \vec{m} \cdot \hat{n} = m_x n_x + m_y n_y \tag{3.95}$$

where the vector $\vec{m}$ represents momentum and is defined as $\vec{m} = (\rho u, \rho v)^T$. The components of the momentum, $\vec{m}$, and the normal vector, $\hat{n}$, are expanded into their zeroth and first-order terms.

$$\begin{aligned} m_x &= m_{x,0} + \epsilon\, m_{x,1} e^{j\omega t} \\ m_y &= m_{y,0} + \epsilon\, m_{y,1} e^{j\omega t} \\ n_x &= n_{x,0} + \epsilon\, n_{x,1} e^{j\omega t} \\ n_y &= n_{y,0} + \epsilon\, n_{y,1} e^{j\omega t} \end{aligned} \tag{3.96}$$

The expanded forms in Eq.(3.96) are substituted into the original definition in Eq.(3.95) giving,

$$\phi = (m_{x,0}\, n_{x,0} + m_{y,0}\, n_{y,0}) + \epsilon\ (m_{x,1}\, n_{x,0} + m_{y,1}\, n_{y,0} + m_{x,0}\, n_{x,1} + m_{y,0}\, n_{y,1})\, e^{j\omega t} \tag{3.97}$$

where higher order terms, any terms containing ϵ^2, ϵ^3, *etc.* are neglected. Equation (3.97) is then split into zeroth and first-order terms giving

$$\begin{aligned} \phi_0 &= (m_{x,0}\, n_{x,0} + m_{y,0}\, n_{y,0}) \\ \phi_1 &= (m_{x,1}\, n_{x,0} + m_{y,1}\, n_{y,0} + m_{x,0}\, n_{x,1} + m_{y,0}\, n_{y,1})\, e^{j\omega t}. \end{aligned} \tag{3.98}$$

By examining ϕ_0 in (3.98) and ϕ in (3.95), it can be seen that the definition for the zeroth-order part is nothing more than the full-order definition with zeroth-order terms replacing the original terms. Also, ϕ_1 can be written in the form,

$$\phi_1 = \left(\frac{\partial \phi}{\partial \vec{m}}\right)_0 \vec{m}_1 + \left(\frac{\partial \phi}{\partial \hat{n}}\right)_0 \vec{n}_1 \tag{3.99}$$

where

$$\begin{aligned} \vec{m}_1 &= \left(m_{x,1}, m_{y,1}\right)^T \\ \vec{n}_1 &= \left(n_{x,1}, n_{y,1}\right)^T \end{aligned} \tag{3.100}$$

and following from (3.95),

$$\begin{aligned} \left(\tfrac{\partial \phi}{\partial \vec{m}}\right)_0 &= \left(n_{x,0}, n_{y,0}\right) \\ \left(\tfrac{\partial \phi}{\partial \hat{n}}\right)_0 &= \left(m_{x,0}, m_{y,0}\right). \end{aligned} \tag{3.101}$$

It may appear that the relationship between the first-order terms and the derivatives presented in (3.99) is coincidental. However, using a more direct derivation it can be shown that the relationship holds for all functions. A truncated Taylor series expansion of (3.95) gives,

$$\phi(\Delta \vec{m}, \Delta \vec{n}) = \phi_0 + \left(\frac{\partial \phi}{\partial \vec{m}}\right)_0 \Delta \vec{m} + \left(\frac{\partial \phi}{\partial \hat{n}}\right)_0 \Delta \vec{n} \tag{3.102}$$

If the delta terms are set equal to the first-order perturbation terms, $\Delta \vec{m} = \vec{m}_1$, $\Delta \vec{n} = \vec{n}_1$, then the relationship in (3.99) is recovered.

Similar to how, in the example, ϕ is defined by a flow variable, $\vec{m}$, and a geometric variable, $\hat{n}$, all terms in the full-order RANS solver are directly or in-directly defined by state variables, $\vec{Q}$, and mesh node locations, $\vec{x}$. Note that all geometric terms such as cell volumes, face areas and normal vectors are ultimately defined by mesh node locations. This suggests that a similar process can be followed to derive the zeroth and first-order parts for every term in the RANS solver. Just as with the example above, the zeroth-order part will match the full-order definition and the first-order part will be made up of a derivative with respect to the state variables and a derivative with respect to the mesh node locations.

The result of this method gives a simple and clean form for the definition of the zeroth and first-order terms. This form is written for a general term, ϕ, as,

$$\phi = \phi_0 + \epsilon\, \phi_1 e^{j\omega t} = \phi_0 + \epsilon \left[\frac{\partial \phi}{\partial \vec{x}} \vec{x}_1 + \frac{\partial \phi}{\partial \vec{Q}} \vec{Q}_1 \right] e^{j\omega t} \tag{3.103}$$

where the derivative within the square brackets are defined as

$$\frac{\partial \phi}{\partial \vec{x}} = \left(\frac{\partial \phi}{\partial x_0}, \frac{\partial \phi}{\partial y_0}, \frac{\partial \phi}{\partial z_0} \right)^T \tag{3.104}$$

and

$$\frac{\partial \phi}{\partial \vec{Q}} = \left(\frac{\partial \phi}{\partial \rho_0}, \frac{\partial \phi}{\partial u_0}, \frac{\partial \phi}{\partial v_0}, \frac{\partial \phi}{\partial w_0}, \frac{\partial \phi}{\partial p_0} \right)^T . \tag{3.105}$$

Equations for ϕ_0 and ϕ_1 can be derived by substituting the assumed forms for the mesh positions, $\vec{x}$, and the state variables, $\vec{Q}$, into the formula for $\phi(\vec{x}, \vec{Q})$. Note that if ϕ refers to a purely geometric term, e.g. Ω_i, S_{ij}, $\hat{n}_{ij}$, it will only be affected by the perturbations of the node locations, $\vec{x}$.

3.3.4 Introduction of Face Velocity

The process of linearizing an established full-order flow solver requires introducing new terms. For this work, the full-order solver was originally developed for non-deforming meshes. For a deforming mesh fluxes are calculated relative to the motion of the faces. Therefore, the face velocity is required to compute any flux through that face. For example, the convective flux vector, $\vec{F}_c$, must be modified to account for the motion of the faces as

$$\vec{F}_c^M = \vec{F}_c - V_t\vec{Q} \tag{3.106}$$

where $\vec{F}_c^M$ is the modified convective flux vector, $\vec{F}_c$ is the convective flux vector defined in (3.3), $\vec{Q}$ is the vector of conservative variables defined in (3.1), and V_t is the contravariant velocity of the face. The contravariant face velocity, V_t, is given by

$$V_t = \hat{n} \cdot \frac{\partial \vec{x}}{\partial t} = n_x \frac{\partial x}{\partial t} + n_y \frac{\partial y}{\partial t} + n_z \frac{\partial z}{\partial t} \tag{3.107}$$

where $\hat{n} = (n_x, n_y, n_z)$ is the face unit normal and $\frac{\partial \vec{x}}{\partial t} = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}, \frac{\partial z}{\partial t}\right)$ is the face velocity.

Linearizing (3.106) and splitting the results into zeroth-order and first-order terms gives

$$\begin{aligned} \vec{F}_{c,0}^M &= \vec{F}_{c,0} - V_{t,0}\vec{Q}_0 \\ \vec{F}_{c,1}^M &= \vec{F}_{c,1} - V_{t,0}\vec{Q}_1 - V_{t,1}\vec{Q}_0 \end{aligned} \tag{3.108}$$

Because zeroth-order terms are associated with a steady domain, $V_{t,0}$ is zero, giving

$$\begin{aligned} \vec{F}_{c,0}^M &= \vec{F}_{c,0} \\ \vec{F}_{c,1}^M &= \vec{F}_{c,1} - V_{t,1}\vec{Q}_0 \end{aligned} \tag{3.109}$$

No new terms are required for the zeroth-order system while a first-order contravariant face velocity, $V_{t,1}$, must be included in the first-order system.

When dealing with a dynamic discretized space an additional consideration must be taken into account to ensure geometric terms agree with one another. This idea, referred to as the Geometric Conservation Law (GCL) [87, 88, 89, 90], was first pointed out by Thomas and Lombard [91]. It states that the change in volume of a given 3-D space must equal the volume swept out by its moving boundaries for any period of time. This is written symbolically as

$$\frac{\partial}{\partial t}\int_{\Omega} d\Omega = \oint_{\partial\Omega} V_t\, dS \tag{3.110}$$

and in finite volume form as,

$$\frac{\partial}{\partial t}\Omega_i = \sum_{k=1}^{N(i)} V_{t,ik}\, S_{ik} \tag{3.111}$$

where N_i are the neighbors of cell i and $V_{t,ik}$ and S_{ik} refer to the contravariant velocity and area of the face between cells i and k. Linearizing Eq.(3.111) and recalling that zeroth-order terms are steady and first-order terms are harmonic in time gives, for the first-order system,

$$j\omega\Omega_{i,1} = \sum_{k=1}^{N(i)} V_{t,ik,1}\, S_{ik,0} \tag{3.112}$$

where ω is the frequency of the harmonic motion. This sets up a relationship between the first-order cell volumes, Ω_1, and the first-order face velocity, $V_{t,1}$, that must hold to satisfy GCL.

To ensure this relationship holds one can either calculate Ω_1 using the direct linearization method described above and then define $V_{t,1}$ based on Ω_1. Or, alternatively, $V_{t,1}$ may be found through direct linearization of the time derivative of the face centroid, and then define Ω_1 based on $V_{t,1}$. Either alternative will satisfy GCL. It was found that defining $V_{t,1}$ based on Ω_1 gave better results but required inverting a large system of equations.

3.3.5 Full Linear System

With the assumed harmonically perturbed forms for the state vector and node locations substituted into the full system from Eq.(3.12),

$$\frac{\partial \vec{Q}_i}{\partial t}\Omega_i = \vec{R}_i, \tag{3.113}$$

and with all other terms developed using the method outlined above, a full system can be formed with unknowns $\vec{Q}_0$ and $\vec{Q}_1$ and prescribed terms $\vec{x}_0$ and $\vec{x}_1$. The full system is written, for a select node i, as

$$\frac{\partial}{\partial t}\left[\vec{Q}_{i,0}\Omega_{i,0} + \epsilon\left(\vec{Q}_{i,0}\frac{\partial \Omega_i}{\partial \vec{x}}\vec{x}_1 + \vec{Q}_{i,1}\Omega_{i,0}\right)e^{j\omega t}\right] = \vec{R}_{i,0} + \epsilon\left(\frac{\partial \vec{R}_i}{\partial \vec{x}}\vec{x}_1 + \frac{\partial \vec{R}_i}{\partial \vec{Q}}\vec{Q}_1\right)e^{j\omega t} \tag{3.114}$$

where, for a general term ϕ,

$$\vec{\phi} = \left(\vec{\phi}_{i=1}, \vec{\phi}_{i=2}, \ldots, \vec{\phi}_{i=N}\right) \tag{3.115}$$

is a global concatenation of the local vectors from every node.

The full system may be split into a zeroth-order and a first-order system by grouping terms by their ϵ order. The zeroth-order system can then be written as

$$\frac{\partial}{\partial t}\left(\vec{Q}_{i,0}\Omega_{i,0}\right) = \vec{R}_{i,0} \tag{3.116}$$

and the first-order system written as

$$\left[j\omega\Omega_{i,0} - \frac{\partial \vec{R}_i}{\partial \vec{Q}}\right]\vec{Q}_1 = -\left[\vec{Q}_{i,0}\frac{\partial \vec{\Omega}_i}{\partial \vec{x}} - \frac{\partial \vec{R}_i}{\partial \vec{x}}\right]\vec{x}_1 \tag{3.117}$$

The steady, zeroth-order system exactly matches the full-order system. Therefore, procedures developed to solve the full-order system can be used to solve the zeroth-order system. The zeroth-order state vector, $\vec{Q}_0$, can be found using the pseudo-time marching method described in the temporal discretization section to solve steady problems.

With a known zeroth-order state vector, $\vec{Q}_0$ and prescribed zeroth and first-order node location vectors, $\vec{x}_0$ and $\vec{x}_1$, the unsteady, first-order system may be solved to obtain the first-order state vector, $\vec{Q}_1$. This system is a large, sparse, linear system with $10N$ degrees of freedom where N is the number of nodes in the mesh. Each node is associated with five conserved variables each consisting of a real and imaginary part. The system grows even larger when turbulence models are included.

The large size of the system precludes the use of many standard methods for solving linear systems due to time and computer memory constraints. Two methods are presented here that are able to handle these constraints. The first method is a pseudo-time marching method that is similar to the approach used to solve the full-order system. The second method uses the Generalized Minimal Residual Method (GMRES) based on a Krylov subspace approach which are well suited for sparse systems.

The first step in solving the unsteady system using the pseudo-time stepping approach is to add a pseudo-time gradient term. With this addition, the first-order system takes the form,

$$\frac{\partial}{\partial t_0}\left(\vec{Q}_{i,1}\right)\Omega_{i,0}\,e^{j\theta}+\left[j\omega\Omega_{i,0}-\frac{\partial\vec{R}_i}{\partial\vec{Q}}\right]\vec{Q}_1=-\left[\vec{Q}_{i,0}\frac{\partial\vec{\Omega}_i}{\partial\vec{x}}-\frac{\partial\vec{R}_i}{\partial\vec{x}}\right]\vec{x}_1 \qquad (3.118)$$

where t_0 is the pseudo-time and θ is a rotation angle that is be explained later in this section. The pseudo-time gradient is multiplied by the zeroth-order cell volume,

Ω_0, so as to match the true time gradient term of the full system. Eq.(3.118) is integrated forward in pseudo-time until a converged solution is reached. At this point the pseudo-time gradient goes to zero and is dropped and the original first-order system is recovered. The procedure to march the system in time closely resembles the procedure used for the full-order and zeroth-order systems, allowing previously developed computer code to be reused with minimal modification.

As seen in Eq.(3.118), the pseudo-time gradient is also multiplied by a rotation $e^{j\theta}$, that effectively phase shifts the gradient. This is used to fix a problem that can arise in cases dominated by the real-time gradient, $\frac{\partial \vec{Q}_1}{\partial t}$. The real time gradient, when applied to a harmonically oscillation, acts as a 90 degree phase shift. The iterative process used by the pseudo-time marching method can become very inefficient and even divergent because of this phase shift. Essentially, any update to the real part of the state vector, $\Re[\vec{Q}_1]$, is immediately shifted into the imaginary part and then, in the next iteration, shifted back again. This strongly couples the real and imaginary parts during the solution process even when they may not be strongly coupled in the final solution. A phase shift of angle θ may be imposed to counteract this issue. When θ is properly chosen, the solution process can be dramatically sped up and even some otherwise divergent cases will converge.

Currently, no quantitative analysis have been performed on selecting the optimal value for θ. The optimal value is case dependent and can range from about 5 degrees to around 85 degrees with higher frequencies tending to require higher θ values. It is left as a user input.

The second method used to solve the unsteady, first-order system is GMRES. An outline of the method can be found in [65]. GMRES can be very memory intensive. For each iteration essentially one full flow field worth of data must be stored. For large cases, this can severely restrict the minimum number of search directions. To get around this memory restriction, a restarted GMRES method is used. Several GMRES processes are run in series with the output of one becoming the initial guess of the next. In this way, the total memory requirement can be kept to a minimum. The issue that arises, however, is that the convergence of the solution may stall before ever reaching an adequate precision. It is therefore best to use as many search directions that the available memory will permit.

Once the first-order state vector, $\vec{Q}_1$, is found using either pseudo-time stepping or GMRES, it can be added to the previously found zeroth-order vector, $\vec{Q}_0$, to recover the full unsteady state vector, $\vec{Q}$.

3.3.6 Unsteady Mesh Generation

As discussed above, the Navier-Stokes equations are projected onto an unsteady domain. The unsteady domain, along with the projected Navier-Stokes equations, are discretized with an unsteady mesh. The unsteady mesh is made up of a steady, zeroth-order part, and a perturbed, unsteady first-order part. The zeroth-order mesh contains the locations and connectivity of the nodes within the undisturbed domain. The first-order mesh contains information describing how each individual node moves, perturbed from their initial position to account for the deformation of the domain. The two parts of the unsteady mesh are generated separately.

The steady, zeroth-order mesh is generated using an established, off-the-shelf mesh generating package.

The perturbed, first-order mesh must match the motion of the rotor and the stator at the boundaries of the domain and offer a smooth variation within the domain. The stator wall is stationary leading to a value of zero for the real and imaginary parts of x_1, y_1, and z_1 at the stator boundary. The motion of the rotor is dependent on the specific case. For an unbending rotor with a circular whirl orbit, for example, values of $x_1 = (0, 0)$, $y_1 = (1, 0)$, and $z_1 = (0, 1)$ must be enforced everywhere on the rotor surface. This assumes the rotor rotates about the x-axis. Note that x_1, y_1, and z_1 have been normalized with respect to the whirl amplitude.

Once x_1, y_1, and z_1 have been set to match the motion of the rotor and stator at the boundaries, a smooth variation can be achieved in the interior using an iterative relaxation method. First, an initial guess is made for x_1, y_1, and z_1 at every node with prescribed values on the rotor and stator surfaces. Then, iteratively looping through the nodes, $x_{i,1}$, $y_{i,1}$, and $z_{i,1}$ are updated according to,

$$\vec{x}_{i,1}^{k+1} = \vec{x}_{i,1}^{k} + \alpha \left(\sum_{j=1}^{N(i)} w_{ij}\, \vec{x}_{j,1}^{k} - \vec{x}_{i,1}^{k} \right) \tag{3.119}$$

where

$$w_{ij} = \frac{|\vec{x}_{i,0} - \vec{x}_{j,0}|}{\sum\limits_{k=1}^{N(i)} |\vec{x}_{i,0} - \vec{x}_{k,0}|} \tag{3.120}$$

The value for α is set between 0 and 1. Values on the stator and rotor have been prescribed and are not updated. Iterations continue until converged values for $x_{i,1}$, $y_{i,1}$, and $z_{i,1}$ are reached for every node.

3.3.7 Calculation of Rotordynamic Coefficients

The time-linearization methods discussed in this section were developed as part of a larger effort to predict rotordynamic stability characteristics of annular gas seals. This subsection focuses on this larger effort, developing the rotordynamic models as well as outlining the integration of these models with the RANS solvers. At the end of the subsection it should be clear how the time-linearized and full-order solvers are used to predict the stability characteristics of annular gas seals.

The development of the rotordynamic model begins by relating the aerodynamic force on the rotor with the rotor displacement. For a displaced rotor, the relationship

$$-\left\{\begin{array}{c} F_x \\ F_y \end{array}\right\} = \left[\begin{array}{cc} K & k \\ -k & K \end{array}\right]\left\{\begin{array}{c} X \\ Y \end{array}\right\} + \left[\begin{array}{cc} C & c \\ -c & C \end{array}\right]\left\{\begin{array}{c} \dot{X} \\ \dot{Y} \end{array}\right\} \tag{3.121}$$

is used where F_x and F_y are the forces on the rotor in the x and y directions, X and Y are the components of the rigid-body displacement of the rotor away for center, and K, k, C, and c are the rotordynamic coefficients. For gas seals, the added-mass coefficients are negligible [92] p. 292. With Eq.(3.121), the stability characteristics of the gas seals is determined entirely by the values for the rotordynamic coefficients. The rest of this subsection focuses on determining these values.

The displacement of the rotor whirling with a circular orbit can be described as

$$\begin{array}{ll} X = \Re\left[\epsilon\, e^{j\omega t}\right] & \dot{X} = \Re\left[j\omega\epsilon\, e^{j\omega t}\right] \\ Y = \Re\left[-j\epsilon\, e^{j\omega t}\right] & \dot{Y} = \Re\left[\omega\epsilon\, e^{j\omega t}\right] \end{array} \tag{3.122}$$

where ϵ is the whirl amplitude, and ω is the frequency.

The force on the rotor can also be written in radial and tangential components where the radial direction is aligned always with the displacement and the tangential direction is tangent to the orbit. This leads to the relationship

$$\left\{\begin{array}{c} F_x \\ F_y \end{array}\right\} = \left\{\begin{array}{c} F_r \\ F_t \end{array}\right\} \Re\left[e^{j\omega t}\right]. \tag{3.123}$$

The relationships in (3.121)-(3.123) should hold for all time. So, without loss of generality, time is set to zero. Substituting (3.122) and (3.123) into (3.121) gives

$$\begin{aligned} -\tfrac{F_x}{\epsilon} &= K + c\omega \\ -\tfrac{F_y}{\epsilon} &= -k + C\omega. \end{aligned} \tag{3.124}$$

The reactionary forces on the rotor are

$$\begin{aligned} F_x &= \int\limits_0^L \int\limits_0^{2\pi} \Re\left[(p_o + \epsilon\, p_1) \cdot n_x\right] d\theta\, dz \\ F_y &= \int\limits_0^L \int\limits_0^{2\pi} \Re\left[(p_o + \epsilon\, p_1) \cdot n_y\right] d\theta\, dz \end{aligned} \tag{3.125}$$

where z is the axial direction, L is the seal length, θ is the angle as measured from the positive X axis, p_0 and p_1 are the zeroth and first-order pressure and n_x and n_y are the components of the unit normal pointing in the X and Y directions respectively. Components of the unit normal can be written in terms of θ as $n_x = \cos\theta$ and $n_y = \sin\theta$.

Integrating the zeroth-order pressure around the annulus results in a zero net force on the rotor. The first-order pressure varies around the annulus as $p_1 = (p_r^* + j\, p_i^*)\; e^{j\theta}$ where p_r^* and p_i^* are the real and imaginary parts of the first-order pressure at $\theta = 0$. Eq.(3.125) can then be written as

$$\begin{aligned} \tfrac{F_x}{\epsilon} &= R\,\pi \int\limits_0^L p_r^*\, dz \\ \tfrac{F_y}{\epsilon} &= -R\,\pi \int\limits_0^L p_i^*\, dz \end{aligned} \tag{3.126}$$

where R is the radius of the rotor.

In Eq.(3.124), the forces F_x and F_y are functions of the whirl frequency ω. Because a single specified whirl frequency is given as an input to the time-linearized RANS solver, several different first-order flow fields must be found, each associated with a different specified whirl frequency, to establish the relationship between forces and the frequency. This suggests, for a given seal, the following procedure to calculate the rotordynamics coefficients.

- Full order RANS solver is used to find the steady zeroth-order flow field associated with a non-whirling (no displacement) rotor.
- Time-linearized RANS solver and zeroth-order results are used to calculate a first-order flow field for a prescribed whirl frequency, ω.
- Use Eq.(3.126) and first-order flow field to calculate the reactionary forces on the rotor.
- Repeat previous two steps for a range of whirl frequencies.
- With reactionary forces known for several whirl frequencies, rotordynamic coefficients, K, k, C and c, are backed out from Eq.(3.121).

A minimum of two whirl frequencies must be used to capture the coefficients, though more may be used for increased certainty. In this work, four frequencies were used. Whirl frequencies were chosen such that the whirl ratio remains around a value of one. The whirl ratio is defined as the ratio between the rotor whirl velocity and the rotor rotational velocity.

This procedure allows the rotor dynamic coefficients to be found, establishing the rotordynamic characteristics of the gas seal, by calculating a single, non-linear, steady, zeroth-order flow field and four linear, first-order flow fields. The four, first-order cases are decoupled from one another and may be run simultaneously. Extracting the reactionary forces and rotordynamic coefficients becomes a quick, post-processing step.

3.4 Boundary Conditions

This section presents boundary conditions for the full-order and time-linearized solvers. The boundary conditions presented include solid wall, inlet and outlet, symmetric plane, translational and rotational periodic, and non-reflective. The first subsection covers the implementation of the boundary conditions for the full-order code. The implementation for the time-linearized code is discussed in the next subsection. The last subsection is dedicated to the non-reflective boundary conditions for both full-order and time-linearized codes.

The boundary conditions for the inlet, outlet and solid walls are enforced in a weak sense. Instead of applying conditions directly to the state variables, intermediate state variables are introduced along the boundaries which are used to evaluate boundary fluxes. The intermediate variables are chosen such that the specified boundary conditions hold.

