High Fidelity Time Accurate CFD Analysis of a Multi-stage Turbofan at Various Operating Points in Distorted Inflow

David Bruce Weston
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd
Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation
Weston, David Bruce, "High Fidelity Time Accurate CFD Analysis of a Multi-stage Turbofan at Various Operating Points in Distorted Inflow" (2014). All Theses and Dissertations. 5604.
https://scholarsarchive.byu.edu/etd/5604

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
High Fidelity Time Accurate CFD Analysis of a Multi-stage Turbofan
at Various Operating Points in Distorted Inflow

David B. Weston

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Steven E. Gorrell, Chair
R. Daniel Maynes
Scott L. Thomson

Department of Mechanical Engineering
Brigham Young University
June 2014

Copyright © 2014 David B. Weston
All Rights Reserved
ABSTRACT

High Fidelity Time Accurate CFD Analysis of a Multi-stage Turbofan at Various Operating Points in Distorted Inflow

David B. Weston
Department of Mechanical Engineering, BYU
Master of Science

Inlet distortion is an important consideration in fan performance. Distortion can be caused through flight conditions and airframe-engine interfaces. The focus of this paper is a series of high-fidelity time accurate Computational Fluid Dynamics (CFD) simulations of a multistage fan. These investigate distortion transfer and generation as well as the underlying flow physics of these phenomena under different operating conditions. The simulations are performed on the full annulus of a 3 stage fan. The code used to carry out these simulations is a modified version of OVERFLOW 2.2 developed as part of the Computational Research and Engineering Acquisition Tools and Environment (CREATE) program. Several modifications made to the code are described within this thesis. The inlet boundary condition is specified as a 1/rev total pressure distortion. Simulations at choke, design, and near stall points are analyzed and compared to experimental data. Analysis includes the phase and amplitude of total temperature and pressure distortion through each stage of the fan and blade loading plots. An understanding of the flow physics associated with distorted flows will help designers account for unsteady flow physics at design and off-design operating conditions and build more robust fans with a greater stability margin.

Keywords: turbofan, CFD, engine stall, inlet distortion
ACKNOWLEDGMENTS

I would like to thank CREATE-AV for the funding they provided for this project. I would also like to thank the Department of Defense HPC Modernization Program and Brigham Young University’s Fulton Supercomputing Laboratory for the computational resources they provided. The computational hours on these machines were vital to produce the results presented in this thesis. I would like to thank Robert Nichols and Jason Klepper for the Arnold Engineering Development Center for their support of this project. They provided several vital modifications to the OVERFLOW code as well as creating the structured grid used for the simulations. I also want to acknowledge Carol Wallis and Joe Capozzi from GE Aviation for providing the geometry of the fan as well as validation data for the fan. I especially would like to thank Dr. Steven Gorrell for his help and guidance through my Master’s program, as well as Dr. Daniel Maynes and Dr. Scott Thomson. Trevor Blanc, Matthew Marshall, and Mark Fernelius provided significant support for my research. I would also like to express my sincere gratitude to my wife Meredith and daughter Lily for their patience and support during the completion of this program.
TABLE OF CONTENTS

LIST OF TABLES .......................................................... vi

LIST OF FIGURES .......................................................... vii

Chapter 1 Introduction .................................................. 1

Chapter 2 Literature Review ............................................. 5
  2.1 Flow Distortion .................................................. 5
  2.2 Engine Flow Instabilities ...................................... 6
  2.3 Reduced Order Models ......................................... 10
  2.4 CFD Simulations ................................................ 12

Chapter 3 Computational Methods ....................................... 15
  3.1 Solver .............................................................. 15
  3.2 Secondary Programs ............................................ 18
  3.3 Computational Domain ......................................... 19
  3.4 Boundary Conditions .......................................... 22
    3.4.1 Walls ..................................................... 22
    3.4.2 Inlet ..................................................... 22
    3.4.3 Exit ....................................................... 24

Chapter 4 Results and Discussion ...................................... 25
  4.1 Time Averaged Results ......................................... 25
  4.2 Distortion Transfer and Generation ............................ 28
  4.3 Blade Loading .................................................. 34
  4.4 Distortion Levels .............................................. 47
  4.5 Stall Inception ................................................ 49