3.4.1 Full-Order Boundary Conditions

3.4.1.1 Wall Boundary Condition

For inviscid fluxes, the solid wall boundary condition enforces the no penetration conditions and holds that the component of the fluid velocity normal to the wall is equal to velocity of the wall itself. For the cases considered here for the full-order code, velocity in the wall normal direction is always equal to zero. The no penetration condition is then written as

$$\vec{v} \cdot \hat{n} = un_x + vn_y + wn_z = 0 \tag{3.127}$$

where $\hat{n}$ is the surface normal.

For viscous flows, in addition to no penertration, the no slip condition is enforced at walls. This holds that, for a stationary wall, components of the velocity are zero.

$$u = v = w = 0 \tag{3.128}$$

For the wall rotating around the x-axis, the components of the velocity are

$$\begin{aligned} u &= 0 \\ v &= -\omega R \cos\theta \\ w &= \omega R \sin\theta \end{aligned} \tag{3.129}$$

where ω is the angular velocity, R is the radius, and θ is the angle as measured from the z-axis.

3.4.1.2 Inlet and Outlet Boundary Conditions

The inlet and outlet boundary conditions, collectively referred to as farfield conditions, are enforced at the entrance and exit of the domain, respectively. It is at these locations that the computational domain is artificially truncated. In the physical system being modeled, the domain extends beyond these artificial boundaries. This leads to additional complexities when the boundary conditions are enforced. Ideally, the truncation of the domain should have no notable effect on the flow as compared to an infinite domain. An inadequate implementation can introduce spurious and unphysical waves into the system leading to a slowdown in convergence or an inaccurate final result. A later subsection will cover in more detail non-reflective boundary conditions which further eliminate truncation effects.

By studying the local characteristics of the system, it can be determined which direction information is flowing. This will determine which and how many conditions can be enforced at the inlet and outlet boundaries. The characteristics correspond to the eigenvalues of the local flux Jacobian matrix. A positive eigenvalue corresponds with a downstream velocity while a negative eigenvalue corresponds with upstream. By looking at the signs of the eigenvalues of the flux Jacobians at the inlet and outlet boundaries, it can be determined how many conditions must be enforced.

The inlet boundary for a subsonic flow has four positive eigenvalues and one negative. The characteristics associated with positive eigenvalues corresponds to information traveling downstream and into the domain. We are able to enforce as many conditions as there are incoming characteristics. In this case, four conditions can be imposed.

The subsonic outlet boundary also has four positive and one negative eigenvalue. This, however, corresponds to only a single incoming characteristic. Therefore, only one condition can be enforced at the outlet.

For supersonic flows, all of the characteristics flow downstream. This results in all five conditions being enforced at the inlet and no conditions enforced at the outlet.

A common procedure to define the four conditions enforced at the subsonics inlet is to specify the total pressure, p^*, total temperature, T^*, and two flow angles α and β. To specify the outgoing characteristic, a Riemann invariant[93, 94, 95], is used. The outgoing Riemann invariant is defined as

$$R^- = \vec{v}_d \cdot \hat{n}_b - \frac{2c_d}{\gamma - 1} \tag{3.130}$$

where $\vec{v}_d$ represents the flow velocity just interior of the inlet boundary, $\hat{n}_b$ represents the unit normal direction of the boundary and conditions taken from the interior of the domain. Here, c_d is the speed of sound. The total pressure and total temperature from the upstream conditions are used to compute the intermediate entropy

$$s = \frac{p^*}{(\rho^*)^\gamma} \tag{3.131}$$

and intermediate total enthalpy

$$H = \frac{\gamma R T^*}{\gamma - 1} \tag{3.132}$$

where ρ^* is defined as

$$\rho^* = \frac{p^*}{RT^*} \tag{3.133}$$

The incoming Riemann invariant, R^+, can be written as

$$R^+ = R^- + \frac{4}{\gamma - 1} c \tag{3.134}$$

where c is the intermediate speed of sound. To remove c from Eq.(3.134), it can be defined using the total enthalpy, component of velocity tangential to the boundary, and the incoming and outgoing Riemann invariants. Therefore, the intermediate speed of sound, c, is defined as,

$$c = (\gamma - 1)\sqrt{H - \frac{v_t^2}{2} - \frac{1}{8}\left(R^+ - R^-\right)} \tag{3.135}$$

where the tangential flow velocity, v_t is defined as

$$v_t = |\vec{v}_d| - \vec{v}_d \cdot \hat{n}_b. \tag{3.136}$$

Substituting Eq.(3.135) into Eq.(3.134) results in a quadratic equation for the incoming Riemann invariant, R^+. A solution for the quadratic equation can be written as

$$R^+ = \frac{1}{\gamma + 1}\left[(\gamma - 3)R^- + 4\sqrt{H - \frac{v_t^2}{2} - \frac{\gamma - 1}{2}\left(R^-\right)^2}\right]. \tag{3.137}$$

Once the incoming invariant, R^+, is known, the intermediate normal velocity, v_b, and the intermediate speed of sound c can be backed out using

$$v_b = \frac{R^+ + R^-}{2} \tag{3.138}$$

and

$$c = \frac{\gamma - 1}{4} \left(R^+ - R^- \right) . \tag{3.139}$$

The intermediate density, ρ and intermediate pressure, p, can be computed using the intermediate speed of sound, c, and the intermediate entropy, s, with

$$\rho = \left(\frac{c^2}{\gamma s} \right)^{\frac{1}{\gamma - 1}} \tag{3.140}$$

and

$$p = \frac{1}{\gamma} \left(\frac{c^{2\gamma}}{\gamma s} \right)^{\frac{1}{\gamma - 1}} . \tag{3.141}$$

Finally, the components of the intermediate velocity, $\vec{v} = (u, v, w)^T$, can be found using the prescribed inlet flow angles, α and β.

$$|\vec{v}| = \sqrt{v_b^2 + v_t^2} \tag{3.142}$$

$$u = |\vec{v}| \cos \alpha \tag{3.143}$$

$$v = |\vec{v}| \sin \alpha \cos \beta \tag{3.144}$$

$$w = |\vec{v}| \sin \alpha \sin \beta \tag{3.145}$$

The intermediate state variables are used to evaluate the flux across the inlet boundary faces.

The subsonic outflow boundary conditions are defined by a user specified static back pressure, p_b, and four conditions from the interior. Using the back pressure, p_b, and the entropy, s, tangential velocity, v_t, speed of sound, c, and total pressure, p^*, from the interior, the intermediate state variables can be defined.

The intermediate pressure, p, is set equal to the specified static back pressure,

$$p = p_b. \tag{3.146}$$

The intermediate entropy, s, and tangential velocity, v_t, are set equal to their interior counterparts,

$$s = s_i \tag{3.147}$$

and

$$v_t = |\vec{v}_i| - \vec{v}_i \cdot \hat{n}_b. \tag{3.148}$$

The incoming and outgoing Riemann invariants, R^- and R^+, are defined by

$$R^+ = \vec{v}_i \cdot \hat{n}_b + \frac{2c_i}{\gamma - 1} \tag{3.149}$$

and

$$R^- = R^+ - \frac{4}{\gamma - 1}\sqrt{\gamma\left(p^*\right)^{\frac{\gamma-1}{\gamma}} s^{\frac{1}{\gamma}}}. \tag{3.150}$$

The intermediate density, ρ, and velocity, $\vec{v} = (u, v, w)^T$, can then be computed using the incoming and outgoing Riemann invariants as shown in Eqs. (3.140) -

(3.145).

The intermediate state variables are used to evaluate the flux across the outlet boundary faces..

Flow conditions at the supersonic inlet boundary can be entirely specified based on prescribed freestream conditions.

$$\vec{Q} = \vec{Q}_b \tag{3.151}$$

Flow conditions at the supersonic outlet boundary are entirely defined by the flow conditions inside the domain.

$$\vec{Q} = \vec{Q}_i \tag{3.152}$$

3.4.1.3 Symmetric Boundary Condition

The symmetric boundary condition enforces flow symmetry with respect to the plane on which it is applied, eliminating the need to solve the full domain. Symmetry is enforced by setting the velocity component normal to the boundary face to zero, no penetration, and enforcing zero gradient of the state variables at the boundary and in the boundary normal direction.

Zero gradient at the boundary is enforced with the help of an adjacent interior node. This location of the interior node is specified by the mesh. The distance between the boundary and interior node is denoted by x_v. The direction $\hat{n}_v$ is defined pointing from the boundary node towards the interior node.

Taylor series expansion is used to find the relationship between the interior and boundary nodes. A general state variable, ϕ, and the component of its gradient in the $\hat{n}_v$ direction, $\frac{\partial \phi}{\partial n_v}$, are expanded about the boundary location towards the interior node.

$$\phi(x_v) = \phi_v = \phi_0 + \frac{\partial \phi_0}{\partial n_v} x_v + \frac{\partial^2 \phi_0}{\partial n_v^2} \frac{x_v^2}{2} + \cdots \tag{3.153}$$

$$\frac{\partial \phi}{\partial n_v}(x_v) = \frac{\partial \phi_v}{\partial n_v} = \frac{\partial \phi_0}{\partial n_v} + \frac{\partial^2 \phi_0}{\partial n_v^2} x_v + \cdots \tag{3.154}$$

Combining the two equations and neglecting higher order terms,

$$\phi_0 = \phi_v - \frac{1}{2}(\frac{\partial \phi_v}{\partial n_v} + \frac{\partial \phi_0}{\partial n_v}) x_v. \tag{3.155}$$

In the above equation x_v, ϕ_v and $\frac{\partial \phi_v}{\partial n} = \vec{\nabla}\phi_v \cdot \hat{n}$ are easily found. The value for $\frac{\partial \phi_0}{\partial n}$ is

$$\frac{\partial \phi_0}{\partial n_v} = \vec{\nabla}\phi_0 \cdot \hat{n_v} \tag{3.156}$$

where

$$\vec{\nabla}\phi_0 = \vec{\nabla}\phi - \frac{\partial \phi}{\partial n}\hat{n} \tag{3.157}$$

$$\frac{\partial \phi}{\partial n} = \vec{\nabla}\phi \cdot \hat{n}. \tag{3.158}$$

Here, $\hat{n}$ denotes the boundary surface normal direction and $\frac{\partial \phi}{\partial n}$ denotes the component of the gradient in the $\hat{n}$ direction. $\vec{\nabla}\phi_0$ is the gradient with the boundary normal components removed to satisfy the zero gradient criterion.

For implementation, however, $\frac{\partial \phi_0}{\partial n_v}$ is rewritten as

$$\frac{\partial \phi_0}{\partial n_v} = \vec{\nabla}\phi \cdot \hat{n}_v - \frac{\partial \phi}{\partial n}(\hat{n} \cdot \hat{n}_v) \tag{3.159}$$

and $\hat{n} \cdot \hat{n}_v$ is precomputed.

The zero gradient criterion is applied to all state variables independently using the method described. The gradients $\vec{\nabla}\phi$ are defined using the methods specified in

Using the newly calculated state variables, the velocity component of the flow normal to the boundary face, $\vec{v}_N$, is now removed.

$$\vec{v}_T = \vec{v} - |\vec{v}_N|\hat{n} \tag{3.160}$$

where

$$|\vec{v}_N| = \vec{v} \cdot \hat{n} \tag{3.161}$$

This enforces no flow through the boundary.

3.4.1.4 Periodic Boundary Conditions

The periodic boundary condition has been implemented for both translationally and rotationally periodic domains. For both types of periodic boundary conditions, a master boundary is paired with its respective slave boundary introducing new connectivity to the mesh. For a translational periodic boundary, a position on the slave boundary is related to a position on the master boundary using

$$\vec{x}_s = \vec{x}_m + \vec{x}_{ms} \tag{3.162}$$

where $\vec{x}_s$ and $\vec{x}_m$ are positions on the slave and master boundaries and $\vec{x}_{ms}$ is a vector that defines the translation from master to slave boundary. A position on a rotational slave boundary is related to a position on the master boundary using the following,

$$\vec{x}_s = \mathbf{T}_{ms}\vec{x}_m \tag{3.163}$$

where $\mathbf{T}_{ms}$ is a rotation matrix. Assuming the rotation is taken about the x-axis, $\mathbf{T}_{ms}$ takes the form,

$$\mathbf{T}_{ms} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{3.164}$$

where θ is the rotation angle.

For nodes on the periodic boundary, the following relationships hold for their respective state vectors, $\vec{Q}_i$. For translational periodicity, the state vector relationship is

$$\vec{Q}_s = \vec{Q}_m \tag{3.165}$$

where $\vec{Q}_s$ and $\vec{Q}_m$ are the state vectors associated with nodes on the slave and master boundaries respectively. For rotational periodicity, the state vector relationship is

$$\vec{Q}_s = \mathbf{U}_{ms}\vec{Q}_m \tag{3.166}$$

where

$$\mathbf{U}_{ms} = \begin{bmatrix} 1 & & \\ & \mathbf{T}_{ms} & \\ & & 1 \end{bmatrix}. \tag{3.167}$$

The components of velocity, u, v and w are rotated while density, ρ, and pressure, p are not.

In the present work, control volumes are defined using a dual mesh, with a node at the center of each control volume. The nodes on periodic boundaries are partnered with a node on the corresponding boundary and their state vectors are connected such that the partnered nodes are treated as essentially a single node. This naturally leads to a single control volume split in two along the periodic boundary with parts on either side. This concept is used when computing the residual for the master and slave nodes.

First, the residuals are computed for both master and slave nodes using the same procedure as is outlined in Chapter 3, by looping through the neighbors of the nodes and computing fluxes through each face. The only exception for periodic boundaries is that the flux through the periodic boundary is omitted. The residuals, therefore, are calculated using

$$\vec{R}_{im} = -\sum_{j=1}^{N_i, j \neq n_s} \vec{F}_{ij} S_{ij} \tag{3.168}$$

and

$$\vec{R}_{is} = -\sum_{j=1}^{N_i, j \neq n_m} \vec{F}_{ij} S_{ij} \tag{3.169}$$

where $\vec{R}_{i,s}$ and $\vec{R}_{i,m}$ are residual vectors associated with a slave and master node pair, F_{ij} and A_{ij} are the flux and area associated with the face between nodes i and j, and $(N_i, j \neq n_s)$ and $(N_i, j \neq n_m)$ represent the set of neighbors of node i excluding the corresponding slave and master nodes.

Next, the master and slave residual vectors are summed,

$$\vec{R}^*_{im} = \vec{R}_{im} + \vec{R}_{is} \tag{3.170}$$

to give the total residual vector, $\vec{R}^*_i$, associated with the recombined control volume spanning both sides of the periodic boundary.

The state vector for the master node, $\vec{Q}_{i,m}$, is then updated using the total residual vector,

$$\vec{Q}_{im}^{n+1} = \vec{Q}_{im}^{n} + \frac{\Delta t_{im}^{*}}{\Omega_{im}^{*}} \vec{R}_{im}^{*} \tag{3.171}$$

where $\Delta t_{i,m}^{*}$ and $\Omega_{i,m}^{*}$ are the time step and volume associated with the combined control volume. Finally, the state vector associated with the slave node, $\vec{Q}_{i,s}$, is updated using the relationships described earlier.

3.4.2 Time-Linearized Boundary Conditions

The previous subsection covered boundary conditions for the full-order Navier-Stokes solver. This subsection goes over how those boundary conditions used to be modified to be applied to the time-linearized code. Recall that the time-linearization method is based on the following linear approximation,

$$\phi = \phi_0 + \epsilon\, \phi_1 \tag{3.172}$$

where, for a given flow variable ϕ, ϕ_0 is the averaged value over a long period of time, and ϕ_1 is the perturbation about the average. The variable ϵ is a very small number representing the difference of time scale that the mean and perturbation effects operate at.

Substituting Eq.(3.172) into the boundary condition formulations in the previous subsection, and omitting terms of order ϵ^2 and higher, gives zeroth, ϵ^0, and first ϵ^1, order terms. The zeroth order terms correspond with the mean flow effects and match the full-order formulation perfectly. The first order terms correspond with the perturbation effects and are linear with respect to the state vector perturbation variables, ϕ_1.

3.4.2.1 Wall Boundary Condition

The formulation for the inviscid solid wall boundary condition says that $\vec{v} \cdot \hat{n} = 0$. Using Eq.(3.127) for the components of velocity u, v, and w and the geometric terms n_x, n_y, n_z, the components of the normal vector, results in

$$\begin{aligned} &(u_0 n_{x,0} + v_0 n_{y,0} + w_0 n_{z,0}) + \epsilon\,(u_0 n_{x,1} + v_0 n_{y,1} \\ &+ w_0 n_{z,1} + u_1 n_{x,0} + v_1 n_{y,0} + w_1 n_{z,0}) = 0. \end{aligned} \tag{3.173}$$

Recall that the mean variables are acquired from a full-order, steady simulation. Therefore, the zeroth order term is already satisfied leaving only the following,

$$u_0 n_{x,1} + v_0 n_{y,1} + w_0 n_{z,1} + u_1 n_{x,0} + v_1 n_{y,0} + w_1 n_{z,0} = 0. \tag{3.174}$$

The geometric terms, both mean and perturbed, are prescribed.

For the viscous solid wall, the velocity is equal to the velocity of the wall, $\vec{v}_w$. The mean part of the wall velocity is zero.

$$u_0 = v_0 = w_0 = 0. \tag{3.175}$$

This gives for the first order terms,

$$u_1 = u_{w,1},\ v_1 = v_{w,1},\ w_1 = w_{w,1} \tag{3.176}$$

where $\vec{v}_{w,1} = (u_{w,1}, v_{w,1}, w_{w,1})^T$ is the first order wall velocities associated, on the rotor, with the whirl motion.

3.4.2.2 Periodic Boundary Conditions

When using the time-linearized method to simulate the effects of rotor whirl, as is done in this work, an additional consideration must be made to the rotational periodic boundary condition. When dealing with the first-order terms, a phase shift must be included between terms on the master and slave boundaries.

The phase shift arises because the forcing on the system is being provided by the whirl of the rotor. To intuitively understand why this leads to a necessary phase shift across the periodic boundary consider the annulus geometry in Fig. 3.6. The inner cylinder is the rotor and it travels, displaced from a central position, harmonically in time, along a circular whirl orbit. This generates, at any given time, a maximum and minimum gap clearance. The location of the maximum and minimum clearance travels around the annulus along with the whirl of the rotor. The clearance at any given location along the annulus will vary harmonically in time, at some moments corresponding to the maximum clearance and at other moments the minimum. When comparing two different locations along the annulus with each other, their respective local clearances will appear to be phase shifted from one another. Because it is the whirling that drives the unsteadiness in the flow field, it can be reasoned the linearized flow field should also exhibit a similar phase shift.

This phase shift, $e^{j\theta}$, appears both in the relationship between the slave and master nodes locations

$$\vec{x}_{s,1} = \mathbf{T}_{ms}\, \vec{x}_{m,1}\, e^{j\theta} \tag{3.177}$$

and the slave and master state vectors,

$$\vec{Q}_{s,1} = \mathbf{U}_{ms}\, \vec{Q}_{m,1}\, e^{j\theta} \tag{3.178}$$

where θ is the angle of the rotational periodicity.

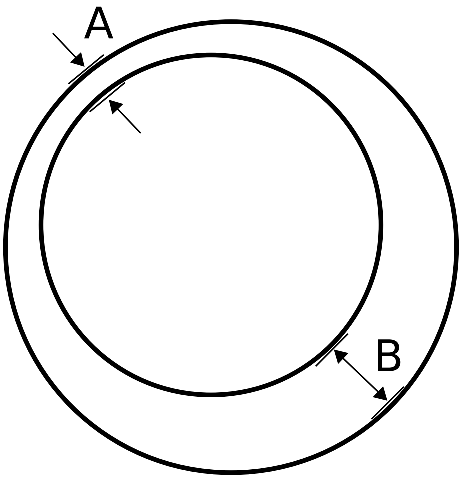


Figure 3.6: Clearance of an annulus geometry with whirling rotor.

3.4.3 Non-Reflective Boundary Conditions

Often, with CFD, it is the desire of the user to place the far-field boundaries as close in as possible. For internal flows, such as those found in this work, the placement of the far-field boundaries are specified by the test case and the user has no freedom to choose the location. In either of those cases, disturbances generated within the domain can reflect off of the far-field boundaries back towards the interior areas of the flow. These reflected waves are unphysical in nature and can interfere with the solution process and degrade the overall results. This subsection will detail methods used to limit or eliminate these reflected waves for both the full-order and time-linearized codes.

The reflective waves are generated by the far-field boundaries in an attempt to enforce the user defined conditions. As a disturbance waves, generated within the domain, exits the computational domain, it can change the flow conditions at the boundary resulting in the boundary condition to no longer be satisfied. To ensure specified conditions are enforced, a reflective wave, equal and opposite to the original disturbance wave, is generated and travels back into the domain. Because it is these reflective waves that enforce the boundary condition they are required to enforce a specified condition. The challenge is to modify the reflective waves in such a way that they are still able to enforce specified boundary conditions while minimizing their negative effects on the solution.

The theory used to enforce the non-reflective boundary conditions is developed by assuming that the flow is governed by the linearized Euler equations at the far-field. For this assumption to be valid the viscous shear stresses and turbulent effects need to be negligible. Also, physical variations in the far-field have long wavelengths as compared to the disturbance waves. The linearized Euler equation can be written in the following form,

$$\frac{\partial \vec{Q}_p}{\partial t} + \mathbf{A}\frac{\partial \vec{Q}_p}{\partial x} = 0 \tag{3.179}$$

where the perturbation state vector, $\vec{Q}_p$, is defined as

$$\vec{Q}_p = \begin{pmatrix} \rho - \bar{\rho} \\ u - \bar{u} \\ v - \bar{v} \\ w - \bar{w} \\ p - \bar{p} \end{pmatrix} = \begin{pmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{p} \end{pmatrix}. \tag{3.180}$$

The elements of $\vec{Q}_p$ represent linear perturbations, $(\tilde{\bullet})$, of the state variables away from average values, $(\bar{\bullet})$. The flux Jacobian matrix, $\mathbf{A}$, is written

$$\mathbf{A} = \begin{bmatrix} \bar{u} & \bar{\rho} & 0 & 0 & 0 \\ 0 & \bar{u} & 0 & 0 & \frac{1}{\bar{\rho}} \\ 0 & 0 & \bar{u} & 0 & 0 \\ 0 & 0 & 0 & \bar{u} & 0 \\ 0 & \gamma\bar{p} & 0 & 0 & \bar{u} \end{bmatrix} \tag{3.181}$$

and is defined by the average state variables. The average state variables define the state about which the Euler equations are linearized. Defining this average state is case dependent and will be addressed later in this subsection.

The numerical procedure introduced in earlier chapters operate on the conserved variables. It is useful, then, to introduce the linear transformation between conserved and primitive variables,

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} = \mathbf{L} \begin{pmatrix} \rho \\ u \\ v \\ w \\ E \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ \frac{V^2}{2} & \rho u & \rho v & \rho w & \frac{1}{\gamma-1} \end{bmatrix} \begin{pmatrix} \rho \\ u \\ v \\ w \\ E \end{pmatrix} \tag{3.182}$$

where

$$V^2 = u^2 + v^2 + w^2. \tag{3.183}$$

The behavior of the waves at the far-field can be analysed by assuming the following form,

$$\vec{Q}_p = \vec{q}_r e^{j(\omega t + k_x x)} \tag{3.184}$$

where $j = \sqrt{-1}$, ω is the frequency, and k_x is the spatial wave number to be determined. Substituting Eq.(3.184) into (3.179) gives

$$(\omega \mathbf{I} + k_x \mathbf{A})\vec{q_r} = 0. \tag{3.185}$$

The above equation can be rearranged to give an eigenvalue problem with eigenvalues k_x and corresponding eigenvectors $\vec{q_r}$. The eigenvalue problem is written

$$-\omega \mathbf{A}^{-1}\mathbf{I}\vec{q_r} = \mathbf{M}\vec{q_r} = k_x\,\vec{q_r}. \tag{3.186}$$

Solving for the wave number, k_x, gives a function in terms of ω. Matrix $\mathbf{M}$ is decoupled with the following transformation,

$$\mathbf{\Lambda} = \mathbf{T}^{-1}\mathbf{M}\mathbf{T} = \begin{bmatrix} -\frac{\omega}{\bar{u}} & 0 & 0 & 0 & 0 \\ 0 & -\frac{\omega}{\bar{u}} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\omega}{\bar{u}} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\omega}{\bar{u}+\bar{c}} & 0 \\ 0 & 0 & 0 & 0 & -\frac{\omega}{\bar{u}-\bar{c}} \end{bmatrix} \tag{3.187}$$

where $\mathbf{T}^{-1}$ and $\mathbf{T}$ are matrices made up of the left and right eigenvectors of $\mathbf{M}$, respectively, and $\bar{c} = \sqrt{\frac{\gamma p}{\rho}}$ is the local averaged speed of sound. Matrices $\mathbf{T}^{-1}$ and $\mathbf{T}$ define a linear transformation between the perturbation state vector, $\vec{Q}_p$, and the vector of characteristic variables, $\vec{\phi}$.

$$\vec{\phi} = \mathbf{T}^{-1}\vec{Q}_p \tag{3.188}$$

and

$$\vec{Q}_p = \mathbf{T}\vec{\phi}. \tag{3.189}$$

The characteristic variables, $\vec{\phi}$, can be written out as

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{bmatrix} -\bar{c}^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \bar{\rho}c & 0 & 0 \\ 0 & 0 & 0 & \bar{\rho}c & 0 \\ 0 & \bar{\rho}c & 0 & 0 & 1 \\ 0 & -\bar{\rho}c & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{p} \end{pmatrix} \tag{3.190}$$

and

$$\begin{pmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{p} \end{pmatrix} = \begin{bmatrix} -\frac{1}{\bar{c}^2} & 0 & 0 & \frac{1}{2\bar{c}^2} & \frac{1}{2\bar{c}^2} \\ 0 & 0 & 0 & \frac{1}{2\bar{\rho}c} & -\frac{1}{2\bar{\rho}c} \\ 0 & \frac{1}{\bar{\rho}c} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\bar{\rho}c} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix}. \tag{3.191}$$

The propagation speeds of the linearized characteristic variables are determined by finding their respective group velocities. The group velocity is given by

$$v_g = -\frac{\partial \omega}{\partial k_x} = -\frac{1}{\frac{\partial k_x}{\partial \omega}} \tag{3.192}$$

where the waves number, k_x, is the eigenvalue associated with the given characteristic variable and can be found on the diagonal of the matrix Λ.

Altogether, the five characteristic variables have the following physical interpretation. The first characteristic variable, ϕ_1, is defined as

$$\phi_1 = -\bar{c}^2\tilde{\rho} + \tilde{p} \tag{3.193}$$

and is associated with an entropy wave. The entropy waves propagate with speed $\bar{u}$. The second and third characteristic variables, ϕ_2 and ϕ_3, are defined as

$$\phi_2 = \bar{\rho}\bar{c}^2\tilde{v} \tag{3.194}$$

and

$$\phi_3 = \bar{\rho}\bar{c}^2\tilde{w} \tag{3.195}$$

and are associated with the components of a vorticity wave. The components of vorticity propagate with speed $\bar{u}$. The fourth and fifth characteristic variables, ϕ_4 and ϕ_5, are defined as

$$\phi_4 = \bar{\rho}\bar{c}^2\tilde{u} + \tilde{p} \tag{3.196}$$

and

$$\phi_5 = -\bar{\rho}\bar{c}^2\tilde{u} + \tilde{p} \tag{3.197}$$

are associated with downstream and upstream traveling pressure waves respectively. The downstream traveling pressure wave propagates with a speed $\bar{u}+\bar{c}$. The upstream traveling pressure wave propagates with a speed $\bar{u} - \bar{c}$.

3.4.3.1 Time-Linearized Non-Reflective Boundary Conditions

The characteristic theory developed above can be applied to the time-linearized system if the zeroth order terms, $\vec{Q}_0$, are used as the averaged variables, $(\bar{\bullet})$, and the first order terms, $\vec{Q}_1$, as the perturbation variables, $\tilde{(\bullet)}$. The characteristic variables can then be written as

$$\begin{pmatrix} \phi_{1,1} \\ \phi_{2,1} \\ \phi_{3,1} \\ \phi_{4,1} \\ \phi_{5,1} \end{pmatrix} = \begin{bmatrix} -c_0^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & (\rho c)_0 & 0 & 0 \\ 0 & 0 & 0 & (\rho c)_0 & 0 \\ 0 & (\rho c)_0 & 0 & 0 & 1 \\ 0 & -(\rho c)_0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \rho_1 \\ u_1 \\ v_1 \\ w_1 \\ p_1 \end{pmatrix} \tag{3.198}$$

with the flux Jacobian matrix containing exclusively zeroth order terms, consistent with the linear nature of the system.

It is then observed that characteristic variables $\phi_{1,1}$, $\phi_{2,1}$, $\phi_{3,1}$, and $\phi_{4,1}$ are propagating downstream while $\phi_{5,1}$ is propagating upstream. In order to enforce the non-reflective condition, no information is allowed to enter the domain and the downstream propagation characteristic variables must be removed at the inlet boundary and the upstream propagating characteristic variable must be removed at the outlet. This can be done by applying a filter at the far-field boundaries to be enforced at every iteration of the code, just after the respective boundary conditions. The filter takes the form of the following operation,

$$\vec{Q}_{1,f} = \mathbf{T}\mathbf{F}^*\mathbf{T}^{-1}\vec{Q}_1 \tag{3.199}$$

where first order state vector, $\vec{Q}_{1,f}$ represents the filtered variables. The matrix $\mathbf{F}^*$ is a diagonal matrix populated with zeros and ones such that the appropriate characteristic variables are removed. It is written as

$$\mathbf{F}^*_{\text{inlet}} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} \tag{3.200}$$

at the inlet and

$$\mathbf{F}^*_{\text{outlet}} = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 1 \end{bmatrix} \tag{3.201}$$

at the outlet.

With the filter applied at each node found on a far-field boundary, it can be insured that no reflected waves can enter the first-order, time linearized flow field.

3.4.3.2 Full-Order Non-Reflective Boundary Conditions

The characteristic theory can be applied to the full-order code by linearizing the Euler equations at each iteration about the current conditions. This makes the averaged, $(\bar{\bullet})$, terms from the characteristic theory associated with the current state vector, $\vec{Q}^n$. The perturbed variables, $(\tilde{\bullet})$, are associated with the vector $\Delta\vec{Q}^n$. This means the characteristic variables can be defined for the full-order system is

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{bmatrix} -c^2 & 0 & 0 & 0 & 1 \\ 0 & 0 & \rho c & 0 & 0 \\ 0 & 0 & 0 & \rho c & 0 \\ 0 & \rho c & 0 & 0 & 1 \\ 0 & -\rho c & 0 & 0 & 1 \end{bmatrix}^n \begin{pmatrix} \Delta\rho \\ \Delta u \\ \Delta v \\ \Delta w \\ \Delta p \end{pmatrix} \tag{3.202}$$

defined at every node.

Unlike the time-linearized code, were completely removed reflective waves is required, the full-order code requires some incoming waves to be able to enforce boundary conditions. Simply removing the incoming characteristic waves is not an option. This challenge is met by allowing some percentage of the energy of the incoming waves to remain. This has the effect of dampening reflections from spurious disturbances and lessening their detrimental effects while still allowing user specified boundary conditions to be enforced.

The filter has a similar form as before. It is written for the full-order system as

$$\Delta\vec{Q}_f = \mathbf{T}^n \mathbf{F}^* \mathbf{T}^{-1,n}\, \Delta\vec{Q} \tag{3.203}$$

where matrices $\mathbf{T}^{-1,n}$ and $\mathbf{T}^n$ are defined separately at every node i and every time step n. The filter matrix, $\mathbf{F}^*$, now takes on the form

$$\mathbf{F}^*_{\text{inlet}} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & \alpha \end{bmatrix} \tag{3.204}$$

at the inlet, and

$$\mathbf{F}^*_{\text{outlet}} = \begin{bmatrix} \alpha & & & & \\ & \alpha & & & \\ & & \alpha & & \\ & & & \alpha & \\ & & & & 1 \end{bmatrix} \tag{3.205}$$

at the outlet, where the coefficient α can be adjusted between zero and one. With α set to zero, no reflective waves are allowed, and with α set to one, the filter is effectively removed. It was found that a value of $\alpha = 0.1$ gives a good balance.

With this filter, unwanted effects from spurious disturbances can be mitigated while user defined boundary conditions can be enforced.

The non-reflective boundary, filter described above offers a computationally inexpensive and conceptually simple method to handle unwanted reflection found at far-field boundaries. For the turbomachinery seal cases in this work, the disturbances at the far-field boundaries were primarily planar in nature. The 1D characteristic theory described in this subsection was adequate for these cases. Some applications may require 2D or 3D characteristic theory to be effective. Development of these higher dimensional theories and their application to non-reflective boundary conditions can be found in [96, 97, 98].

4. RESULTS

4.1 Full-Order Solver: Cavity Flow Results

This section[1] presents the geometry and flow conditions used for the simulation of single and multiple cavity channel flows. A grid convergence test is then presented, followed by a comparison of the RANS and LES results. The numerical results are then used to determine the empirical values of the Rossiter's formula for cavities in channel flow. The section ends with results for multiple cavities.

4.1.1 Geometry and Flow Conditions

Computer simulations were run to provide a numerical model for experiments conducted at the Turbomachinery Laboratory at Texas A&M University. These experiments were performed to investigate the characteristics of cavity flow instabilities in annular seals. The hole-pattern of the annular stator seals was modeled using a flat plate with cylindrical cavities. To model the clearance between the rotor and stator a second flat plate with no cavities was located above the first. The experiments were conducted for several different cavity depths and channel heights. The numerical results shown herein had a channel height of 0.7112 mm and the cavities had diameters and depths equal to 3.175 mm.

For the numerical simulations the model for the hole-pattern seal was simplified to a single, rectangular, two-dimensional cavity with a length and depth of 3.175 mm. The cavity was located in a channel with a height of 0.7112 mm, which extended

[1] Part of this section is reprinted with permission from "Prediction of Aeroacoustic Resonance in Cavities of Hole-Pattern Stator Seals" by D. N. Liliedahl, F. L. Carpenter, P. G. A. Cizmas, 2010. *J. Eng. Gas Turbines Power*, vol. 133(2), p.022504, Copyright 2011 by American Society of Mechanical Engineers.

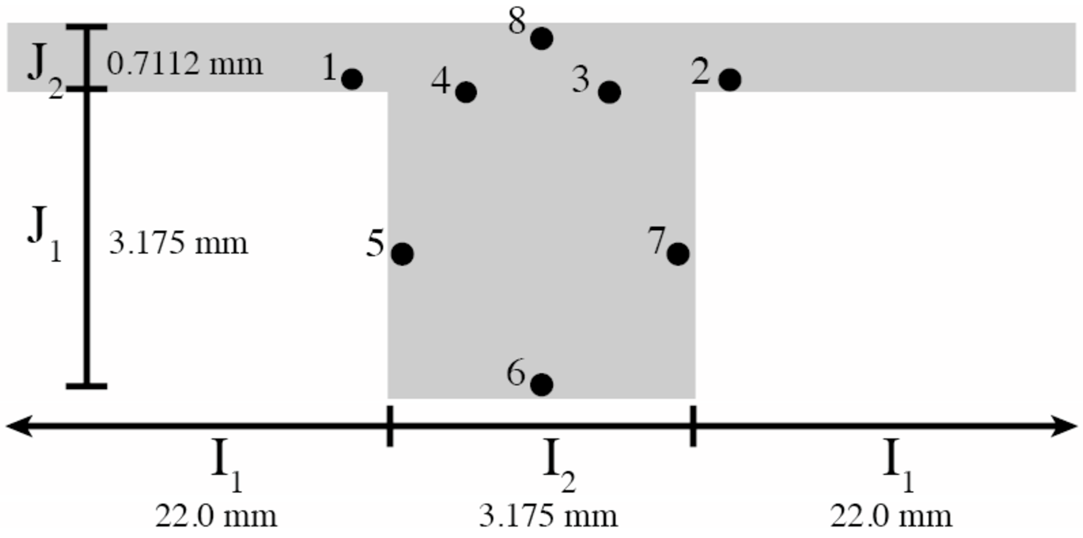


Figure 4.1: Computational domain with pressure probe locations.

22 mm (6.9 cavity lengths) in front of and behind the cavity. This allowed for dissipation of spurious waves that reflect from the inlet and outlet boundaries and prevented them from interfering with the cavity flow. The computational domain is shown in Fig. 4.1. For clarity, the extremities of the channel are not shown. As shown in the previous section, the flow solver is three-dimensional. To reduce the computational time, the width of the computational domain was limited to one cell. This simplified domain allowed for reasonable computation times while still capturing the necessary flow features.

The inlet total pressure was 104,190 Pa, the exit static pressure was 101,325 Pa and the total temperature was 305.4 K. The Reynolds number based on the cavity length was 14,300. Each simulation was started from an initial condition with a uniform velocity of 68 m/s parallel to the channel. These parameters were chosen to provide a flow with a Mach number of approximately 0.2.

In the experiments, the flow through the channel was fully developed by the time it reached the cavities. To ensure the numerical simulation modeled the proper flow conditions, a fully developed flow was specified at the inlet boundary of the computational domain. It was found that numerical simulations using the fully developed condition predicted a dominant frequency approximately 1.4 kHz higher than using a uniform flow inlet condition. The fully developed inlet boundary condition was used for the remainder of the study.

4.1.2 Grid Convergence

Although the flow solver used herein was unstructured, the domain was discretized using an entirely structured grid. Cells were clustered near the boundaries and along the top of the cavities to capture the larger gradients present in the boundary layers and shear layer. The cell size along the walls was specified such that the y^+ number was less than 1.4. The growth rate of the cell height in the direction normal to the wall was 1.3.

Table 4.1: Details of grids used in convergence test.

	Grid 1	Grid 2	Grid 3
I_1	45	90	180
I_2	60	120	240
J_1	25	50	100
J_2	40	80	160
Total Cells	5766	23826	96846
Total Nodes	11952	48502	195402

A grid convergence test was carried out to ensure that the flow solution was independent of the grid size. Grid independence was determined by comparing the predicted lock-in (or dominant) frequencies. The dominant frequency was obtained by measuring the pressure time history at several probe locations in the domain and then performing a Fourier transform to determine the peak frequency of oscillation. The probe locations are shown in Fig. 4.1.

The grid convergence test was conducted using three grids shown in Table 4.1. For each subsequent grid, the number of nodes along each boundary was doubled. Variables I_1, I_2, J_1 and J_2 denote the number of nodes corresponding to the locations shown in Fig. 4.1. Figure 4.2 shows a plot of the dominant frequency versus total number of nodes. As the number of nodes increases and the grid is refined, the dominant frequency converges to a value near the value recorded from the experiment. Table 4.2 shows a summary of the results. Grid 2, shown in Fig. 4.3, was considered a good compromise between solution accuracy and computational cost and was used for the remainder of the study.

It should be noted that the experimental investigation, which had several hundred cavities, showed two dominant frequencies at Mach 0.17: 15.62 and 21.75 kHz. These dominant frequencies corresponded to the second and third mode, m, in Rossiter's formula (1.2).

Table 4.2: Dominant frequencies in experiment and using RANS solver.

	Dominant Frequency [kHz]
Experiment	21.75
Grid 1	16.22
Grid 2	20.95
Grid 3	22.76

4.1.3 RANS vs. LES

To verify the RANS model, large-eddy simulations (LES) were performed using similar mesh and boundary conditions. These LES computations were performed

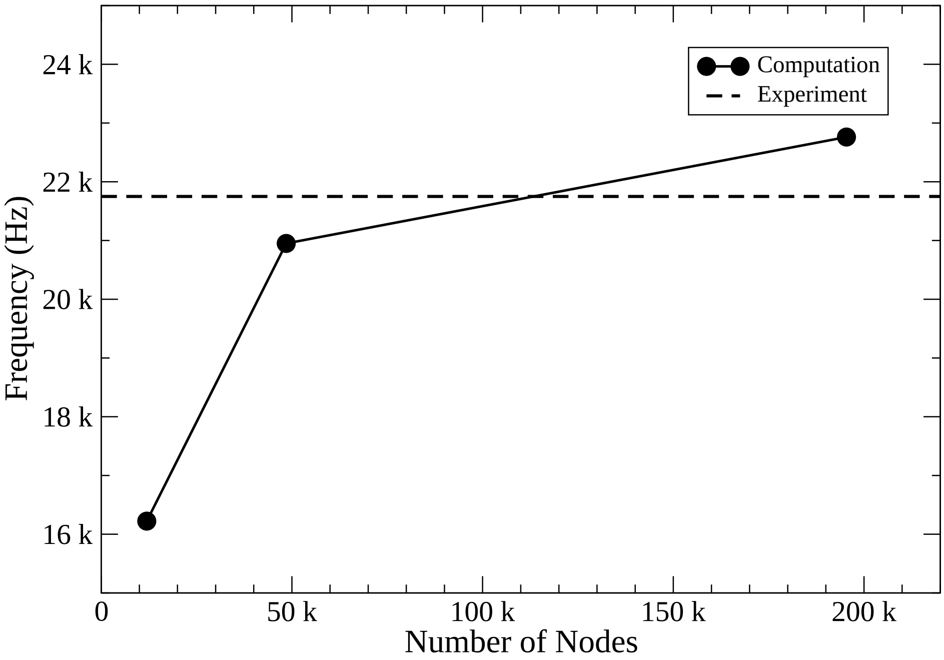


Figure 4.2: Grid convergence: dominant frequency vs. total number of nodes.

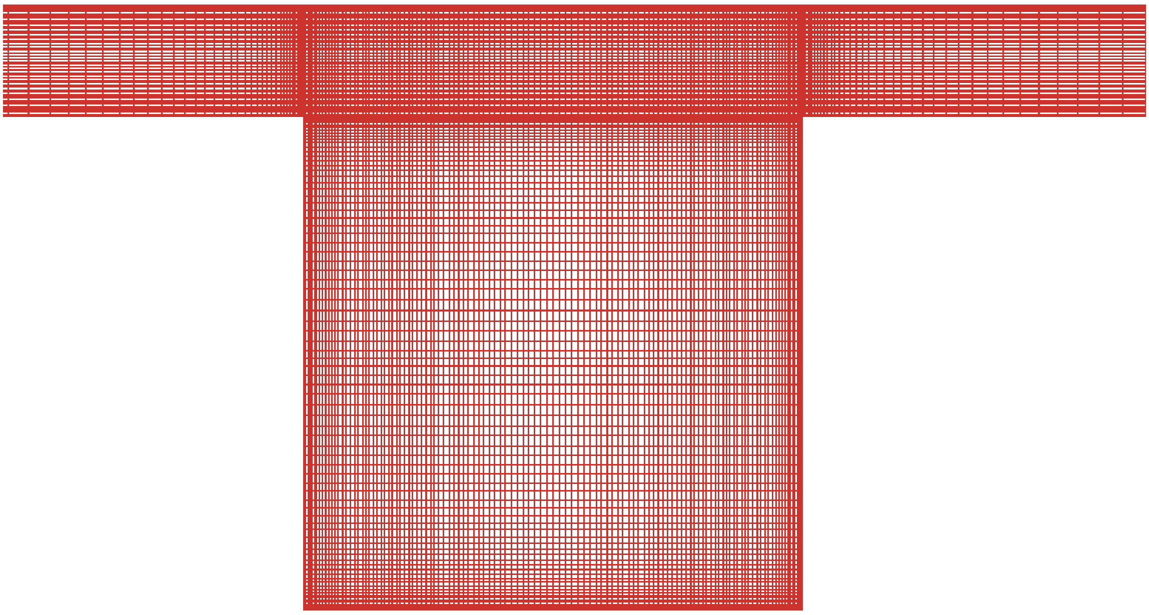
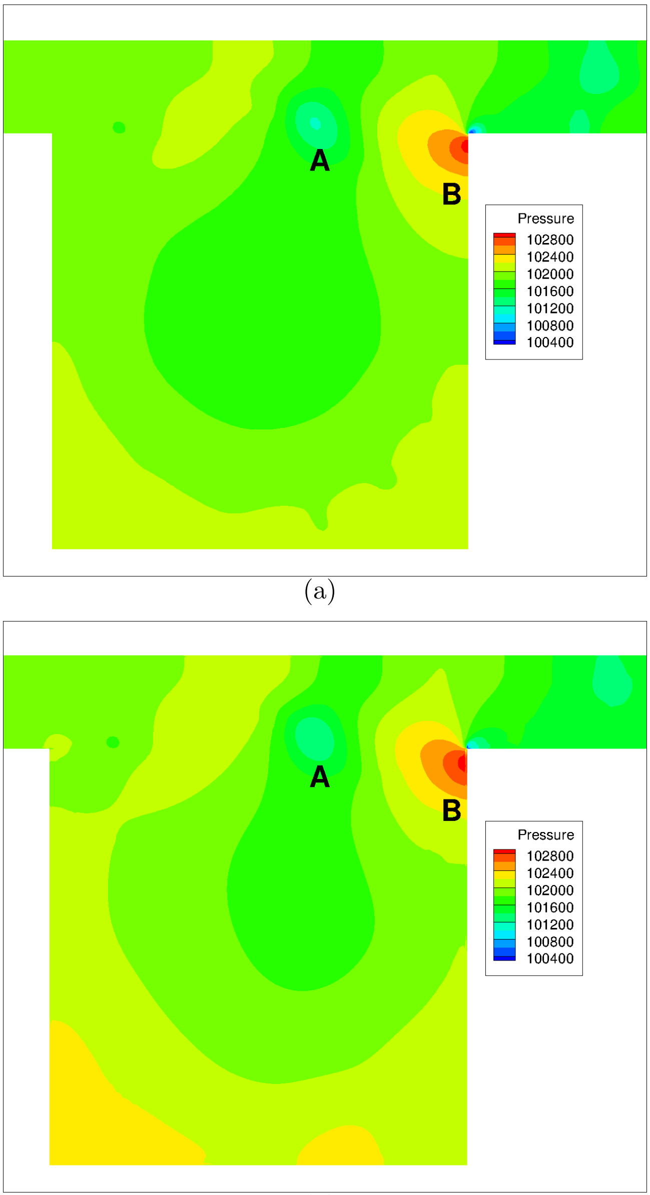


Figure 4.3: Grid 2 - detail of the cavity region.

using the commercial code FLUENT, Version 6.3. A second-order implicit scheme was used with a least-squares cell based method to calculate the gradients. Bounded central differencing was used to discretize the governing equations. The sub-grid scale turbulent viscosity was modeled using the dynamic kinetic energy subgrid-scale model [99]. The pressure, momentum and sub-grid kinetic energy variables were relaxed using the factors 0.3, 0.7, and 0.8, respectively. To capture the physics of the cavity flow, the simulations were run using a time step of 10^{-7} seconds.

Figure 4.4 shows the pressure contours generated by the RANS and LES solvers for a flow with a Mach number of 0.17 at the leading edge of the cavity. The vortex (A) located at approximately 60% of the cavity length rises the same distance above the cavity opening using both simulation methods. The dipole (B) which is formed at the trailing lip of the cavity is shown to be in very nearly the same phase. This shows that LES and RANS simulations qualitatively agree well and capture the same



(a)

(b)

Figure 4.4: Pressure contours with vortex A and dipole B: (a) LES (b) RANS

salient features of the flow.

Velocity profiles are shown in Fig. 4.5 for five evenly spaced locations through the cavity. These profiles are for the same flow and time as the snapshots in Fig. 4.4. The velocity profiles produced with the LES and RANS simulations are nearly identical except near the bottom of the cavity.

Figure 4.6 shows a comparison of the pressure time history for point 3 as indicated in Fig. 4.1. The wavelengths of the periodic signal produced by the LES and RANS methods are nearly identical and the amplitudes match well. The frequency distribution is shown in Fig. 4.7. Both plots show a dominant frequency just above 20 kHz as well as a strong harmonic frequency above 40 kHz. The dominant frequency of the LES simulation was 21.80 kHz while the RANS simulation was 20.94 kHz. This shows good quantitative agreement between the two models.