Chapter 5 Conclusion .................................................... 54
  5.1 Future Work ..................................................... 55

REFERENCES .............................................................. 56

Appendix A Boundary Condition Modifications ......................... 58

Appendix B Secondary Programs ......................................... 73
  B.1 Time Accurate Data Extraction ................................. 73
  B.2 TAFE Modification ............................................. 77
    B.2.1 POD Data Extraction ..................................... 77
    B.2.2 Blade Loading Plots ..................................... 78
  B.3 Case Specific Matlab Coding ................................... 79
LIST OF TABLES

3.1  Mesh Sizes ................................................................. 22

4.1  Nozzle Area Reduction Factor ........................................ 26
4.2  Peak to Peak Magnitudes - 30% Immersion ......................... 30
4.3  Distortion Phase - 30% Immersion ................................... 31
4.4  Coefficient of Determination for Sine Fit .......................... 45
<table>
<thead>
<tr>
<th>FIGURE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>General Cross Section of a Turbofan Engine</td>
</tr>
<tr>
<td>1.2</td>
<td>Sample Distortion Patterns</td>
</tr>
<tr>
<td>1.3</td>
<td>Possible Causes of Inlet Distortion</td>
</tr>
<tr>
<td>2.1</td>
<td>Sample Compressor Speedlines with Stall Margin</td>
</tr>
<tr>
<td>2.2</td>
<td>Types of Rotating Stall</td>
</tr>
<tr>
<td>2.3</td>
<td>Formation of Rotating Stall</td>
</tr>
<tr>
<td>2.4</td>
<td>Deep Surge</td>
</tr>
<tr>
<td>2.5</td>
<td>Comparison of Bow Shock Location [7]</td>
</tr>
<tr>
<td>2.6</td>
<td>Parallel Compressor Model</td>
</tr>
<tr>
<td>2.7</td>
<td>NASA Stage 35 Geometry</td>
</tr>
<tr>
<td>3.1</td>
<td>Overset Airfoil</td>
</tr>
<tr>
<td>3.2</td>
<td>Computational Domain</td>
</tr>
<tr>
<td>3.3</td>
<td>Total Pressure Contours in both Inlet Grids</td>
</tr>
<tr>
<td>3.4</td>
<td>Inlet Distortion and Variations through Fan</td>
</tr>
<tr>
<td>4.1</td>
<td>Time-Averaged Flow Characteristics</td>
</tr>
<tr>
<td>4.2</td>
<td>Circumferential Plots for 30% immersion at the Design Point</td>
</tr>
<tr>
<td>4.3</td>
<td>Circumferential Plots at 30% immersion at the Near Stall Point</td>
</tr>
<tr>
<td>4.4</td>
<td>Swirl Generation</td>
</tr>
<tr>
<td>4.5</td>
<td>Pressure Contours at 30% Immersion for the Near Stall Simulation</td>
</tr>
<tr>
<td>4.6</td>
<td>Blade Loading Profiles in Rotor 1 Stage, 30% Immersion, Peak Efficiency Point</td>
</tr>
<tr>
<td>4.7</td>
<td>Blade Loading Profiles in Rotor 1 Stage, 30% Immersion, Near Stall Point</td>
</tr>
<tr>
<td>4.8</td>
<td>Blade Loading Profiles in Rotor 2 Stage, 30% Immersion, Peak Efficiency Point</td>
</tr>
<tr>
<td>4.9</td>
<td>Blade Loading Profiles in Rotor 2 Stage, 30% Immersion, Near Stall Point</td>
</tr>
<tr>
<td>4.10</td>
<td>Blade Loading Profiles in Rotor 3 Stage, 30% Immersion, Peak Efficiency Point</td>
</tr>
<tr>
<td>4.11</td>
<td>Blade Loading Profiles in Rotor 3 Stage, 30% Immersion, Near Stall Point</td>
</tr>
<tr>
<td>4.12</td>
<td>Rotor 3 Blade at 70% Immersion with Attached Bow Shock</td>
</tr>
<tr>
<td>4.13</td>
<td>Rotor 3 Blade at 30% Immersion with Attached Bow Shock</td>
</tr>
<tr>
<td>4.14</td>
<td>Potential Effect - Contours of Static Pressure through Rotor 1</td>
</tr>
<tr>
<td>4.15</td>