Numerical simulations using RANS and LES solvers produce very similar results. However, the computational time required by the RANS solver was more than one order of magnitude smaller than that of the LES solver. As a result, the RANS solver was used for the remainder of the study.

The RANS simulations were run on a 3 GHz Quad-Core Intel Xeon processor. The CPU time required was approximately $7{\times}10^{-5}$ seconds per node per iteration while using the second-order accurate scheme. The simulations were each run for approximately 120k iterations using a CFL number of 3.5. The average residuals were all less than approximately 10^{-5}.

4.1.4 Single Cavity Channel Flow

Figure 4.8 shows a series of pressure contours illustrating a complete cycle of the cavity flow mechanism described above. A vortex is shed from the leading edge of the cavity and increases in magnitude as it propagates along the shear layer. As

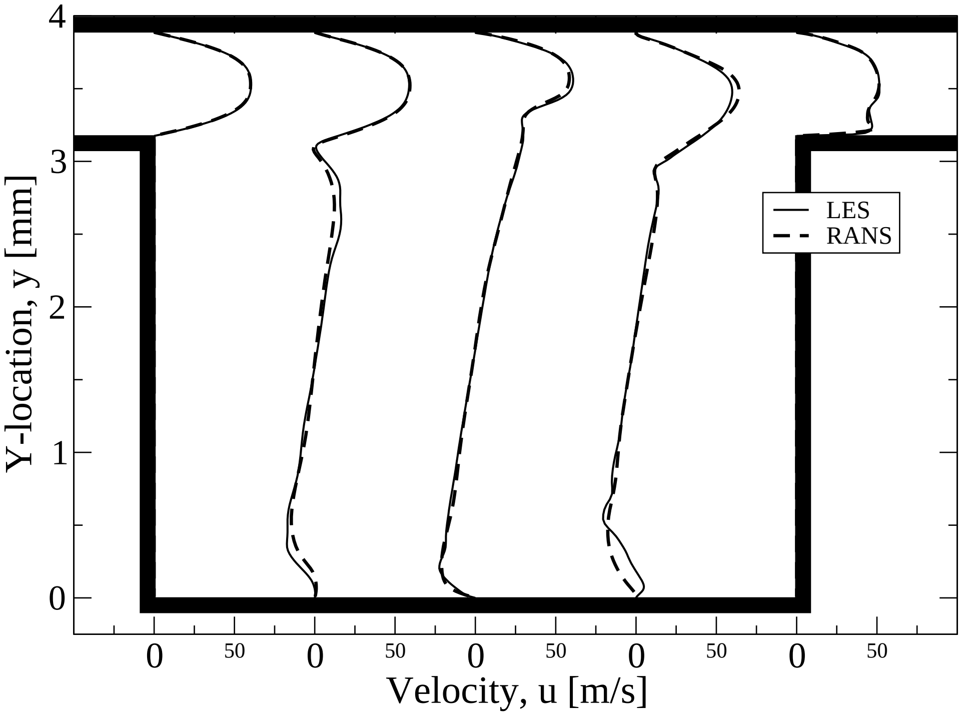


Figure 4.5: Velocity profiles generated with LES and RANS.

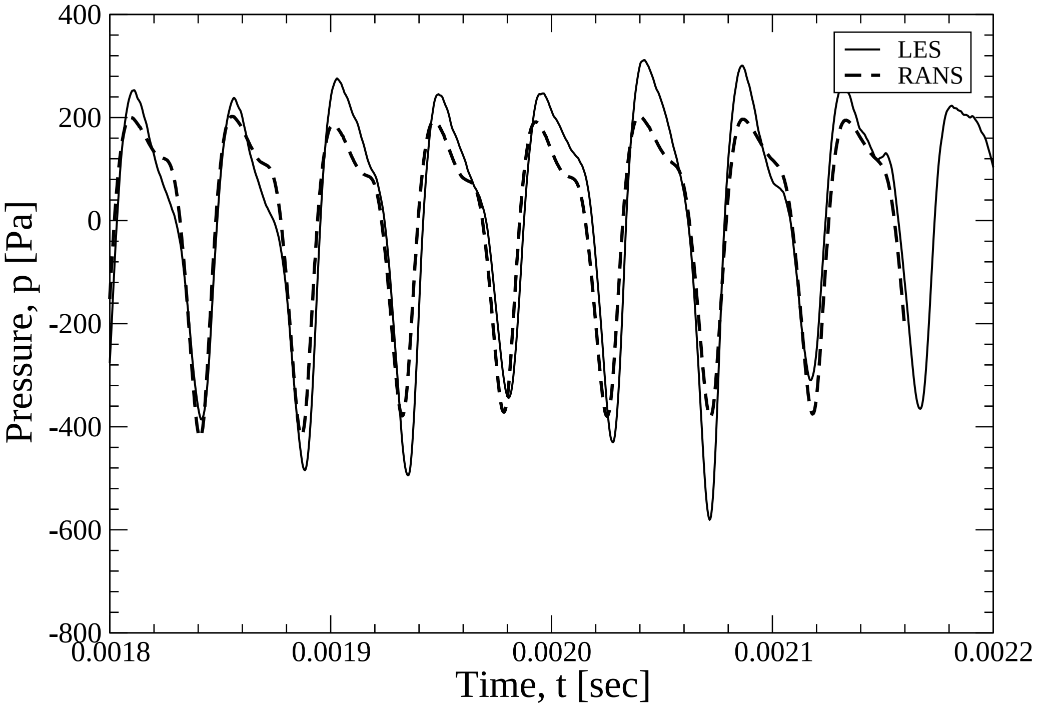


Figure 4.6: Pressure time history generated with LES and RANS.

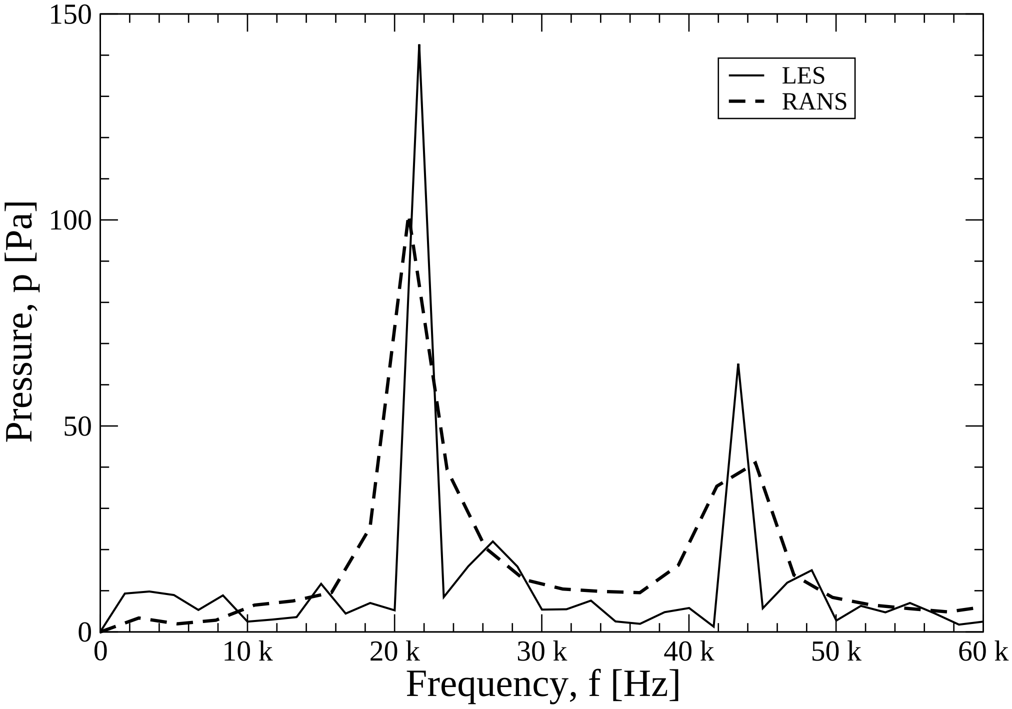


Figure 4.7: Pressure variation vs. frequency generated with LES and RANS.

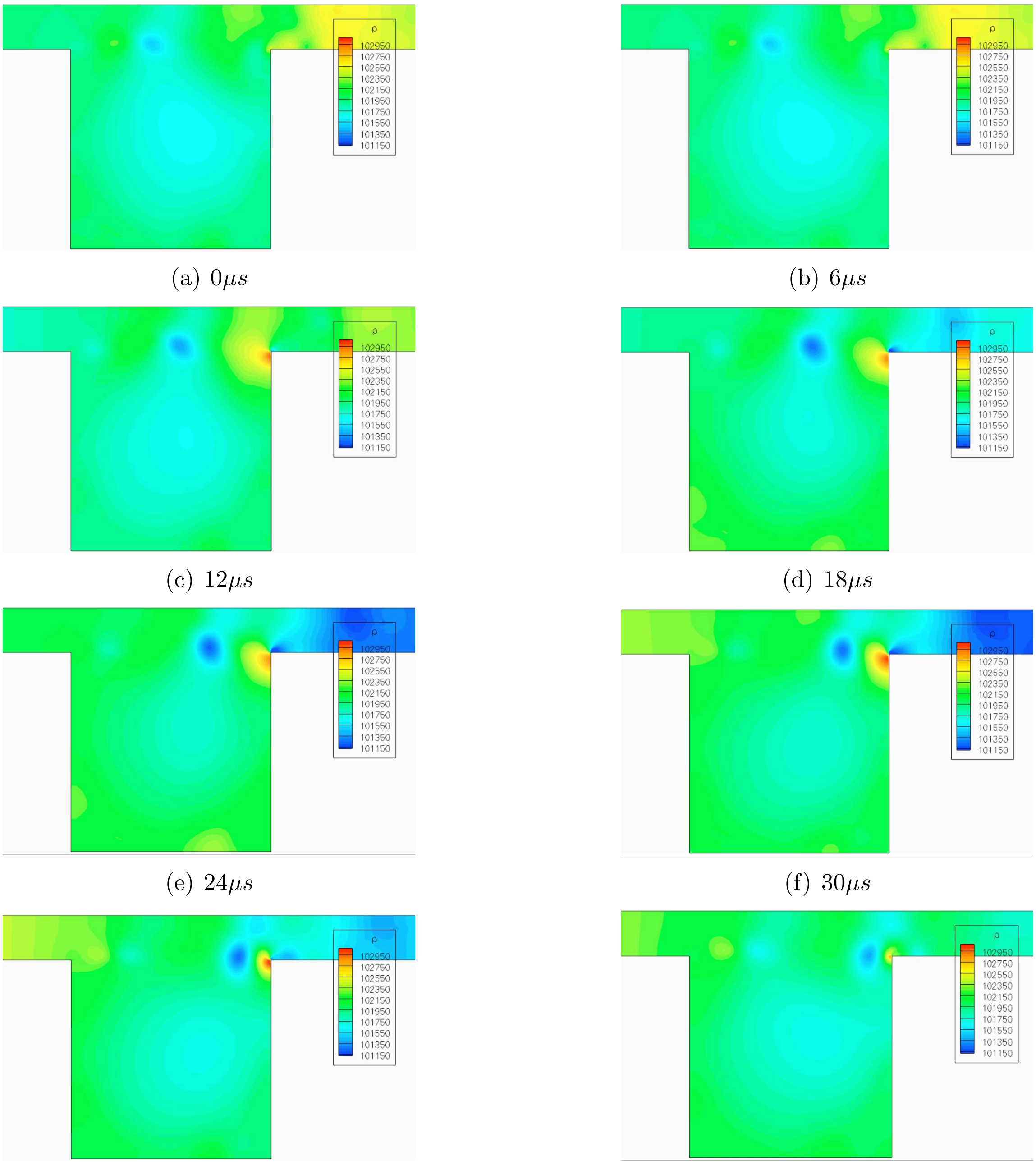
the vortex approaches the trailing edge of the cavity, a dipole is formed releasing an acoustic wave. As the original vortex impinges on the trailing edge, a second vortex is shed. Figure 4.9 shows the pressure contour for a moment in the feedback cycle where both the vortex and dipole are easily observable. The vorticity distribution for the same moment is shown in Fig. 4.10. The influence of the vortex on the shear layer is apparent.

Simulations using the flow conditions described above, for a cavity with a length and depth of 3.175 mm and a channel height of 0.7112 mm, predicted a dominant frequency of 20.95 kHz. This compares well with the experimental result of 21.75 kHz. This shows that, even though the flow domain was significantly simplified, the numerical simulations were able to accurately capture the important flow features and their interactions, and to accurately predict the dominant frequency.

4.1.5 Determination of Empirical Values

In this study it was found that Rossiter's formula (1.2) as reported in literature [13, 27] was not able to accurately predict dominant frequencies. It was believed that the error was due to the influence of the opposing wall of the channel. This section presents a modification to Rossiter's formula to allow for prediction of dominant frequencies for cavity flow within a channel. Numerical simulations showed that the feed-back mechanism for cavity channel flows was the same as that used to derive Rossiter's formula. For this reason the form of Rossiter's formula remained valid while only the value of one empirical variable was modified.

The empirical term γ can be physically understood as a phase shift between the shear layer oscillations and the acoustic waves. In past studies, the value has consistently been found to be 0.25. The value of $\gamma = 0.25$ was also found herein to accurately predict dominant frequencies for cavities in a channel. Therefore, the



(a) $0\mu s$

(b) $6\mu s$

(c) $12\mu s$

(d) $18\mu s$

(e) $24\mu s$

(f) $30\mu s$

(g) $36\mu s$

(h) $42\mu s$

Figure 4.8: Pressure contours at various times within a cycle.

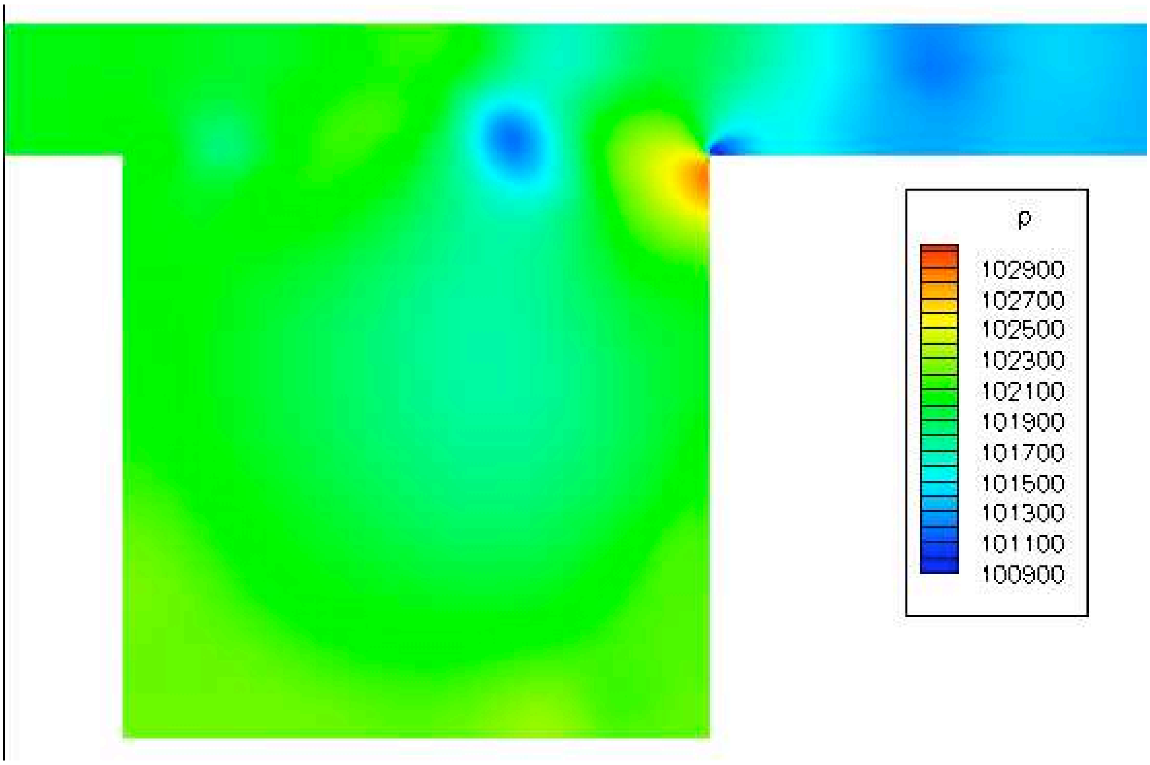


Figure 4.9: Snapshot of pressure contours for channel cavity flow.

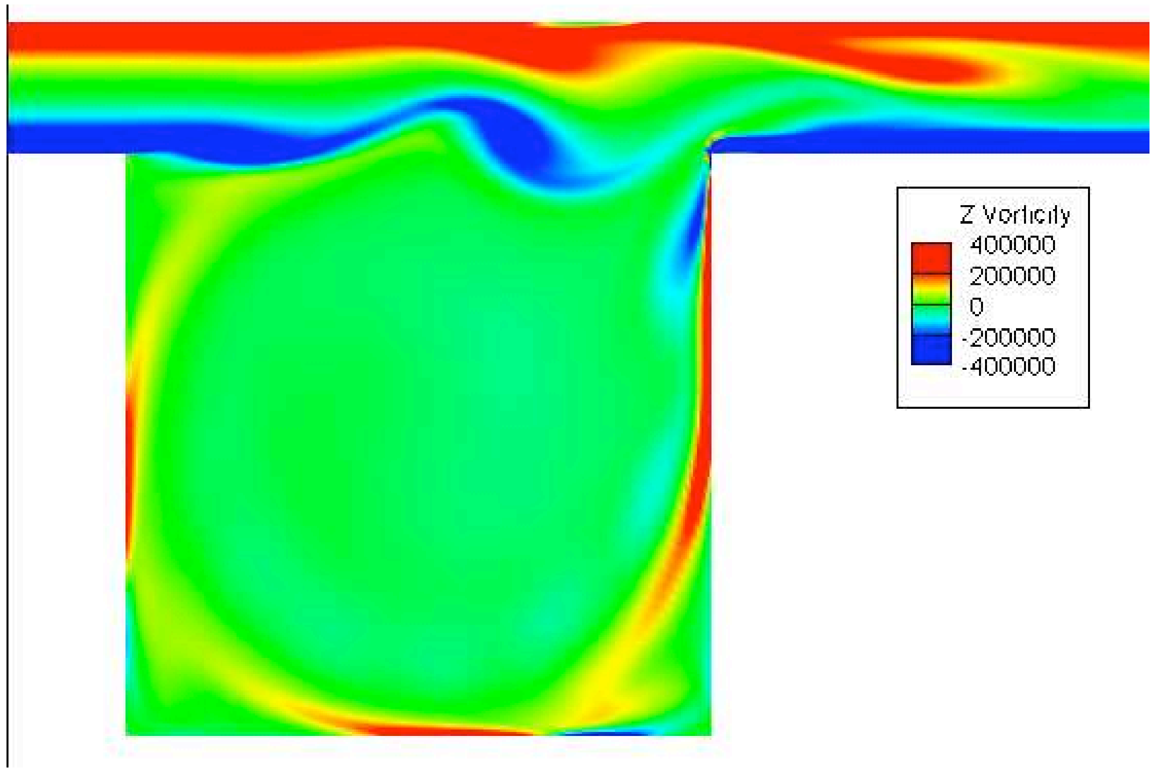


Figure 4.10: Snapshot of vorticity contours for channel cavity flow.

value for γ was not modified.

The empirical term κ can be physically understood as the ratio between the propagation speed of the vortices and the freestream velocity. Originally, Rossiter found the value for κ by fitting the equation to experimental results. Using CFD, however, the position of the vortices can be directly tracked throughout a given cycle. The velocity is then easily calculated giving the κ ratio. Reported κ values for open cavity flow range from 0.57 by Larcheveque *et al.* [27] to 0.66 by Rossiter [13]. Using the technique described above, the authors found the κ ratio to be 0.52.

Rossiter's formula assumes that the velocity of the vortices remains constant throughout the length of the cavity. Simulation results clearly shows this is an approximation of a more complicated phenomenon. Figure 4.11 shows the velocity of a vortex with respect to its position within the cavity. The velocity is normalized against the free stream velocity and the position is normalized against the cavity length. The vortex begins with a velocity near zero before it is shed from the leading edge of the cavity. As the vortex convects downstream, its normalized velocity increases to a peak value of around 0.63 near 70% of the cavity length. The vortex then slows down before impinging on the trailing edge of the cavity. Averaging the normalized vortex velocity across the length of the cavity yields a κ value of 0.52.

Figure 4.12 shows pressure variation as a function of frequency for two Mach numbers at the cavity leading edge, 0.166 and 0.183. The pressure variation was calculated using the RANS solver results at the points 2, 3 and 4 shown in Fig. 4.1. All points predict the same dominant frequency: 23.68 kHz for Mach 0.166 and 26.32 kHz for Mach 0.183. The dominant frequency was also calculated with Rossiter's formula (1.2) using the third mode, m=3. Two values of κ constant were used: Rossiter's κ=0.66 and the value calculated herein for channel flows, κ=0.52. Figure 4.12 shows a good match between the dominant frequency predicted by the RANS

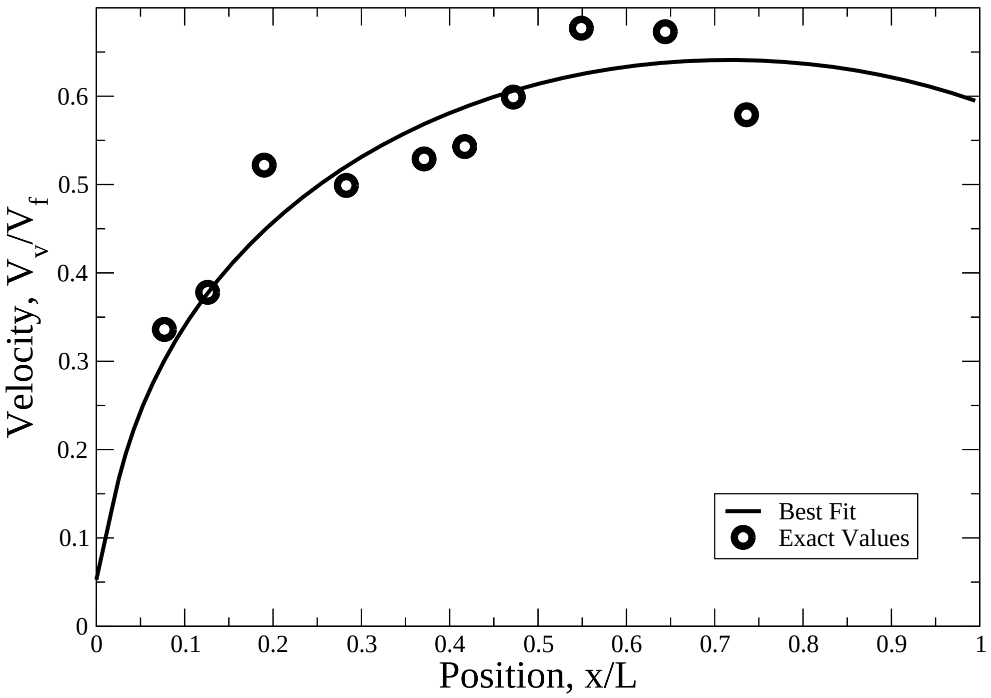


Figure 4.11: Vortex convection velocity as a function of cavity location.

solver and Rossiter's formula with κ=0.52, for both Mach numbers. A summary of the predicted dominant frequencies is shown in Table 4.3.

Table 4.3: Predicted dominant frequencies: RANS and Rossiter's formula.

Mach	RANS	Rossiter's formula (1.2)	
		κ=0.52	κ=0.66
0.166	23.68 kHz	23.90 kHz	29.70 kHz
0.183	26.32 kHz	26.14 kHz	32.42 kHz

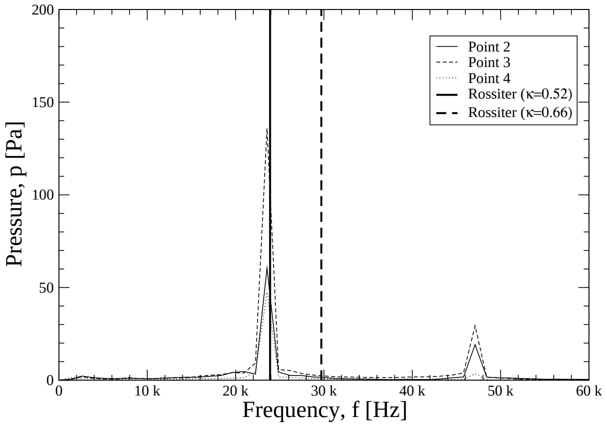
4.1.6 Multiple Cavity Channel Flow

The hole-pattern stator seal used in the experiments contained several hundred cavities. To investigate the possible interactions between neighboring cavities, simulations were run using configurations with three, five, and seven cavities. Each cavity had a length and a depth of 3.175 mm. The channel height was 0.7112 mm. The distances between cavities were varied between 0.5 and 1.0 mm to assess the cavity spacing effect on the two-dimensional results [100].

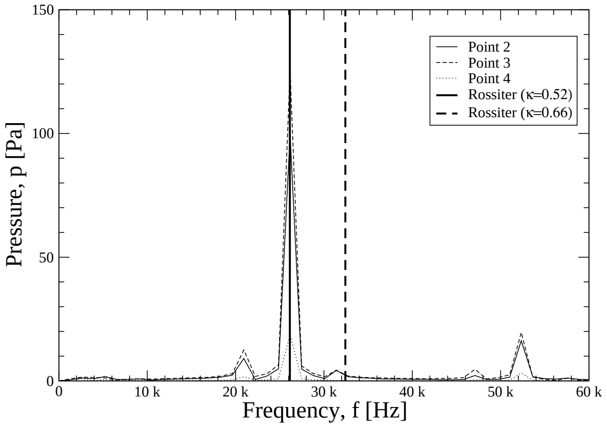
The computational domain of the multiple cavities repeated the grid generated for the single cavity 3, 5, and 7 times. There were 30 nodes added between the cavities, referred to as I_3 and shown in Fig. 4.13. A summary of the grid sizes for the multiple cavity domains is shown in Table 4.4.

Table 4.4: Grid sizes for multiple cavity configurations.

Cavities	3	5	7
Total Cells	59718	95610	131502
Total Nodes	121306	194110	266914



(a) M=0.166



(b) M=0.183

Figure 4.12: Pressure variation vs. frequency.

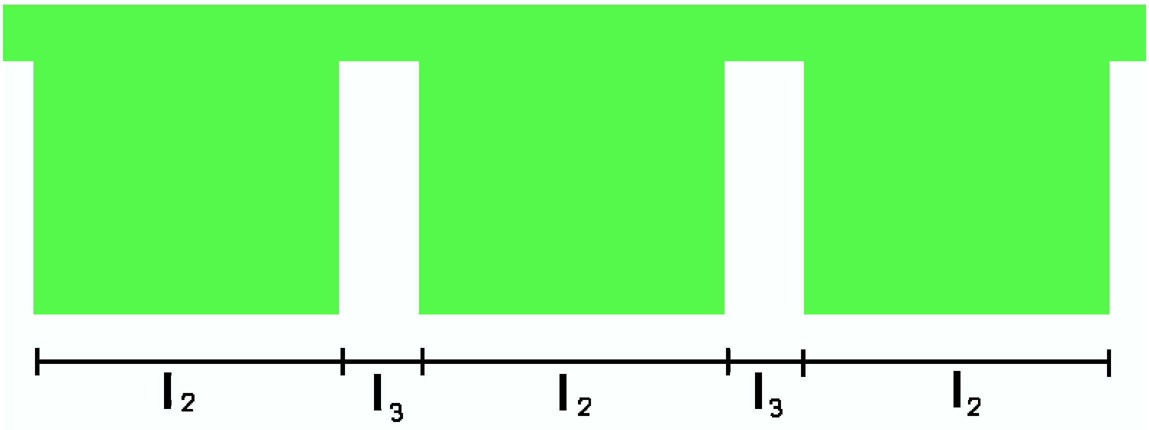


Figure 4.13: Computational domain for multiple cavity configuration.

Figures 4.14, 4.15, 4.16, and 4.17 show the pressure variation vs. frequency for a single cavity and for multiple cavities with a 0.5 mm gap. The dominant frequency for the single cavity is 15.77 kHz, with no other significant frequencies present. The dominant frequency for the multiple cavities with 0.5 mm gap remains approximately 15.77 kHz, while higher modes of the dominant frequency are introduced near 45 and 60 kHz. The higher modes become more powerful as more cavities are added. Figure 4.18 shows the frequency distribution of a geometry with five cavities that had a gap of 1.0 mm. The dominant frequency remains near 15.60 kHz but higher modes are no longer present.

The numerical results generated by the RANS and LES solvers properly captured the dominant frequencies of the cavity channel flow. The agreement between the RANS and LES solvers was also proven for the velocity and pressure variation. For a single cavity channel flow the RANS solver predicted 15.77 and 20.95 kHz for the dominant frequencies, the LES solver predicted 15.11 and 21.8 kHz, while the experimental data reported 15.62 and 21.75 kHz. It should be noted that the stator

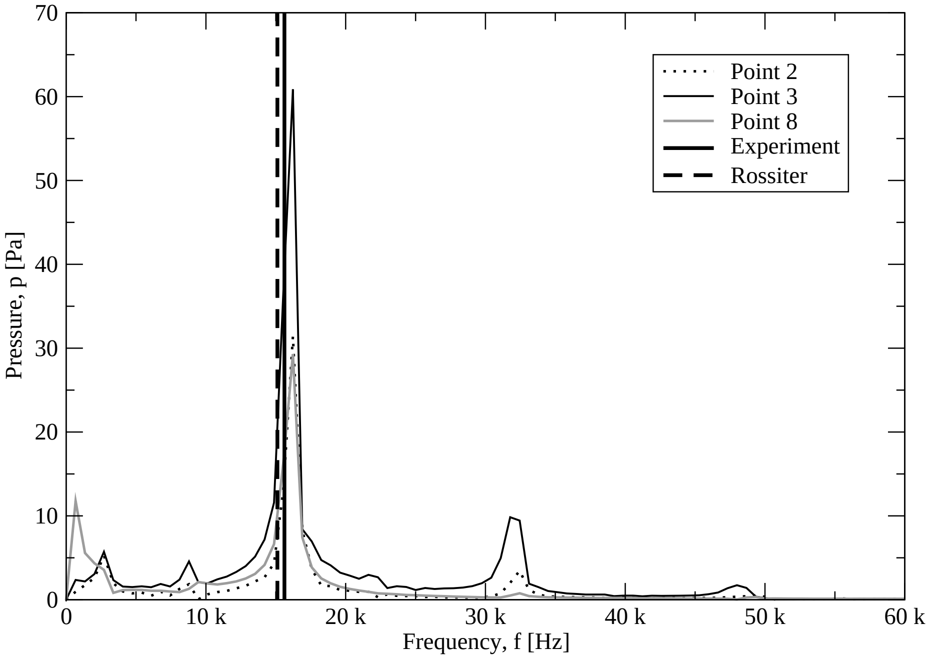


Figure 4.14: Frequency distributions for single cavity.

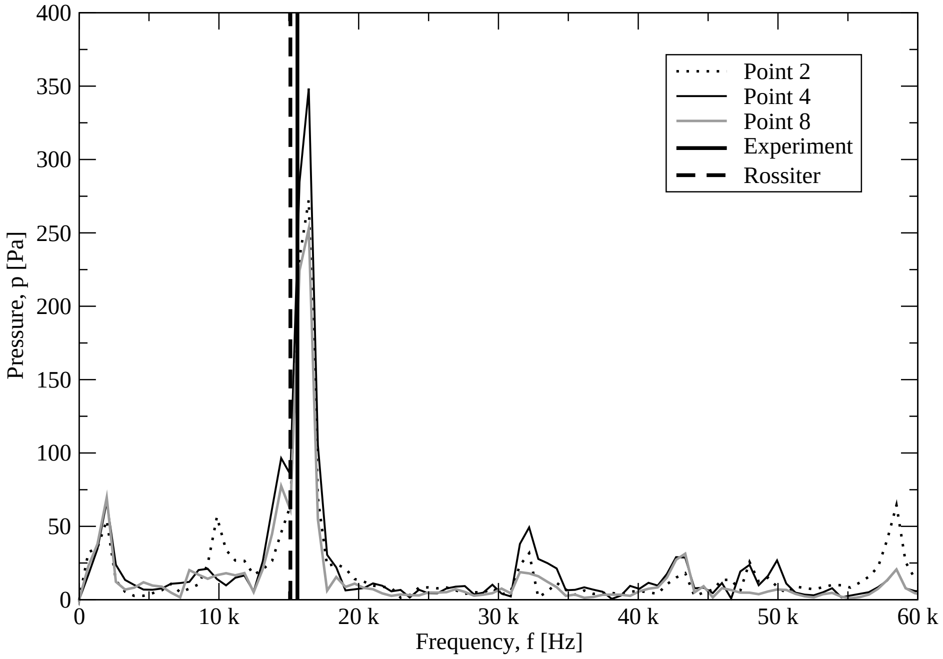


Figure 4.15: Frequency distributions for three cavities with 0.5 mm gap.

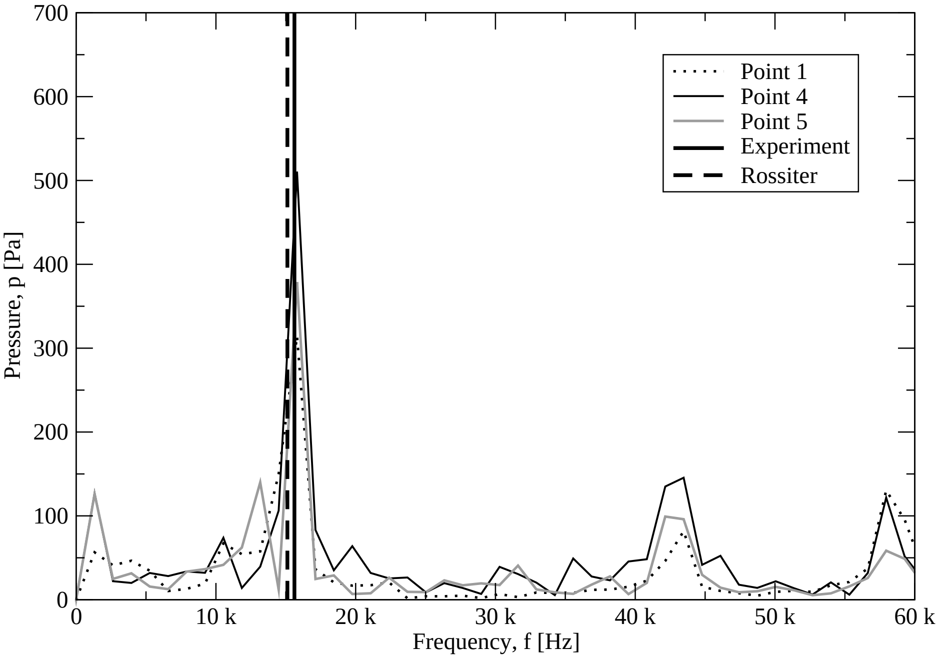


Figure 4.16: Frequency distribution for five cavities with 0.5 mm gap.

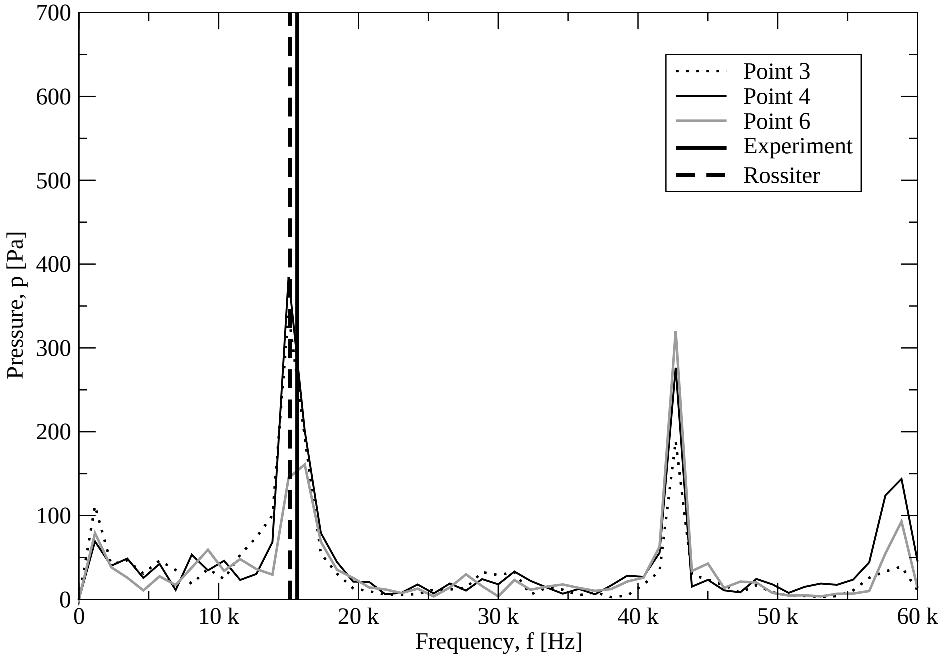


Figure 4.17: Frequency distribution for seven cavities with 0.5 mm gap.

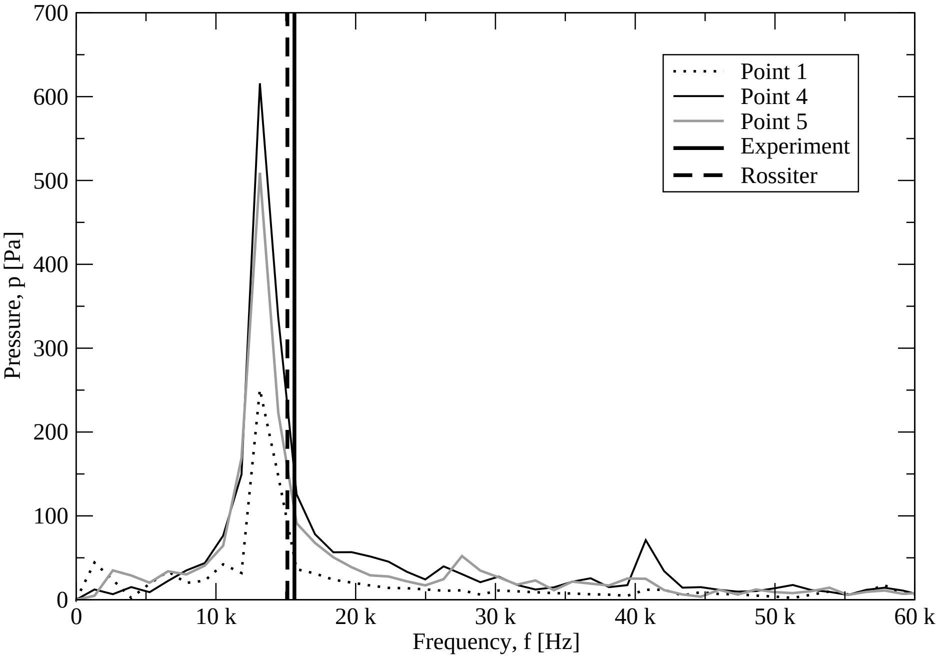


Figure 4.18: Frequency distribution for five cavities with 1.0 mm gap.

seal used in the experiment had several hundred cavities.

The agreement between the numerical results and the experimental data is remarkable considering that the simulation was basically two-dimensional and that only a single cavity was modeled. This indicates that cross flow and cavity interactions are secondary effects for the dominant frequencies. When multiple cavities were simulated, the predicted dominant frequencies did not change; however, higher harmonic frequencies were excited. The numerical simulation showed that the variation of the freestream Mach number significantly affected the dominant frequency.

4.2 CFD-Perturbation Solver: Validation and Rotordynamic Results

Three test cases were used to demonstrate the capabilities of the time-linearized RANS solver. The first test case consisted of airflow through a straight, square channel with a pressure oscillation enforced at the exit. Acoustic waves travel from the exit, upstream through the channel, and leave the domain at the entrance. The second case is Stokes' second problem; a well-known case consisting of fluid excited only by a wall oscillating parallel to itself. The third case is a straight annular gas seal with parameters roughly based on the High Pressure Oxidizer Turbopump (HPOTP) of the space shuttle main engine (SSME). Rotordynamic coefficients for this case were compared against bulk flow results from Nelson [101]. These cases were chosen to reflect the internal flows and moving wall effects encountered in gas seals.

4.2.1 Straight Channel with Oscillating Back Pressure

The domain for the first test case, a straight channel with an oscillating back-pressure, consisted of a square channel one meter in length and two centimeters in height and depth. Flow conditions at the inlet and exit were chosen such that the flow had a Mach number of 0.6 with an inlet total pressure of 129 kPa and an average backpressure of 101 kPa. A mesh of 6,400 evenly spaced nodes was used. The simulation was inviscid and second order accurate in space and time.

Results from the time linearized simulation were compared against results from a time accurate simulation. The time accurate simulation was run with an oscillating backpressure with amplitude of 5 kPa and frequency of 7,000 Hz. The simulation was marched forward in time until a steady state periodic solution was found. For the time linearized simulation, first a steady solution was found using a RANS solver. Next, the effect of the oscillating backpressure was accounted for by enforcing the perturbed pressure at the exit of the channel and using the time linearized CFD code to find the resulting perturbed flow field. A non-reflecting boundary condition was used at the channel entrance to allow the acoustic waves to freely leave the domain.

A plot of the instantaneous perturbed pressure field through the length of the channel is shown in Fig 4.19. The perturbed pressure has been non-dimensionalized with respect to the amplitude of the enforced oscillation. In this plot fluid is flowing to the right while the wave of interest is traveling to the left. The solid and dashed lines show the results of the time accurate and time linearized simulations, respectively. The acoustic wave inherently contains some non-linear effects, which are captured by the time accurate simulation but cannot be captured by the time linearized simulation. This accounts for the difference in wavelength between the two results. Also, the time linearized simulation used a non-reflecting boundary condition at the entrance of the channel while no such boundary condition was available for the time accurate simulation, further contributing to disagreement near the channel entrance. Even with these errors, results from the time linearized method match well with the time accurate.

The time accurate simulation required a large number of time steps to reach a periodic steady state. To ensure accuracy of the results, convergence within a set

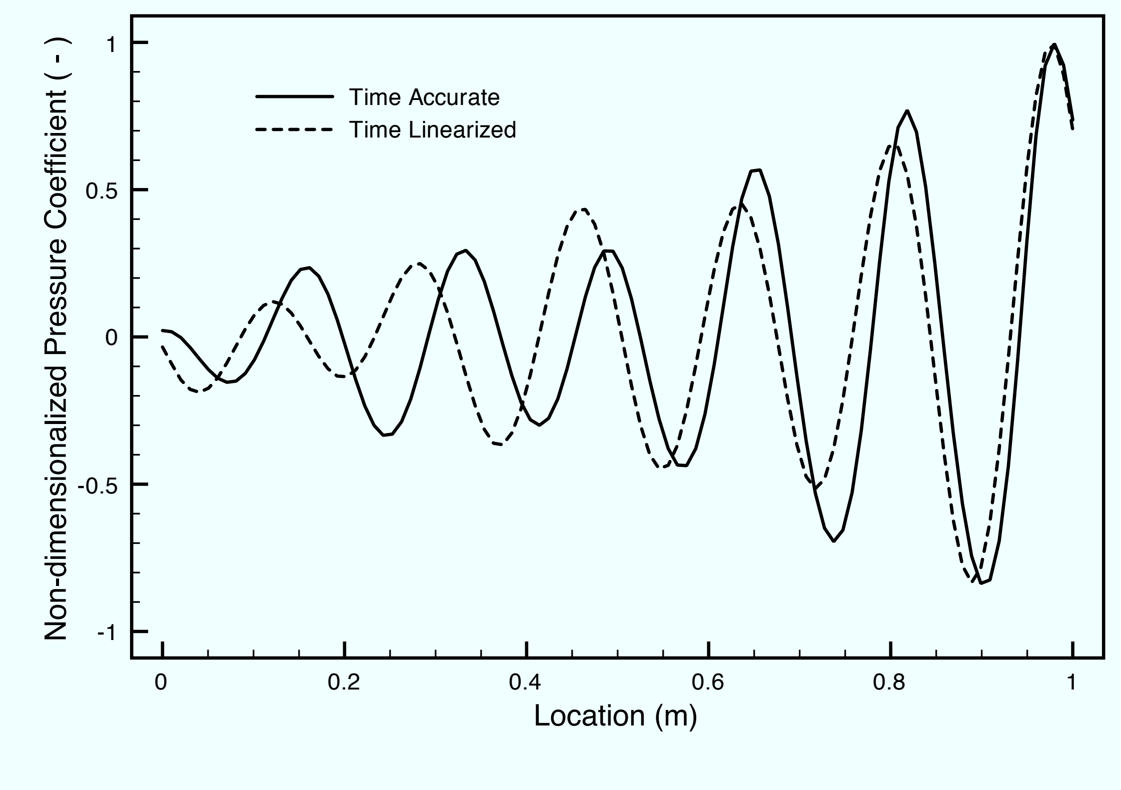


Figure 4.19: Comparison of unsteady pressure between time-linearized and exact.

tolerance must be reached for each time step. This process is very computationally expensive. The time linearized simulation required only a single non-linear convergence for the steady zeroth order solution, and a single linear convergence for the unsteady perturbed solution. A speed up of approximately an order of magnitude is seen when using the time linearized method versus the time accurate.

4.2.2 Oscillating Flat Wall - Stokes' Second Problem

Stokes' second problem assumes an infinite wall with a flow field that extends infinitely away from the wall. To approximate this for the CFD simulations, a domain was chosen 0.1 meters long with periodic boundary conditions with a solid wall boundary along the bottom. The height of the domain, 5 cm, was chosen such that the influence of the moving wall dissipated before reaching the top boundary. A depth of 1 mm was chosen to enforce the 2D nature of Stokes' second problem while using a 3D code. A mesh of 4,500 nodes was used. The nodes were clustered near the wall to capture viscous effects. Standard atmospheric conditions were enforced at the boundaries. The simulation was viscous and laminar and second order accurate in space.

The results from the time-linearized simulation were compared against the exact solution

$$\frac{u}{u_0} = \exp\left(-\frac{y}{y^*}\right)\cos\left(\theta - \frac{y}{y^*}\right) \tag{4.1}$$

where u is the fluid velocity,u_0 is the maximum wall velocity, y is the distance from the wall, $y^* = \sqrt{\frac{2\nu}{\omega}}$, ν is the viscosity, ω is the frequency, and $\theta = \omega t$ is the phase angle.