<td>Vorticity Contours from Stator 1 and Rotor 2 Blade Rows at Near Stall</td>
</tr>
<tr>
<td>4.16</td>
<td>Circumferential Variation of Blade Work at 10% Immersion</td>
</tr>
<tr>
<td>4.17</td>
<td>Circumferential Variation of Blade Work at 50% Immersion</td>
</tr>
<tr>
<td>4.18</td>
<td>Circumferential Variation of Blade Work at 90% Immersion</td>
</tr>
<tr>
<td>4.19</td>
<td>Sample Sine Fit Curves</td>
</tr>
<tr>
<td>4.20</td>
<td>Work Variation for Uniform Flow at 50% Immersion</td>
</tr>
<tr>
<td>4.21</td>
<td>Distortion Levels at 10% Immersion</td>
</tr>
<tr>
<td>4.22</td>
<td>Distortion Levels at 50% Immersion</td>
</tr>
<tr>
<td>4.23</td>
<td>Distortion Levels at 90% Immersion</td>
</tr>
<tr>
<td>4.24</td>
<td>Part-Span Stall Cells - Iso-Surface of Zero Massflow Colored by Total Pressure</td>
</tr>
<tr>
<td>4.25</td>
<td>Stall Structure in Rotor 2 - Iso-Surface of Zero Massflow Colored by Total Pressure</td>
</tr>
<tr>
<td>4.26</td>
<td>Property Comparison from Stator 1 to Rotor 2</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>C.1</td>
<td>Blade Loading in the First Rotor Stage at 70% Immersion - Peak Efficiency</td>
</tr>
<tr>
<td>C.2</td>
<td>Blade Loading in the First Rotor Stage at 30% Immersion - Peak Efficiency</td>
</tr>
<tr>
<td>C.3</td>
<td>Blade Loading in the First Rotor Stage at 70% Immersion - Near Stall</td>
</tr>
<tr>
<td>C.4</td>
<td>Blade Loading in the First Rotor Stage at 30% Immersion - Near Stall</td>
</tr>
<tr>
<td>C.5</td>
<td>Blade Loading in the Second Rotor Stage at 70% Immersion - Peak Efficiency</td>
</tr>
<tr>
<td>C.6</td>
<td>Blade Loading in the Second Rotor Stage at 30% Immersion - Peak Efficiency</td>
</tr>
<tr>
<td>C.7</td>
<td>Blade Loading in the Second Rotor Stage at 70% Immersion - Near Stall</td>
</tr>
<tr>
<td>C.8</td>
<td>Blade Loading in the Second Rotor Stage at 30% Immersion - Near Stall</td>
</tr>
<tr>
<td>C.9</td>
<td>Blade Loading in the Third Rotor Stage at 70% Immersion - Peak Efficiency</td>
</tr>
<tr>
<td>C.10</td>
<td>Blade Loading in the Third Rotor Stage at 30% Immersion - Peak Efficiency</td>
</tr>
<tr>
<td>C.11</td>
<td>Blade Loading in the Third Rotor Stage at 70% Immersion - Near Stall</td>
</tr>
<tr>
<td>C.12</td>
<td>Blade Loading in the Third Rotor Stage at 30% Immersion - Near Stall</td>
</tr>
<tr>
<td>C.13</td>
<td>Stagnation Temperature Circumferential Variation in the 30% Nozzle</td>
</tr>
<tr>
<td>C.14</td>
<td>Stagnation Pressure Circumferential Variation in the 30% Nozzle</td>
</tr>
<tr>
<td>C.15</td>
<td>Static Temperature Circumferential Variation in the 30% Nozzle</td>
</tr>
<tr>
<td>C.16</td>
<td>Static Pressure Circumferential Variation in the 30% Nozzle</td>
</tr>
<tr>
<td>C.17</td>
<td>Stagnation Temperature Circumferential Variation in the 37% Nozzle</td>
</tr>
<tr>
<td>C.18</td>
<td>Stagnation Pressure Circumferential Variation in the 37% Nozzle</td>
</tr>
<tr>
<td>C.19</td>
<td>Static Temperature Circumferential Variation in the 37% Nozzle</td>
</tr>
<tr>
<td>C.20</td>
<td>Static Pressure Circumferential Variation in the 37% Nozzle</td>
</tr>
<tr>
<td>C.21</td>
<td>Stagnation Temperature Circumferential Variation in the 39% Nozzle</td>
</tr>
<tr>
<td>C.22</td>
<td>Stagnation Pressure Circumferential Variation in the 39% Nozzle</td>
</tr>
<tr>
<td>C.23</td>
<td>Static Temperature Circumferential Variation in the 39% Nozzle</td>
</tr>
<tr>
<td>C.24</td>
<td>Static Pressure Circumferential Variation in the 39% Nozzle</td>
</tr>
</tbody>
</table>
CHAPTER 1. INTRODUCTION