For the time-linearized simulation, a steady zeroth order solution was used with zero velocity and constant pressure and density at standard atmospheric conditions. The effects of the oscillating wall were accounted for with the unsteady perturbed

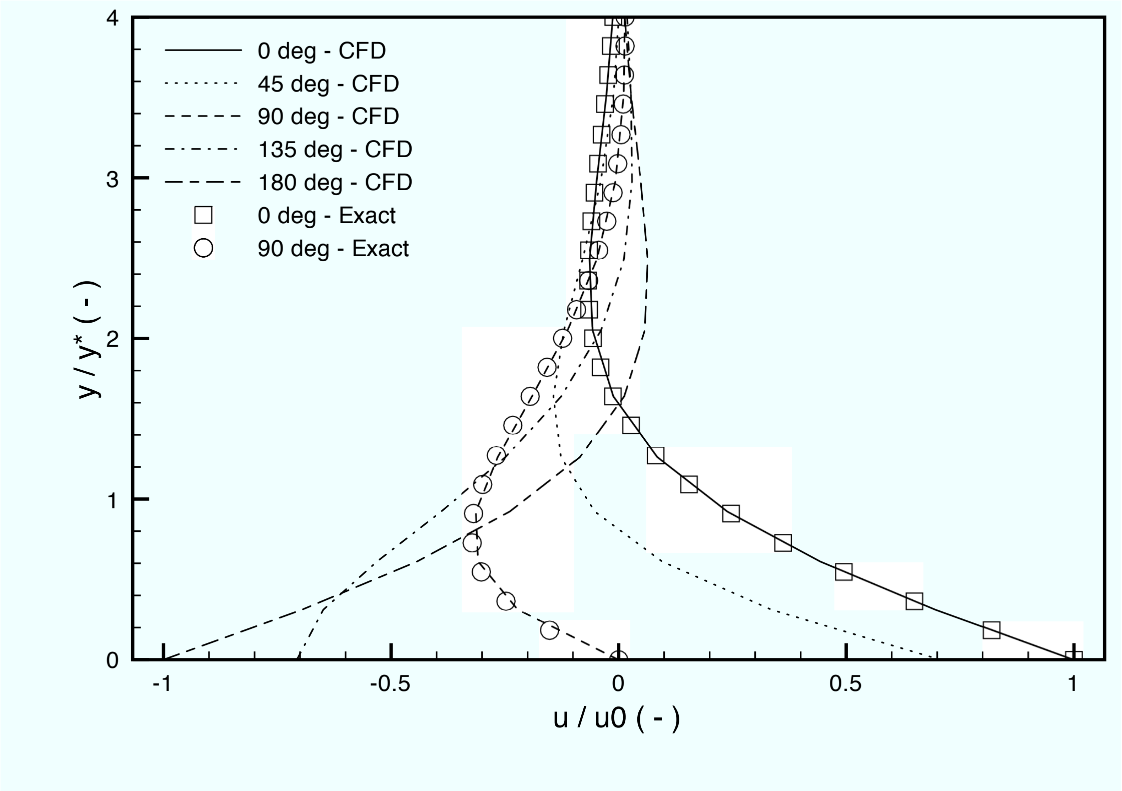


Figure 4.20: Comparison of flow profiles time-linearized and exact.

grid. The unsteady grid matched the oscillations at the wall and smoothly died out moving away from the wall.

A plot of the velocity profile from the time linearized simulation at different phase angles is shown in Fig. 4.20. The exact solution for phase angels 0 and 90 degrees are also shown. The agreement is very good for all phase angles demonstrating the ability of the time-linearized CFD code to predict perturbed viscous effects as well as the effects of the moving mesh.

4.2.3 Straight Annular Seal - HPOTP of the SSME

A straight annular seal case was run to validate the time linearized RANS code. The results from the time linearized CFD-perturbation method were compared against bulk flow results from Nelson [101]. The parameters for the straight seal used are roughly equivalent to the hot gas secondary seal of the High Pressure Oxidizer Turbopump (HPOTP) of the space shuttle Main Engine (SSME).

The domain used for the time linearized simulations was a 2 degree slice of a straight annular seal with smooth surfaces on the rotor and stator. Parameters for the spatial dimensions of the domain can be found on Table 4.5 along with operating conditions. For the steady, zeroth order solution, periodic boundary conditions were used to take advantage of the rotational symmetry. For the unsteady, first order simulation, the periodic boundary condition included a phase shift for the perturbed state vector, $\vec{Q}_1$, to ensure the proper axial phase distribution.

Total inlet pressure	Backpressure	Length	Radius	Clearance	Shaft rotational velocity	Shaft whirl velocity
1.52 MPa	0.65 MPa	6.5 mm	3.25 cm	0.086 mm	30,400 rpm	30,400 rpm

Table 4.5: Parameters for the straight annular seal case.

The steady mesh contained nodes evenly distributed along the axial and annular directions, and clustered near the solid surfaces to capture the viscous effects. The unsteady perturbed mesh was defined to match the whirl motion of the rotor at the inner radius and match the stationary stator at the outer radius. To ensure a smooth transition, the motion of the mesh varied linearly with the radius in the interior of the domain. Non-reflective boundary conditions were used at the entrance and exit

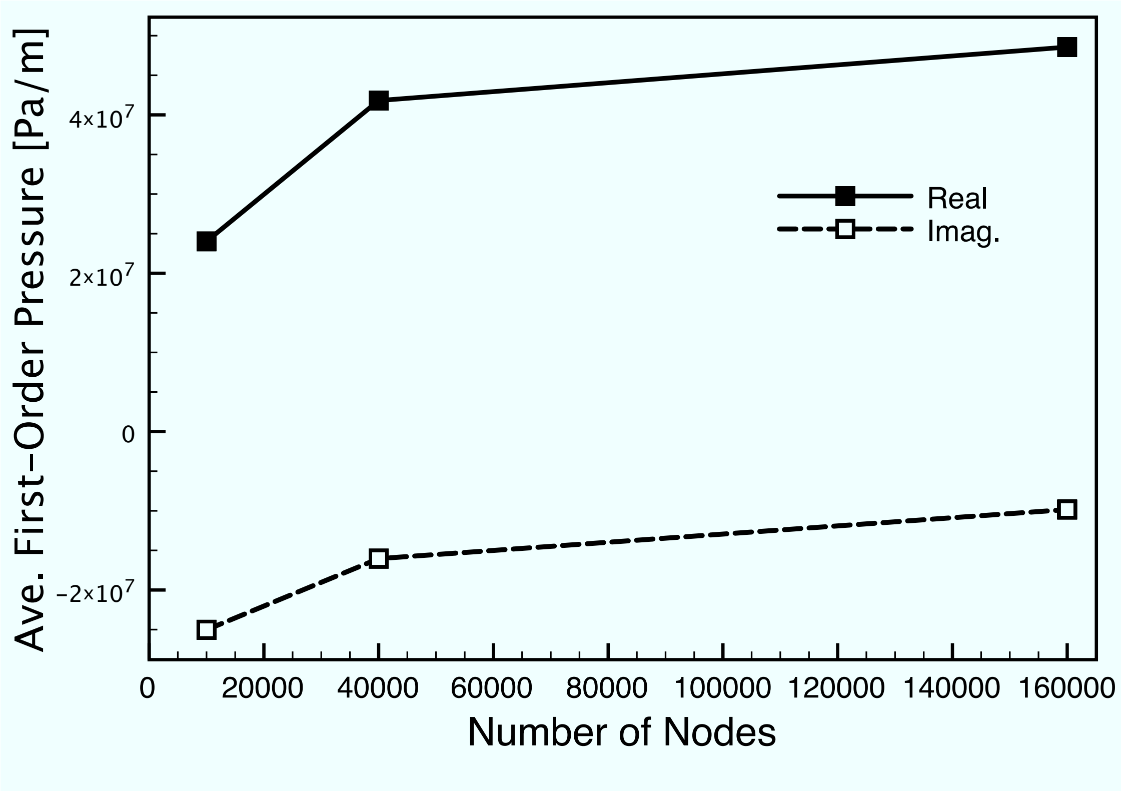


Figure 4.21: Straight seal mesh refinement study.

to minimize spurious waves.

A mesh convergence study was performed for the straight seal case to determine an adequate spatial refinement. Three mesh refinements were used with each refinement level containing twice as many nodes in the axial and radial directions as the next coarser level. Figure 4.21 shows the average first-order pressure within the seal plotted against the number of nodes for each mesh. It was determined that the medium mesh, with 40,000 nodes, offered a good balance between accuracy and efficiency. All cases run used meshes with the same or similar refinement level as the medium mesh.

Figures 4.22 and 4.23 show a comparison of the stiffness and damping coefficients,

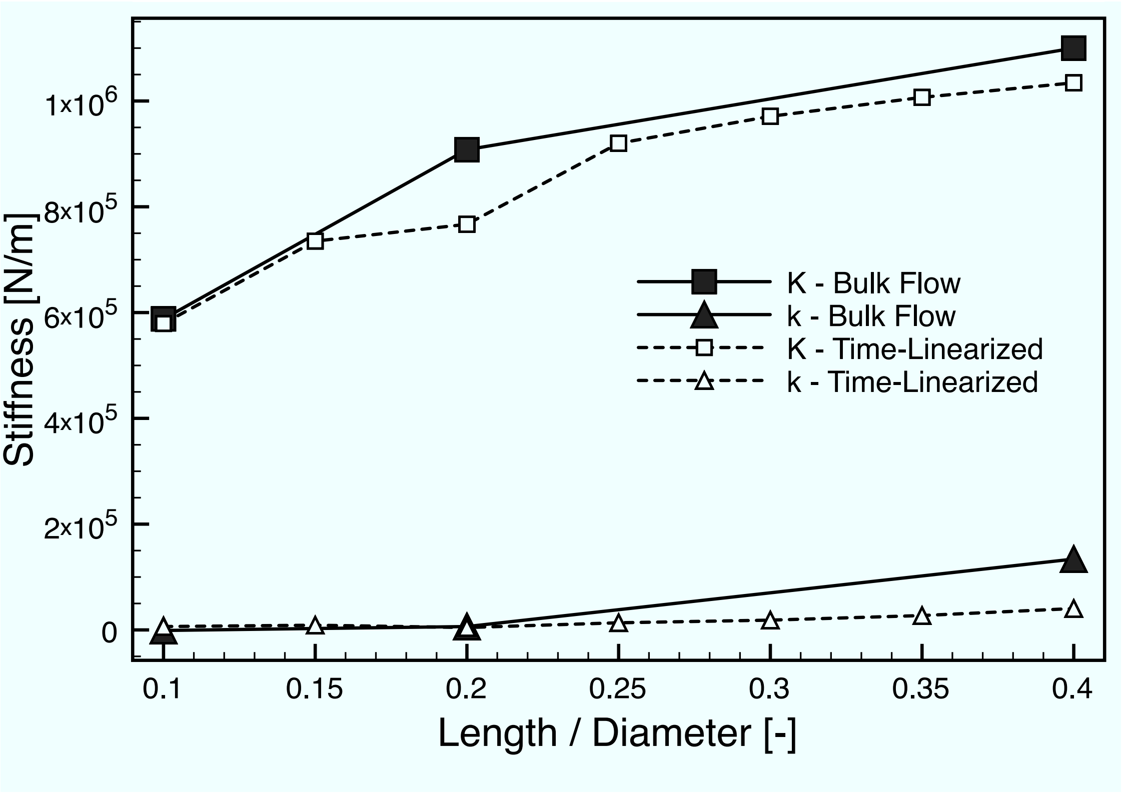


Figure 4.22: Direct and cross-coupled stiffness coefficients with varying L/D.

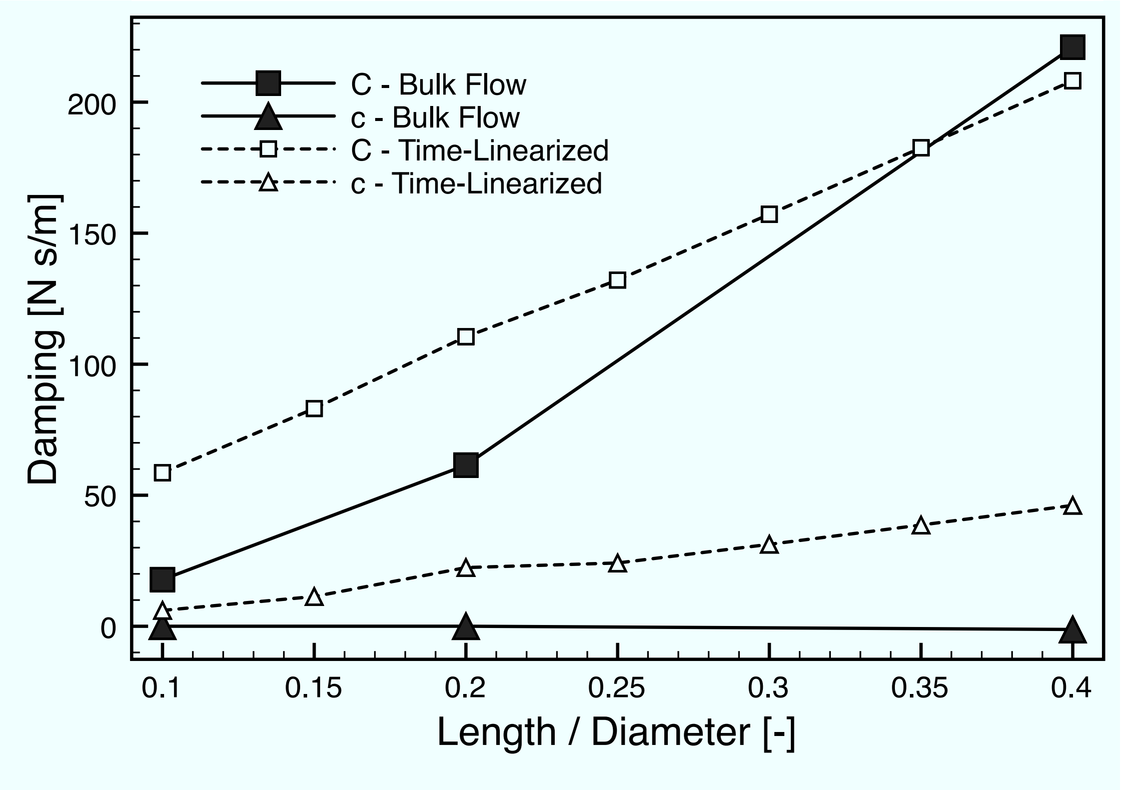


Figure 4.23: Direct and cross-coupled damping coefficients with varying L/D.

respectively, between the time linearized solver and Nelson's bulk flow method [101]. Nelson reported that the values for the direct and cross-coupled inertial coefficients were negligible for these cases and, therefore, were not reported. The current research also found this to be true.

Excellent agreement is seen between the bulk flow and time-linearized models for both the direct and cross-coupled stiffness. Time-linearized cases were run while varying the length to diameter by a value of 0.05 starting from 0.1 and ending at 0.4. Nelson reported stiffness values for length to diameter ratios of 0.1, 0.2 and 0.4. A positive trend is predicted, using either method, for both the direct and cross-coupled stiffness with a greater effect seen in the direct.

Predictions for the direct and cross-coupled damping coefficients show good agreement between the time-linearized and bulk flow models. As with the stiffness, cases were run with length to diameter ratios ranging from 0.1 to 0.4 and compared against reported bulk flow values from Nelson. Both the bulk flow and time-linearized models showed a positive trend for the direct damping with a similar range of values. A slightly greater effect was predicted by the bulk flow model. The length to diameter ratio had very little effect on the cross-coupled damping as predicted by the bulk flow model while the time-linearized model predicted a positive correlation.

To demonstrate some the ability provided by the time-linearized method, Fig. 4.24 reports the variation of the real and imaginary parts of the first-order pressure with respect to varying length to diameter ratios. The rotordynamic coefficients are calculated using first-order forces on the rotor and the forces are calculated by integrating the first-order pressure. Plots of the pressure, therefore, provide a more localized breakdown of the effects of varying length to diameter ratio than global values, such as force or rotordynamic coefficients. Pressure values are plotted against location along the axis with the location normalized against to total seal length to

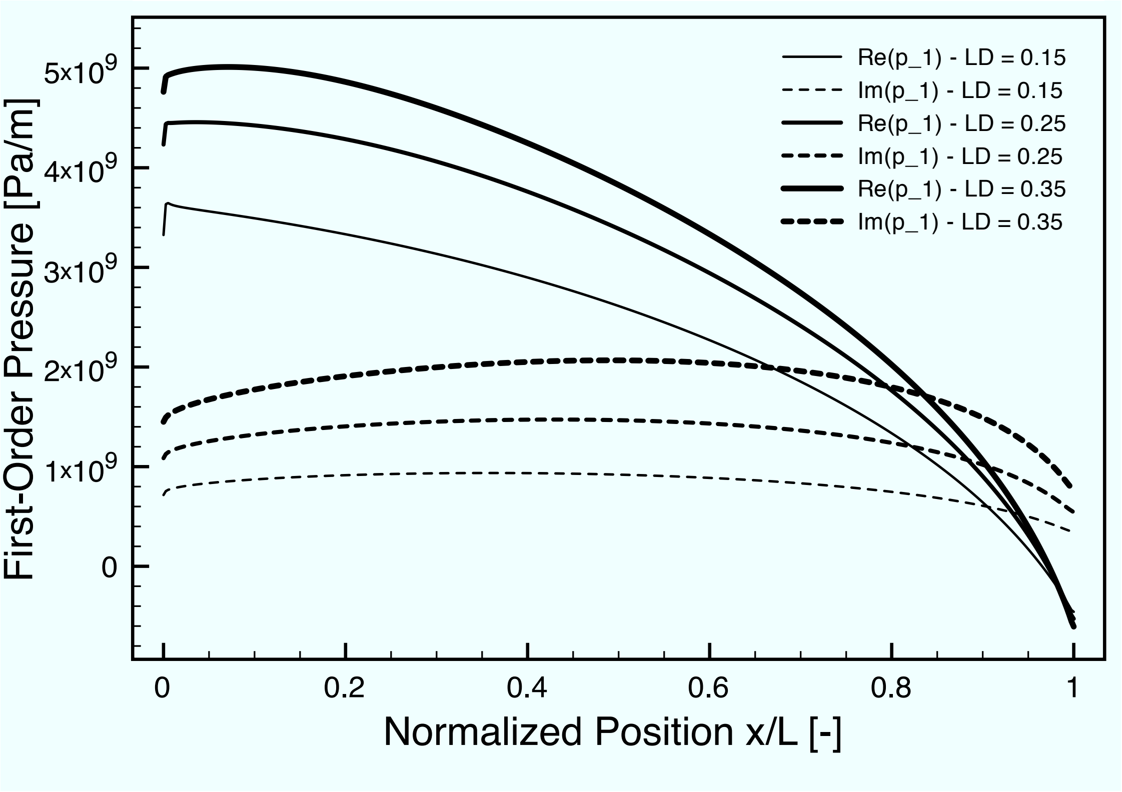


Figure 4.24: First-order pressure for varying L/D.

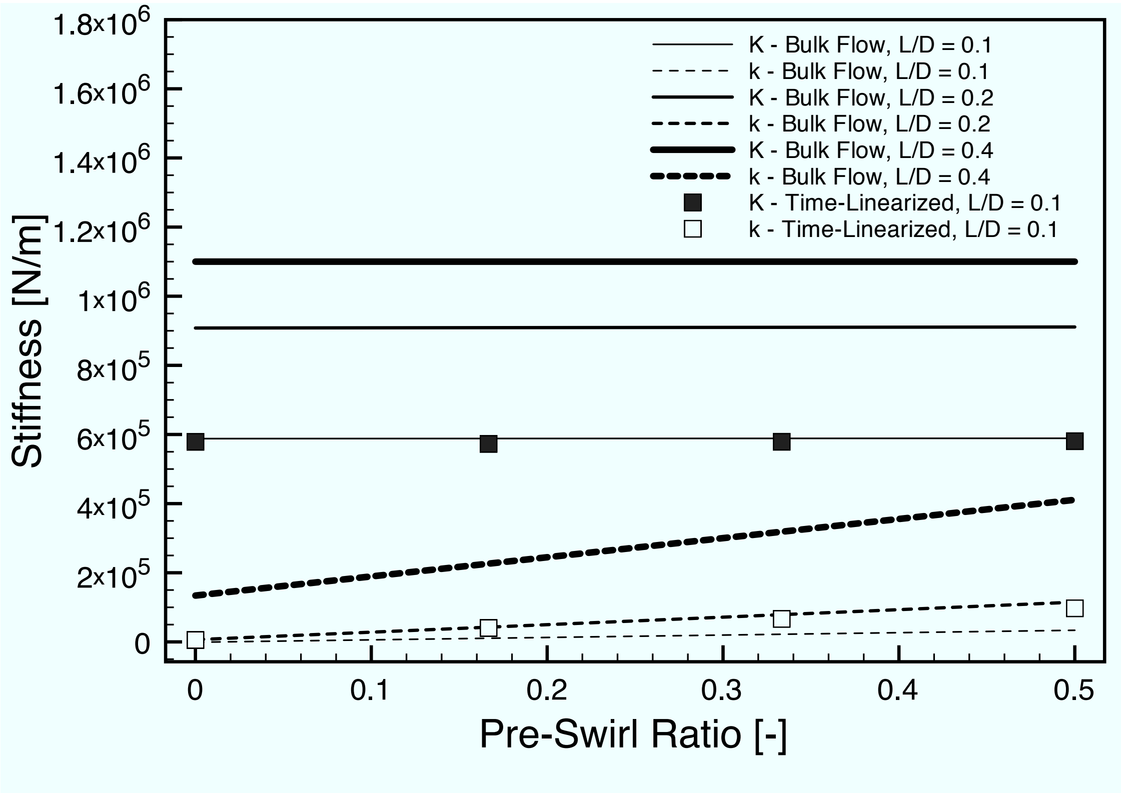


Figure 4.25: Direct and cross-coupled stiffness coefficients with varying pre-swirl.

allow comparison between seals of different lengths. The solid lines show the real part while the dashed lines show the imaginary.

For all three reported length to diameter ratios, the real part of the first-order pressure drops from a maximum value near the seal inlet to just below zero near the outlet. Recall that the first-order pressure is just a perturbed term, so negative values are physical. The imaginary part remains more constant throughout the length of the seal than the real part, but still shows a slight drop near the the outlet. The maximum values for both the real and imaginary parts of the first-order pressures increase with increasing length to diameter ratios.

A comparison of predicted rotordynamic coefficients between the time-linearized

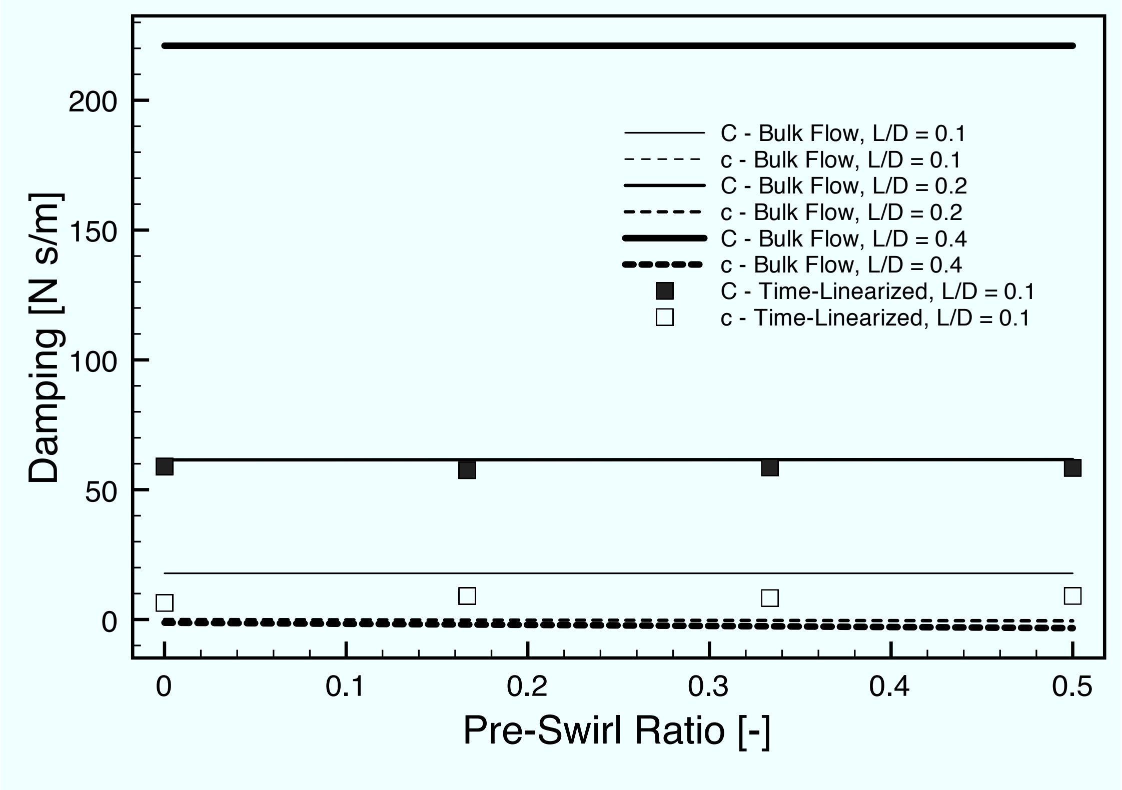


Figure 4.26: Direct and cross-coupled damping coefficients with varying pre-swirl.

and bulk flow methods for a range of pre-swirl ratios are shown in Figs. 4.25 and 4.26. The pre-swirl ratio is the ratio of the rotational flow velocity at the inlet of the swirl with respect to the rotational velocity of the rotor. Values were reported by Nelson with pre-swirl ratios of 0.0 and 0.5 with length to diameter values of 0.1, 0.2 and 0.4. Time-linearized simulations were run with a length to diameter ratio of 0.1 at pre-swirl ratios of 0.0, 0.167, 0.333 and 0.5.

The time-linearized and bulk flow models were in excellent agreement for the direct stiffness at a length to diameter ratio of 0.1. Both methods showed almost no correlation with pre-swirl. A positive correlation was shown by both the time-linearized and bulk flow models for the cross-coupled stiffness, with a greater effect predicted by the time-linearized.

Both the time-linearized and bulk flow models predicted almost no correlation with respect to pre-swirl ratio for the direct damping coefficient and a slight positive correlation for the cross-coupled. Predicted values for the damping coefficients were higher using the time-linearized method, but fell near or within the ranges predicted by the bulk flow model.

A localized breakdown of the effect of varying pre-swirl ratio on the real and imaginary parts of the first-order pressure is shown in Fig. 4.27. The real part is shown with solid lines and the imaginary part is shown with dashed. The pre-swirl had almost no effect on the real part of the first-order pressure. There is a positive correlation between pre-swirl ratio and the imaginary part of the first-order pressure.

4.2.4 Stepped Labyrinth Seal

A stepped labyrinth seal case was run to validate the time linearized RANS solver. The seal geometry and flow conditions were chosen to match tests run by Wagner et al. [102] at Siemens Energy Sector, Oil and Gas Division. In that work,

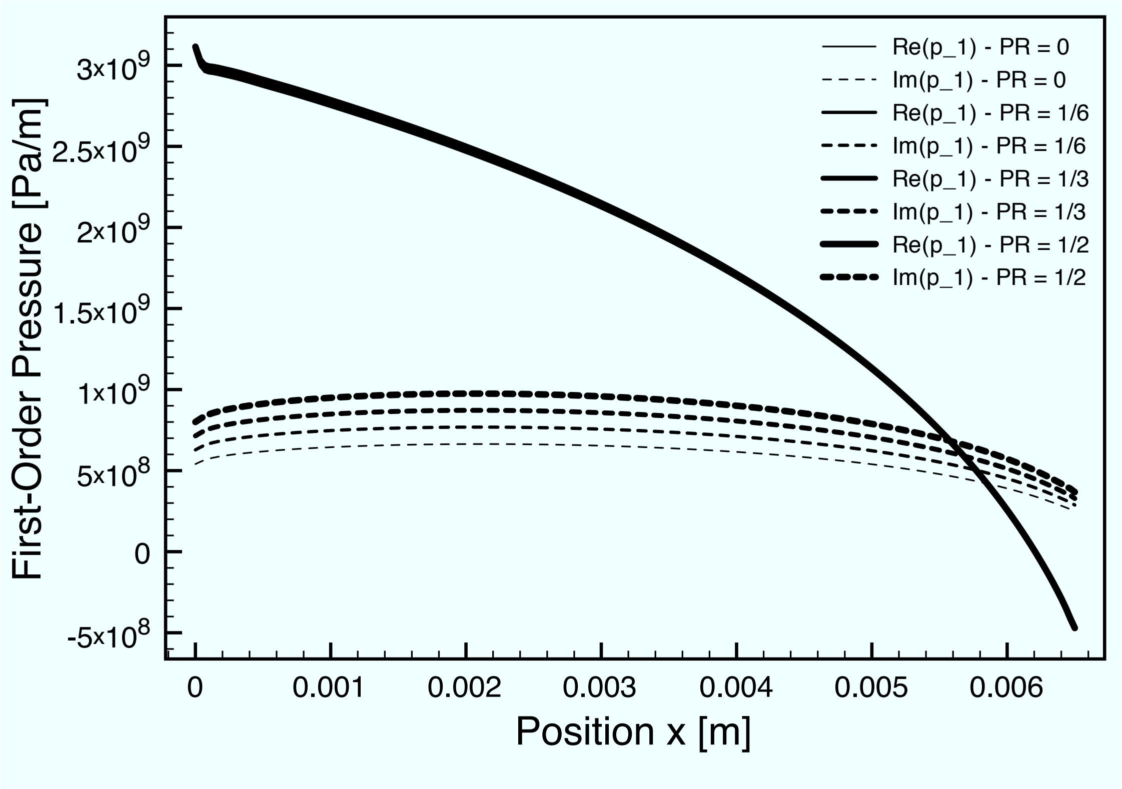


Figure 4.27: First-order pressure for varying pre-swirl.

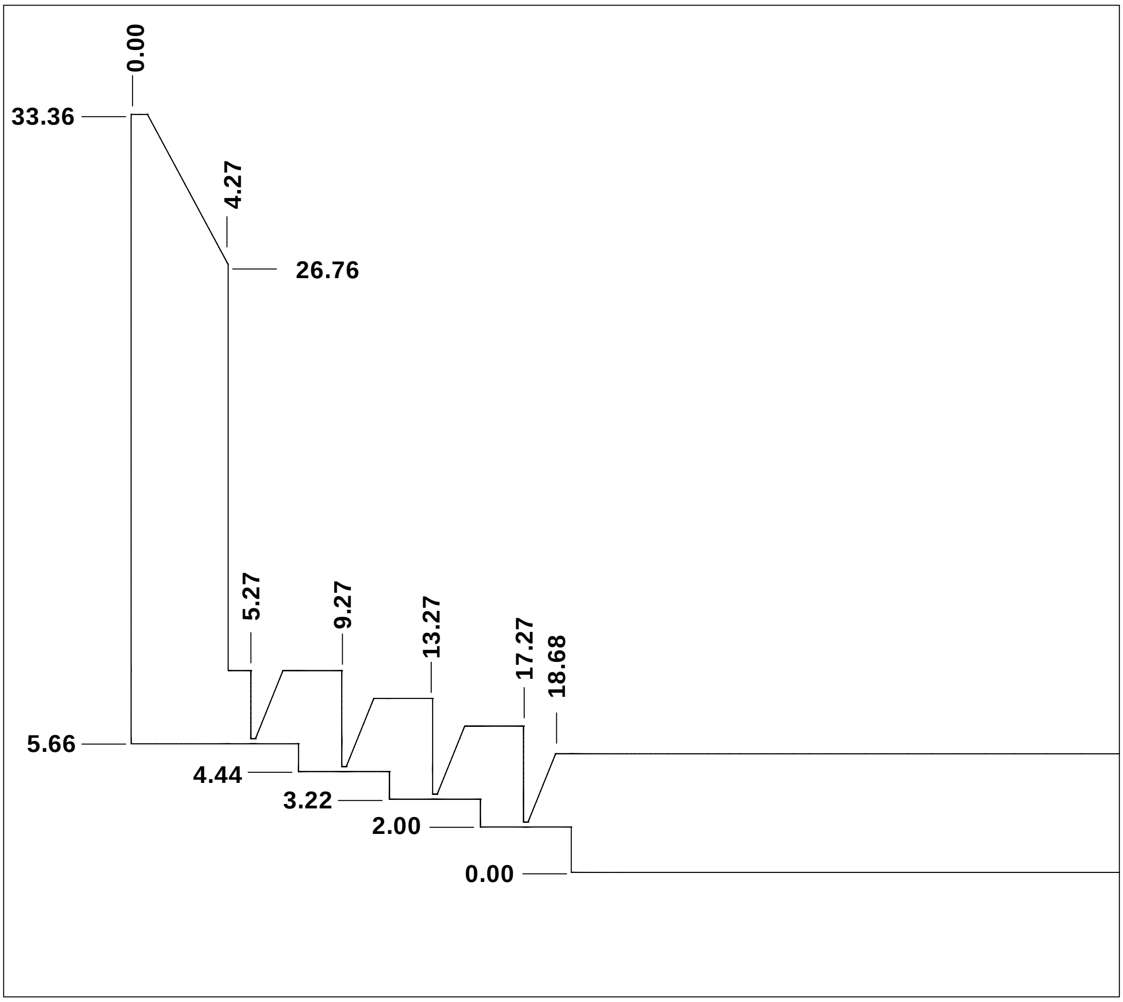


Figure 4.28: Details of the geometry of the stepped labyrinth seal (mm).

a full-scale high-pressure test rig was used to measure the rotordynamic coefficients for an impeller-eye-seal. A time-accurate CFD model was also developed to provide further insight on the seal dynamics. Experimental measurements and time-accurate CFD results from Wagner et al. are compared against predictions made by the time-linearized RANS solver developed here.

The domain used for the time-linearized simulations was a 0.25 degree slice of a stepped labyrinth annulus seal. A cross-section view of the domain detailing the

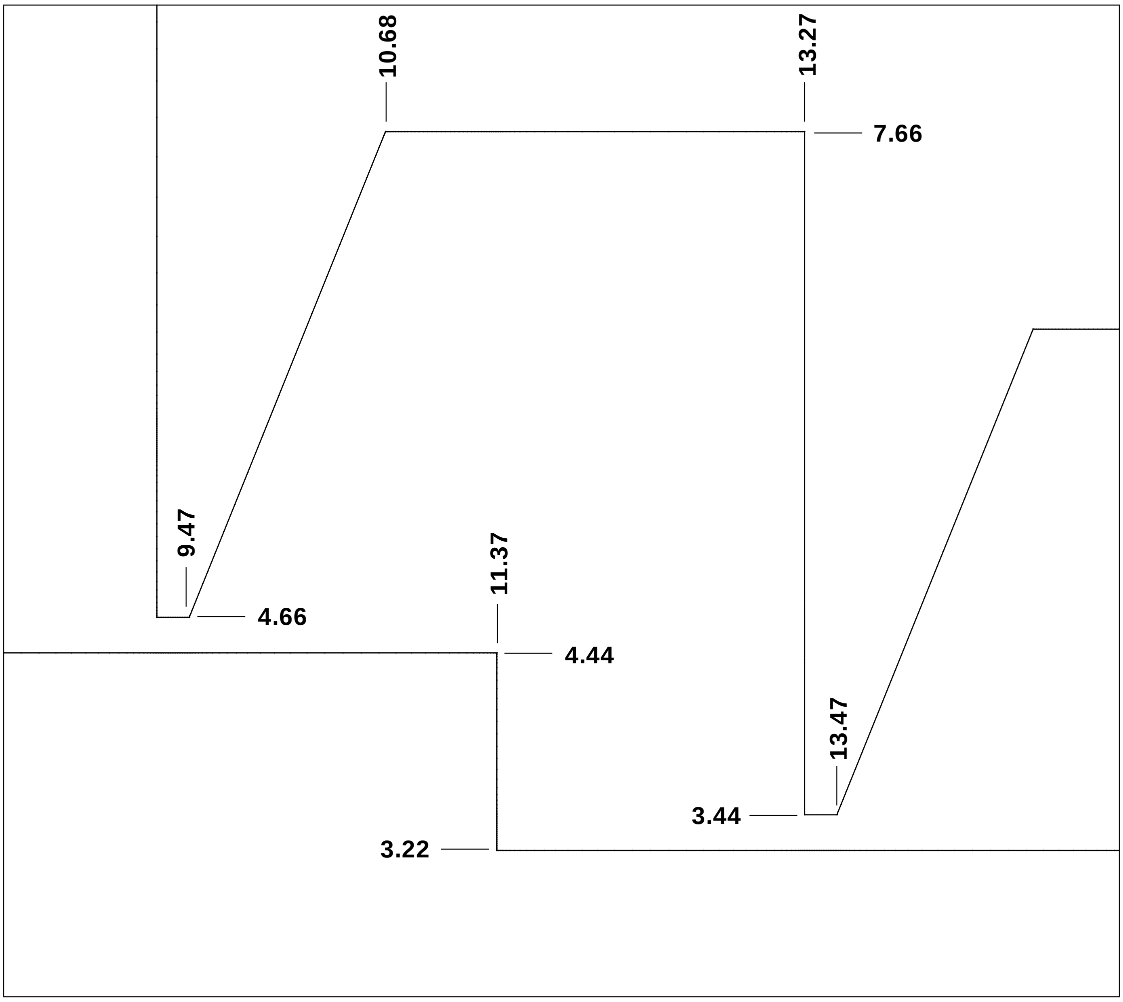


Figure 4.29: Details of the geometry of the cavities of the labyrinth seal (mm).

geometry are shown in Figs. 4.28 and 4.29. Four teeth, extending from the stator, separate the seal into an inlet region, three stepped cavities, and a downstream annulus extending to the right. Figure 4.28 shows the relative positions of each area of the geometry while Fig. 4.29 details the shape of the cavities. Measurements in the radial direction are taken from the radius of the rotor in the downstream annulus region while measurements in the axial direction are taken from the leading edge of the seal. Table 4.6 provides further spatial parameters along with operating conditions.

Table 4.6: Geometric and operating parameters for labyrinth seal test case.

Number of tips	4		Rotational speed	15,000 rpm
Tip pitch	4.0 mm		Inlet swirl	98.0 m/s
Tip height	4.0 mm		Static inlet pressure	2000 kPa
Radial gap width	0.10 mm		Inlet temperature	49.7° F
Axial tip width	0.20 mm		Static outlet pressure	1600 kPa
Radial step height	1.0 mm		Gas: nitrogen	28.0 kg/kmol
Diameter under 1st tip	130.44 mm		Whirl amplitude	10.0 μm

As with the straight annular seal case, periodic boundary conditions were used for both the steady, zeroth-order and unsteady, first-order solutions to take advantage of the rotational symmetry. A phase shift of the first-order state vector, $\vec{Q}_1$, was enforced to ensure proper axial phase distribution.

The mesh contained 180,000 nodes, clustered near solid surfaces to capture viscous effects. The distance from the surface to the first interior node was carefully chosen to ensure the $y+$ number never exceeded 3. The nodes were distributed with growth rates of at most 1.3. A grid convergence study was carried out and detailed in Wagner, et al.[102] which resulted in each cavity being gridded with about 2,000 nodes for each tangential slice. To remain consistent, the grid used here contained about the same.

Figures 4.30-4.32, show comparisons of the magnitude of the first-order pressure in each of the three cavities between experimental measurements and time-accurate CFD predictions from Wagner, *et al.* [102] as well as time-linearized CFD-perturbation predictions generated by the time-linearized CFD-perturbation method. Values are shown for a range of excitation frequencies. Stepped labyrinth seals have very complex flow patterns made up of several distinct pieces working in concert. Comparing pressures in each cavity, as opposed to an overall force or rotordynamic coefficient, provides a more detailed breakdown of the flow. Predictions from CFD can be compared against measured values from each area of the seal and can be analyzed separately. Extracting these details from the flow is trivial using CFD, but can be very difficult using an experimental test rig. Pressure measurements from each cavity can reasonably be made with the test rig and provide good comparisons for the CFD, detailing the separate areas of the flow.

Values predicted by the time-linearized CFD-perturbation method for all three cavities agreed very well with measured values at 100 and 200 Hz. The predicted values, however, remained about constant or dropped for 300 and 400 Hz while the measured values showed a positive quadratic correlation. Predictions from the time-accurate CFD method were consistently lower than measured but captured the quadratic trend better than time-linearized.

The experimental measurements clearly indicate a quadratic relationship with respect to frequency, most likely due to added inertia from gas trapped in the seal. The overall magnitude of the perturbed pressures agreed well with measurements but the quadratic relationship was missed. This suggests the linear approximation made by the CFD-perturbation method is missing an important contributor to the

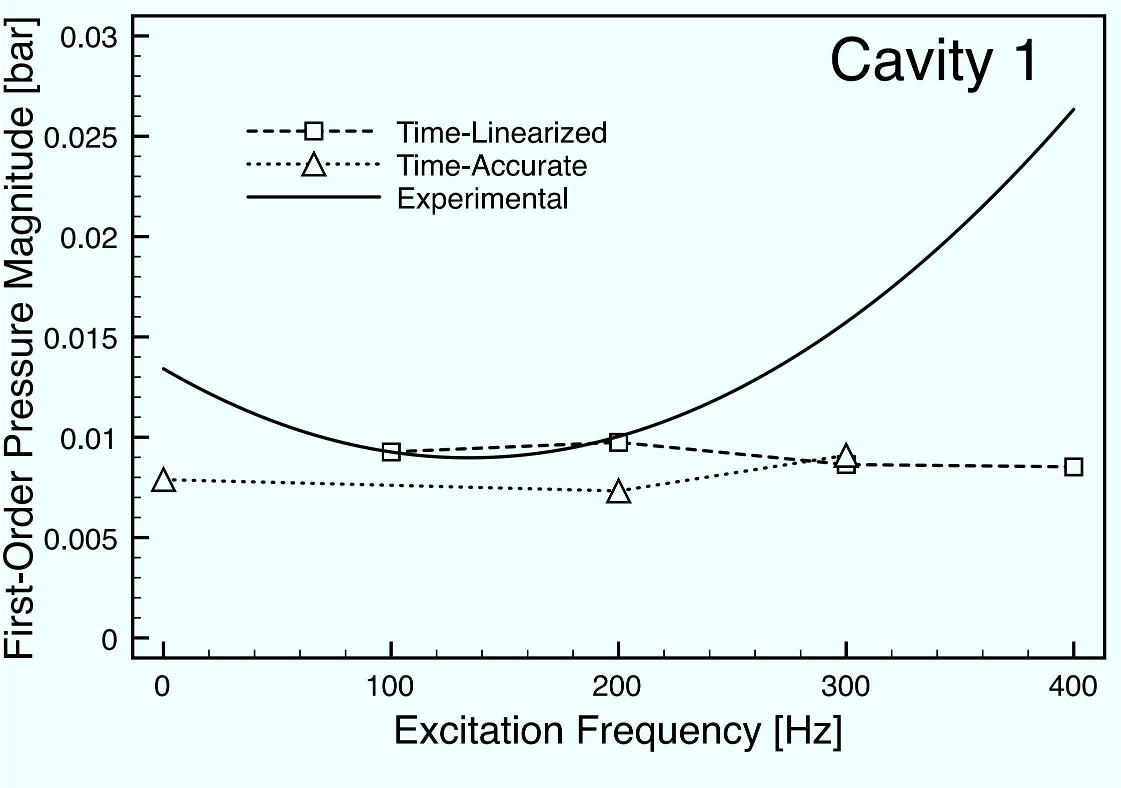


Figure 4.30: Magnitude of the first-order pressure within the first cavity.

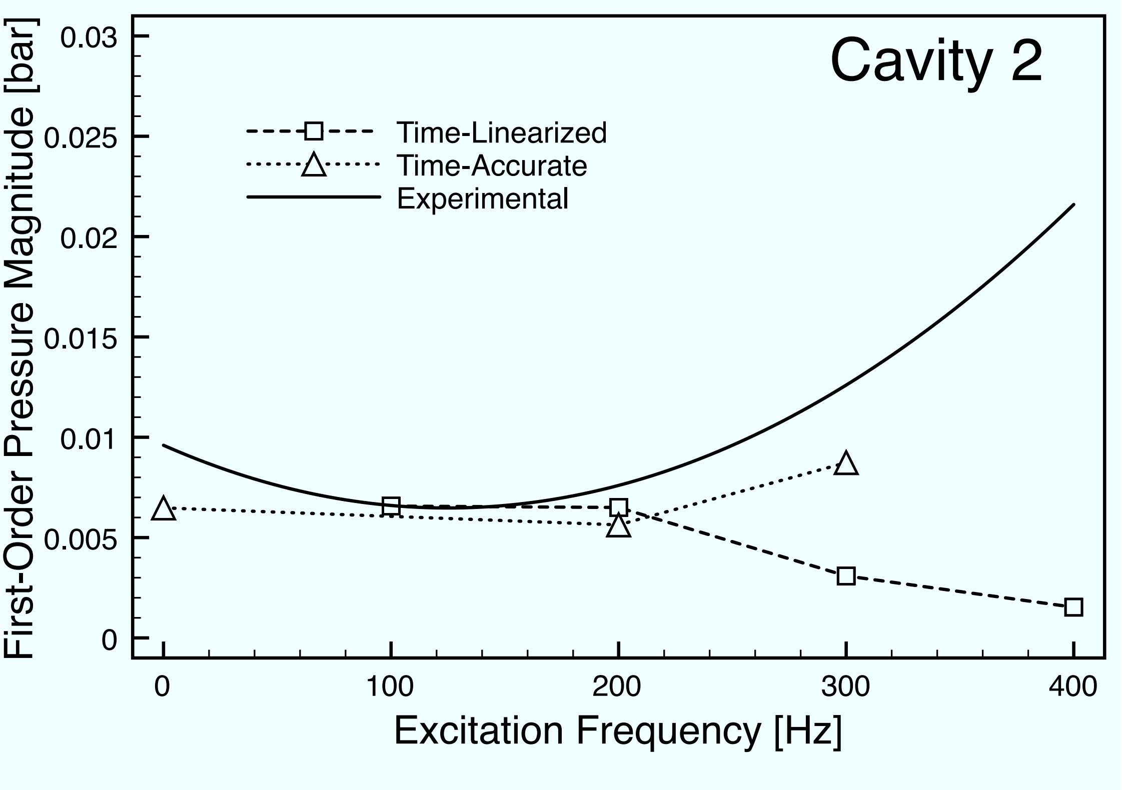


Figure 4.31: Magnitude of the first-order pressure within the second cavity.

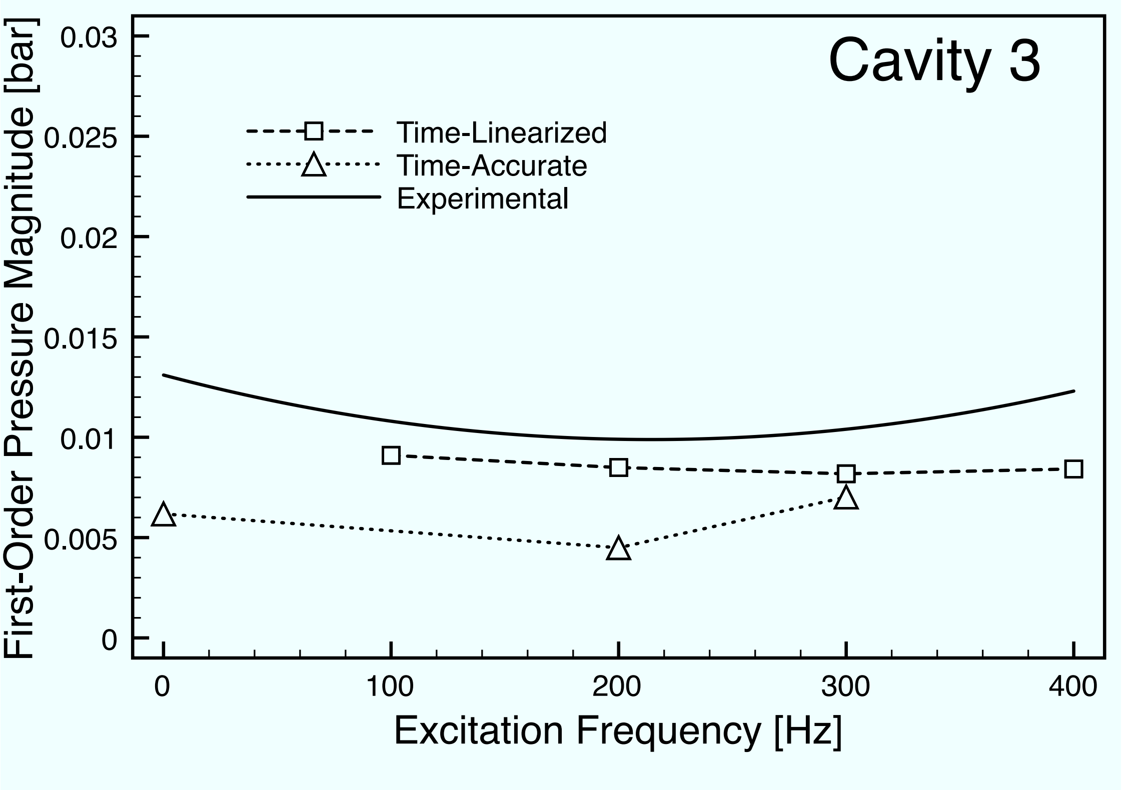


Figure 4.32: Magnitude of the first-order pressure within the third cavity.

rotordynamic stability for labyrinth seals. To capture the added inertial effects in labyrinth seals, quadratic terms or even higher should be included in the CFD-perturbation model.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

A Reynolds-averaged Navier-Stokes (RANS) solver developed in-house was used to simulate grazing channel flow past single and multiple cavities. The objective of this investigation was to predict fluid instabilities in hole-pattern stator seals. The numerical results generated with the RANS solver showed good agreement with those obtained using a commercial Large Eddy Simulation code. In addition, the numerical results agreed well with experimental data. Rossiter's formula, a popular semi-empirical model used to predict frequencies of hole-tone acoustic instabilities caused by grazing fluid flow past open cavities, was modified using the RANS solver results to allow for its application to channel flows. This was done by modifying the empirical constant κ, the ratio of vortex velocity and the freestream velocity, by tracking vortex position through a cycle of the flow. The dominant frequencies predicted using the Rossiter's formula with the new κ value matched well the experimental data for hole-pattern stator seals. The RANS solver accurately captured the salient features of the flow/acoustic interaction and predicted well the dominant acoustic frequencies measured in an experimental investigation. The flow solver also provided detailed physical insight into the cavity flow instability mechanism.