Gas turbine engines are very complex pieces of machinery that are becoming more and more prevalent in our society today. We depend on turbine engines for a large portion of our transportation and power generation. Nearly all commercial and military air travel uses turbine engines. Helicopters generally use a turbine engine to generate power for their rotor blades. Gas turbine engines for power generation applications are becoming more common and efficient. Combined cycle power generation is the combination of two different thermodynamic cycles, and allows for greater efficiency than either cycle alone could obtain. The combination of Brayton and Rankine cycles (essentially a large jet engine that turns a generator and uses the heat from the exhaust to drive a steam turbine) provides a very efficient means of providing large amounts of power. These plants are driven by natural gas, which was responsible for 36% of the power generation in the United States in 2012. Smaller turbine powerplants can also be used to generate power in applications such as offshore oil rigs and other remote locations. Gas turbines have become so common because, while they are technically difficult to construct, they provide excellent power output at a high efficiency and low relative weight.

Though gas turbine engines are becoming more and more prevalent, our understanding of the flow phenomena within them is still incomplete. This is hardly surprising when considering the complexity of a turbomachine. The technology of a gas turbine engine requires expertise in nearly every field of engineering, and the challenges in the field of aerodynamics alone are daunting. A turbine engine is filled with complex, heavily time-dependent flows including viscous vortex formation, moving shock waves of various strengths,
and flow separation. Each of these flow structures individually can be quite difficult to analyze, and the combination of all of them in an unsteady environment makes the performance of a turbo-machine very difficult to analyze without a very powerful solver or a very simplified model.

Non-uniform inflow conditions to gas turbine engines magnify the complexity of flow through a turbine engine. These conditions can be caused by any variety of events ranging from routine flight maneuvers to military weapon gas ingestion to complex inlet designs. Any non-uniformity in the inflow is referred to as a distortion. Distortion is generally defined as a variation in the pressure, temperature, or swirl of the flow. To add to their complexity, distortions can have various shapes and extents. (Figure 1.2 shows the total pressure contours of several patterns that can be produced by an aircraft inlet.) Circumferential distortions are (quite understandably) those where the variation occurs mainly in the circumferential direction, while radial distortions vary in the radial direction. Compound distortion patterns can have a variation in both the circumferential and radial directions, and irregular patterns have no particular trend. The type of pattern produced depends on the condition producing it, but the wide variety of applications for which gas turbines are used ensures that nearly every type of distortion possible will be seen by some gas turbine somewhere.

Distorted flows can affect engine stability, performance, and structural integrity. They also decrease the stall margin, which defines the point where the fan/compressor blades begin to stall and the entire component loses compression. With the advent of blended-wing body airframes
(which cause the engines to ingest the boundary layer from the airframe) and the increased use of advanced inlet designs (which generally cause separation that the engine ingests), distorted inlet flow is becoming more common in “everyday” use of gas turbines (see Figure 1.3). As such, predicting fan performance under distorted flows is essential for the safe and efficient design of propulsion systems. Computational fluid dynamics (CFD) can be used to simulate the response and performance of the fan and compressor under a wide range of distortion patterns.