An unstructured, fully 3D, Reynolds' averaged Navier-Stokes (RANS), time-linearized based CFD-perturbation solver was developed to predict rotordynamic coefficients for annular gas seals. It demonstrates an improvement over past CFD-perturbation methods used for turbomachinery by being unstructured, fully 3D and including turbulent effects. Earlier CFD-perturbation solvers were structured and 2D or semi-3D. This limited the geometries of the seals that were accessible to them.

With an unstructured and fully 3D solver, virtually any seal geometry may be simulated. The time-linearized solver, being based on an established full-order in-house RANS solver, inherits all of the features of the full-order code. The methods used in development of the time-linearization are robust enough to ensure their applicability to any new features that may be added to the full-order code. This ensures the CFD-perturbation method will benefit from any future development of the full-order in-house RANS solver.

Results were presented for the following test cases: channel with oscillating backpressure, Stokes' second problem—oscillating flat wall, straight seal—HPOTP, and a stepped labyrinth seal. The results for the time-linearized RANS solver agreed well with the time accurate RANS solver for the channel with oscillating backpressure case with slight differences in the acoustic wavelength due to unaccounted non-linear effects. The time-linearized RANS solver required approximately an order of magnitude less computational effort. Results for Stokes' second problem matched the exact solution extremely well, demonstrating the ability of the method to predict perturbed viscous effects and effects due to the moving mesh. Rotordynamic coefficients were presented for a straight annular gas seal at conditions roughly matching the HPOTP of the space shuttle main engine. The results from the time linearized RANS solver matched well with bulk flow results presented by Nelson [101] for a varying length to diameter ratio as well as a varying pre-swirl ratio. Overall values for the first-order pressure in the stepped labyrinth seal matched well with empirical measurements presented by Wagner, et al. However, the correlation of the first-order pressures with the excitation frequency did not match indicating the inability of the time-linearized assumption to capture quadratic effects.

5.2 Future Work

Research will continue in the analysis of cavity flows and their application to seals. Currently, only 2D approximations of the cavities have been used. Future studies will include full 3D simulations with multiple cavity shapes including grooves, cylindrical holes and hexagonal holes. These 3D cavities contain more complex flow features and may require a more intricate analytical model than is offered by Rossiter's equation. Based on the results presented in this dissertation, the 2D approximation does a good job capturing the dominant flow features of the 3D cavities. This suggests that Rossiter's equation should be at the core of any future analytical models developed for more complex seal geometries.

Additionally, only a small number of cavities have been used to explore the interaction of multiple cavities, which in reality can number in the hundreds or thousands. Separate cavity flows may interact with one another by way of the sound waves, emanating from each cavity, exciting neighboring cavity flows and potentially forcing the neighbor into a different frequency. The neighbor, in turn, may also excite the original cavity flow. With hundreds of cavity flows interacting, each seeing a slightly different local Mach number, a rich and highly complex flow emerges. Simulating this flow with current computational tools is a daunting task, particularly when complex cavity geometries are included.

The primary difficulty faced in simulating these complex seal geometries is the computational cost required to capture the complex flow features. Parallelization schemes currently being developed in-house and implemented in UNS3D should provide the computational power to run these simulations.

The primary limitation of the time-linearized based CFD-perturbation solver presented in this dissertation is the linear approximation as demonstrated by the labyrinth seal case. The small perturbation method used to linearize the code may be continued beyond linear and additional terms may be kept. This would allow the simulations to capture quadratic, cubic or even higher order effects depending on how many terms are kept. Because of the nature of small perturbation approximations, each additional term would generate an additional linear system, similar in form to the current first-order system. The zeroth-order system would remain the same and be the only truly non-linear system to be solved. For instance, if three terms are kept there would be the nonlinear zeroth order system and three linear systems associated with the first-order, second-order and third-order terms. The additional linear systems would be solved using the scheme developed in this dissertation. The would be solved one at a time, in order from low to high. This would also allow a user to terminate or continue the simulation until a desired accuracy is reached.

To make the CFD-perturbation method fully nonlinear, a harmonic balance approach should be used. Harmonic balance schemes attempt to solve the non-linear system using an initial guess, linearized corrections and Newton iterations. They are commonly used to analyze nonlinear electrical circuits and have also been used recently to analyze instabilities in turbomachinery [103, 104]. The most difficult component of a harmonic balance scheme is linearizing the full system, which has already been done for the time-linearization method presented in this dissertation. Future work will be done to fit the time-linearized and full-order solvers into the harmonic balance, Newton iteration framework to produce a fully nonlinear CFD-perturbation solver.

REFERENCES

[1] F. J. Dietzen and R. Nordmann, “Calculating rotordynamic coefficients of the seals by finite-difference techniques,” *ASME J. of Tribology*, vol. 109, pp. 338–394, 1987.

[2] N. Kim and D. Rhode, “A new CFD-perturbation model for the rotordynamics of incompressible flow seals,” in *ASME Turbo Expo*, 2000-GT-402, (Munich, Germany), May 2000.

[3] J. Xi and D. Rhode, “Rotordynamics of turbine labyrinth seals with rotor axial shifting,” *International J. of Rotating Machinery*, pp. 1–11, 2006.

[4] K. C. Hall and C. B. Lorence, “Calculation of three-dimensional unsteady flows in turbomachinery using the linearized harmonic Euler equations,” *J. of Turbomachinery*, vol. 115, pp. 800–809, October 1993.

[5] P. G. A. Cizmas and K. C. Hall, “Computation of steady and unsteady viscous flows using a simultaneously coupled inviscid-viscous interaction technique,” *Journal of Fluid and Structures*, vol. 9, pp. 639–657, August 1995.

[6] W. S. Clark and K. C. Hall, “A time-linearized Navier-Stokes analysis of stall flutter,” *J. of Turbomachinery*, vol. 112, pp. 467–476, 2000.

[7] D. Childs, C. Nelson, J. Sharrer, D. Elrod, and K. Hale, “Theory versus experiment for the rotordynamic coefficients of annular gas seals: Part 1 - test facillity and apparatus,” *J. of Tribology*, no. 108, pp. 426–432, 1986.

[8] D. Childs, D. Elrod, and K. Hale, “Annular honeycomb seals: Test results for leakage and rotordynamic coefficients; comparisons to labyrinth and smooth configurations,” *J. of Tribology*, no. 111, pp. 293–301, 1989.

[9] D. Childs, D. Elrod, and C. Ramsey, “Annular honeycomb seals: Additional

test results for leakage and rotordynamic coefficients," in *IFToMM, Proceedings of the Third International Conference on Rotordynamics*, (Lyon, France), pp. 303–312, 1990.

[10] T.-W. Ha, G. Morrison, and D. Childs, "Friction-factor characteristics for narrow-channels with honeycomb surfaces," *ASME Transactions - Journal of Tribology*, vol. 114, pp. 714–721, 1992.

[11] A. Roshko, "Some measurements of flow in a rectangular cutout," tech. rep., N.A.C.A. Tech. Note no. 3488, 1955.

[12] K. Krishnamurty, "Acoustic radiation from two-dimensional rectangular cutouts in aerodynamic surfaces," tech. rep., N.A.C.A. Tech. Note no. 3487, 1955.

[13] J. E. Rossiter, "Wind-Tunnel Experiments on the Flow over Rectangular Cavities at Subsonic and Transonic Speeds," tech. rep., Ministry of Aviation, 1966.

[14] C. K. W. Tam and P. J. W. Block, "On the tones and pressure oscillations induced by flow over rectangualr cavities," *J. Fluid Mech.*, vol. 89, no. 2, pp. 373–399, 1978.

[15] L. F. East, "Aerodynamically induced resonance in rectagular cavities," *Journal of Sound and Vibration*, vol. 3, no. 3, pp. 277–287, 1966.

[16] H. H. Heller, D. G. Holmes, and E. E. Convert, "Flow-induced pressure oscillations in shallow cavities," *Journal of Sound and Vibration*, vol. 18, no. 4, pp. 545–553, 1971.

[17] H. H. Heller and D. B. Bliss, "The physical mecahnism of flow-induced pressure fluctuations in cavities and concepts for their suppression," in *2nd AIAA Aero-Acoustics Conference*, AIAA Paper 75-491, (Hampton, VA), March 1975.

[18] H. Heller and J. Delfs, "Cavity pressure oscillations: The generating mechanism visualized," *Journal of Sound and Vibration*, vol. 196, no. 2, pp. 248–252, 1996.

[19] J. Henderson, K. J. Badcock, and B. E. Richards, “Understanding subsonic and transonic open cavity flows and suppression of cavity tones,” in *38th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2000-0658, (Reno, NV), January 2000.

[20] G. R. Srinivasan, “Acoustics and unsteady flow of telescope cavity in an airplane,” *Journal of Aircraft*, vol. 37, no. 2, pp. 274–281, 2000.

[21] B. Henderson, “Automobile noise involving feedback-sound generation by low speed cavity flows,” in *Third Computational Aeroacoustics Workshop on Benchmark Problems*, NASA/CP-2000-209790, pp. 95–100, 2000.

[22] H. Ahn, F.-S. Lien, J. Larsson, and J. Hines, “Simulation of aeroacoustic resonance in a deep cavity with grazing flow using a pressure-based solver,” *International Journal of Computational Fluid Dynamics*, vol. 22, pp. 39–47, Jan.-Feb. 2008. DOI: 10.1080/10618560701733608.

[23] S. M. Grace, “An Overview of Computational Aeroacoustic Techniques Applied to Cavity Noise Prediction,” AIAA Paper 2001-0510, 2001.

[24] A. Rona and E. J. Brooksbank, “POD analysis of cavity flow instability,” in *41st Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2003-178, January 2003.

[25] L. S. Ukeiley, J. M. Seiner, S. Arunajatesan, N. Sinha, and S. Dash, “Low-dimensional description of resonating cavity flow,” AIAA Paper 2000-2459, June 2000.

[26] N. S. Dougherty, J. B. Holt, T. E. Nesman, and R. A. Farr, “Time-accurate Navier-Stokes computations of self-excited two-dimensional unsteady cavity flows,” in *28th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 90-691, (Reno, NV), January 1990.

[27] L. Larcheveque, P. Sagaut, I. Mary, and O. Labbe, “Large-eddy simulation of

a compressible flow past a deep cavity," *Physics of Fluids*, vol. 15, pp. 193–210, January 2003.

[28] *Computational Fluid Dynamics: Principles and Applications.* Elsevier, 2005.

[29] *Turbulence Modeling for CFD.* La Canada, California: DCW Industries, 1993.

[30] F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA Journal*, vol. 32, no. 8, pp. 1598–1605, 1994.

[31] *Elements of Gas Dynamics.* New York: John Wiley & Sons, 1957.

[32] J. F. Thompson and N. P. Weatherill, "Aspects of numerical grid generation: Current science and art," No. AIAA-93-3539, August 1993.

[33] J. Blazek, S. Irmisch, and A. Haselbacher, "Unstructured mixed-grid navier-stokes solver for turbomachinery applications," AIAA Paper 99-0664, (Reno, NV), January 1999.

[34] A. Jameson, W. Schmidt, and E. Turkel, "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.

[35] B. van Leer, "Flux-vector splitting for the euler equations," in *8th Int. Conf. on Numerical Methods in Fluid Dynamics*, pp. 507–512, 1982.

[36] J. L. Thomas, B. van Leer, and R. W. Walters, "Implicit flux-split schemes for the euler equations," AIAA 85-1680, 1985.

[37] W. K. Anderson, J. L. Thomas, and B. van Leer, "A comparison of finite volume flux vector splittings for the euler equations," *AIAA Journal*, vol. 24, pp. 1453–1460, 1986.

[38] M. S. Liou and C. J. S. Jr., "A new flux splitting scheme," *J. Computational Physics*, vol. 107, pp. 23–39, 1993.

[39] M. S. Liou, "On a new class of flux splittings," in *13th Int. Conf. on Numerical Methods in Fluid Dynamics*, 1992.

[40] Y. Wada and M. S. Liou, "A flux splitting scheme with high-resolution and robustness for discontinuities," AIAA 94-0083, 1994.

[41] A. Jameson, "Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flow," AIAA 93-3559, 1993.

[42] A. Jameson, "Positive schemes and shock modelling for compressible flow," *Int. J. Numerical Methods in Fluids*, vol. 20, pp. 743–776, 1995.

[43] A. Jameson, "Analysis and design of numerical schemes for gas dynamics ii: Artificial diffusion and discrete shock structure," *Int. J. Computational Fluid Dynamics*, vol. 5, pp. 1–38, 1995.

[44] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, pp. 357–372, 1981.

[45] S. Osher and F. Solomon, "Upwind difference schemes for hyperbolic systems of conservation laws," *Math. Comp.*, vol. 38, pp. 339–374, 1982.

[46] K. Xu, "Does perfect Riemann solver exist?," in *14th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 99-3344, (Norfolk, VA), June 1999.

[47] J. Gressier and J. M. Moschetta, "On the pathological behavior of upwind schemes," in *36th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 98-0110, (Reno, NV), January 1998.

[48] S. Kim, C. Kim, O. Rho, and S. Hong, "Cure for shock instability: Development of an improved roe scheme," in *40th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2002-0548, (Reno, NV), January 2002.

[49] K. M. Peery and S. T. Imlay, "Blunt-body flow simulations," AIAA Paper 88-2904, 1988.

[50] A. Harten, "Self adjusting grid methods for one dimensional hyperbolic conservation laws," *Journal of Computational Physics*, vol. 50, pp. 235–269, 1983.

[51] A. Haselbacher and J. Blazek, "Accurate and Efficient Discretization of Navier-Stokes Equations on Mixed Grids," *AIAA Journal*, vol. 38, pp. 2094–2102, November 2000.

[52] T. J. Barth, "Aspects of unstructured grids and finite-volume solvers for the euler and navier-stokes equations," von Karman Institute Lecture Series 1994-05, 1994.

[53] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical Recipes in FORTRAN - The Art of Scientific Computing.* Cambridge, second ed., 1992.

[54] G. H. Golub and C. F. van Loan, *Matrix Computation.* Baltimore: The Johns Hopkins University Press, second ed., 1989.

[55] T. J. Barth and D. C. Jesperson, "The design and application of upwind schemes on unstructured meshes," AIAA Paper 89-0366, 1989.

[56] B. van Leer, "Towards the ultimate conservative difference scheme. v. a second order sequel to godunov's method," *J. Computational Physics*, vol. 32, pp. 101–136, 1979.

[57] V. Venkatakrishnan, "On the accuracy of limiters and convergence to steady state solutions.," AIAA Paper 93-0880, 1993.

[58] V. Venkatakrishnan, "Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters," *Journal of Computational Physics*, vol. 118, pp. 120–130, 1995.

[59] R. Courant, K. O. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *Math. Ann.*, vol. 100, pp. 32–74, 1928. Transl.: IBM Journal, 11 (1967) pp. 215-234.

[60] J. D. Anderson, *Computational Fluid Dynamics: The Basics with Applications.* New York: McGraw-Hill, 1995.

[61] P. Vijayan and Y. Kallinderis, "A 3d finite-volume scheme for the euler equations on adaptive tetrahedral grids," *J. of Computational Physics*, vol. 113, pp. 249–267, 1994.

[62] P. Sonneveld, "CGS, a fast lanczos-type solver for nonsymmetric linear systems," *SIAM J. Scientific and Statistical Computing*, vol. 10, pp. 36–52, 1989.

[63] H. A. V. der Vorst, "BiCGSTAB: A fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Scientific and Statistical Computing*, vol. 13, pp. 631–644, 1992.

[64] R. W. Freund, "A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems," *SIAM J. Scientific Computing*, vol. 14, pp. 470–482, 1993.

[65] Y. Saad, "GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems," *SIAM J. Scientific and Statistical Computing*, vol. 7, pp. 856–869, 1986.

[66] Y. Saad, "Krylov subspace techniques, conjugate gradients, preconditioning and sparse matrix solvers," von Karman Institute Lecture Series 1994-05, 1994.

[67] H. F. Walker, "Implementation of the GMRES method using householder transformations," *SIAM J. Scientific Computing*, vol. 9, p. 152, 1988.

[68] R. B. Morgan, "A restarted GMRES method augmented with eigenvectors," *SIAM J. Matrix Anal. Appl.*, vol. 16, pp. 1154–1171, 1995.

[69] A. Chapman and Y. Saad, "Deflated and augmented krylov subspace techniques," Tech. Rep. UMSI 95/181, Minnesota Supercomputer Institute, 1995.

[70] S. Yoon and A. Jameson, "A multigrid lu-ssor scheme for approximate newton-iteration applied to the euler equations.," tech. rep., NASA, 1986. CR-17954.

[71] S. Yoon and A. Jameson, "Lower-upper symmetric-gauss-seidel method for the euler and navier-stokes equations," *AIAA Journal*, vol. 26, pp. 1025–1026,

1988. AIAA Paper 87-0600.

[72] S. Yoon and A. Jameson, “Lu implicit schemes with multiple grids for the euler equations,” *AIAA Journal*, vol. 7, pp. 929–935, 1987. AIAA Paper 86-0105.

[73] R. F. Chen and Z. J. Wang, “Fast, block lower-upper symmetric gauss-seidel scheme for arbitrary grids,” *AIAA Journal*, vol. 38, pp. 2238–2245, December 2000.

[74] J. S. Kim and O. J. Kwon, “An efficient implementation of implicit operator for block lu-sgs method,” *Computational Fluid Dynamics J.*, pp. 154–159, July 2005.

[75] A. Arnone, M. S. Liou, and L. A. Povinelli, “Multigrid time-accurate integration of navier-stokes equations,” AIAA Paper 93-3361, 1993.

[76] N. D. Melson, M. D. Sanetrik, and H. L. Atkins, “Time-accurate navier-stokes calculations with multigrid acceleration,” in *6th Copper Mountain Conf. on Multigrid Methods*, pp. 423–439, 1993.

[77] V. Venkatakrishnan and D. J. Mavriplis, “Implicit method for the computation of unsteady flows on unstructured grids,” *Journal of Computational Physics*, vol. 127, pp. 380–397, 1996.

[78] A. Jameson, “Time-dependent calculations using multigrid with applications to unsteady flows past airfoils and wings,” AIAA Paper 91-1596, 1991.

[79] J. J. Alonso and A. Jameson, “Fully-implicit time-marching aeroelastic solution,” AIAA Paper 94-0056, 1994.

[80] A. Belov, L. Martinelli, and A. Jameson, “A new implicit algorithm with multigrid for unsteady incompressible flow calculations,” AIAA Paper 95-0049, 1995.

[81] N. A. Pierce and J. J. Alonso, “A preconditioned implicit multigrid algorithm for parallel computation of unsteady aeroelastic compressible flows,” AIAA Paper 97-0444, 1997.

[82] K. P. Singh, J. C. Newman, and O. Baysal, "Dynamic unstructured method for flows past multiple objects in relative motion," *AIAA Journal*, vol. 33, pp. 641–649, 1995.

[83] L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K. J. Badcock, and B. E. Richards, "Solution of the unsteady euler equations using an implicit dual-time method," *AIAA Journal*, vol. 36, pp. 1417–1424, 1998.

[84] T. H. Pulliam, "Time accuracy and the use of implicit methods," AIAA Paper 93-3360, 1993.

[85] R. E. Bartels, "An elasticity-based mesh scheme applied to the computation of unsteady three-dimensional spoiler and aeroelastic problems," AIAA Paper 99-3301, 1999.

[86] *Introduction to Perturbation Techniques.* Wiley-VCH, 2004.

[87] J. T. Batina, "Unsteady euler airfoil solutions using unstructured dynamic meshes," AIAA Paper 89-0115, 1989.

[88] Y. Tamura and K. Fujii, "Conservation law for moving and transformed grids," AIAA Paper 93-3365, 1993.

[89] M. Lesoinne and C. Farhat, "Geometric conservation laws for flow problems with moving boundaries and deformable meshes and their impact on aeroelastic computations," *Comp. Meth. Appl. Mech. Eng.*, vol. 134, pp. 71–90, 1996.

[90] H. Guillard and C. Farhat, "On the significance of the gcl for flow computations on moving meshes," AIAA Paper 99-0793, 1999.

[91] P. D. Thomas and C. K. Lombard, "Geometric conservation law and its application to flow computations on moving grids," *AIAA Journal*, vol. 17, pp. 1030–1037, October 1979.

[92] *Turbomachinery Rotordynamics: Phenomena, Modeling, and Analysis.* New York: John Wiley & Sons, 1993.

[93] D. G. Holmes and S. S. Tong, "A Three-Dimensional Euler Solver for Turbomachinery Blade Rows," *Journal of Engineering for Gas Turbines and Power*, vol. 107, pp. 258–264, 1985.

[94] D. G. Holmes and S. D. Connell, "Solution of the 2d Navier-Stokes equations on unstructured adaptive grids," in *9th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 89-1932, (Buffalo, NY), June 1989.

[95] D. G. Holmes, "Inviscid 2d solutions on unstructured, adaptive grids.," *Numerical Methods for Flows in Turbomachinery*, vol. VKI-lS 1989-06, 1989.

[96] M. B. Giles, "Nonreflecting boundary conditions for Euler equation calculations," *AIAA Journal*, vol. 28, pp. 2050–2058, December 1990.

[97] W. S. Clark and K. C. Hall, "A time-linearization Navier-Stokes analysis of stall flutter," in *International Gas Turbine & Aeroengine Congress & Exhibition*, ASME Paper 99-GT-383, (Indianapolis, Indiana), ASME, June 1999.

[98] K. C. Hall, C. B. Lorence, and W. S. Clark, "Nonreflecting boundary conditions for linearized unsteady aerodynamic calculations," in *31st Aerospace Sciences Metting & Exhibit*, AIAA 93-0882, (Reno, NV), January 1993.

[99] W. W. Kim and S. Menon, "Application of the localized dynamic subgrid-scale model to turbuelnt wall-bounded flows," Tech. Rep. AIAA-97-0210, AIAA, January 1997.

[100] D. Liliedahl and P. Cizmas, "Prediction of fluid instabilities in hole-pattern stator seals," in *47th AIAA Aerospace Sciences Meeting*, AIAA Paper 2009-0786, (Orlando, FL), January 2009.

[101] C. C. Nelson, "Rotordynamic coefficients for compressible flow in tapered annular seals," *J. of Tribology*, vol. 107, pp. 318–325, July 1985.

[102] N. G. Wagner, K. Steff, R. Gausmann, and M. Schmidt, "Investigations on the dynamic coefficients of impeller eye labyrinth seals," in *38th Turbomachinery*

Symposium, 2009.

[103] K. C. Hall, J. P. Thomas, and W. S. Clark, "Computation of unsteady nonlinear flows in cascades using a harmonic balance technique.," *AIAA Journal*, vol. 40, pp. 879–886, 2002.

[104] E. Kivanc and K. C. Hall, "Nonlinear analysis of unsteady flows in multistage turbomachines using harmonic balance.," *AIAA Journal*, vol. 45, pp. 1047–1057, 2007.