Figure 1.3: Possible Causes of Inlet Distortion

CFD provides an excellent opportunity to study the flow physics in a turbine engine. Experimentation provides a more inherently accurate view of the flow in an engine, since the data produced in the experimentation reflects the physical flow through the engine. However, the time, expertise, and material required for an engine test make it very expensive. Additionally, there is only so much instrumentation that can be put into an engine before it begins to interfere with the normal flow. This puts a hard upper limit on the amount of data that can be collected from an experimental analysis of flow phenomena in a turbine engine. While CFD simulations can be quite expensive depending on the level of fidelity required, CFD simulations don’t suffer from a limitation on the amount of data than can be extracted. Once a simulation is performed, any point in the flow domain can be probed and analyzed. The only limit to the amount of data that can be produced through CFD simulation is the density of the grid used to produce the data, and this makes this type of analysis ideal for studying the intricate interactions between the various components of a turbine engine. In this process, of course, care must be taken to ensure an accurate simulation.
before drawing conclusions from the results. Once this validation has taken place, though, CFD can be used to follow entire structures in the flow to determine their effect on the turbine.

This project was initiated by the CREATE-AV project, which is a government funded group tasked to create a series of analysis tools to aid in the optimal integration of airframe inlets with propulsion systems. The contract to Brigham Young University (BYU) was to validate the OVERFLOW 2.2 code for a multi-stage turbofan in distorted flow. The manufacturer of the fan has been involved in this research as well, since they have an interest in protecting their information and investigating the physics in their product. This thesis primarily addresses the latter purpose, which is to form a greater understanding of the effect that a particular type of total pressure distortion has on the performance of the fan as it approaches stall.

A preliminary task required for completion of this thesis was to validate the OVERFLOW 2.2 code used for this research for use with a multistage turbofan. Experimental data was provided by the manufacturer of the turbofan from a series of tests that they performed in the development of the engine. Validating the code required that simulations that matched the operating conditions of the engine test be performed and then compared to the experimental data. The simulation data necessary for validation is determined by the data available in the experimental data file. Once the simulation has completed, it is relatively simple to extract the required data and compare it. This task was completed previous to the beginning of this line of research by the author and Matthew Marshall.

The goal for this thesis was to investigate the flow physics of distortion transfer and generation, as well as the change in the stability limit of the fan under distorted flow. While this task offered more freedom for exploration, it also presented a greater challenge, as the direct cause of the decreased stability of a fan in distorted flow is unknown. This required the creation of several new methods of analyzing the data in an attempt to isolate the difference between a stable and unstable flow. This is a significant analysis because of the amount of detail included. It marks the first time that a multistage fan has been simulated through various operating points approaching stall. This permits the study of the changes in the way the fan interacts with the distortion pattern at different conditions, which is a precursor to a deeper understanding of how compressor stall begins.
CHAPTER 2. LITERATURE REVIEW

This section will explain previous research on distortion patterns and how inlet distortion is generated. It will describe flow instabilities in turbomachinery and some research that has been performed into the physics of flow instability. Brief mention will be made of some simplified models that are used in distorted turbomachinery analysis. Finally, previous CFD modeling attempts in the field of distorted flow and stall prediction will be summarized.

2.1 Flow Distortion

Flow distortion refers to any non-uniformity in the inlet flow to a turbine engine. It can consist of variations in the velocity, pressure, or temperature at the inlet. This thesis is concerned only with a total pressure distortion pattern. The specific pattern is one that reflects the total pressure seen after a serpentine inlet duct. However, the experimental counterpart of the distortion used in this study was generated by a set of wire mesh screens of varying density, and as such does not contain the counter-rotating vortices seen in a serpentine inlet. The distortion pattern varies only in the circumferential direction, and is referred to as a 1/rev distortion since it only varies from high to low pressure once. The research presented here is concerned with serpentine inlet distortion, because although the specific pattern used for this thesis was not exactly what is created by a serpentine inlet, the ultimate goal of this type of research is to understand the effect that the distortion from a peculiarly shaped inlet will have on compressor performance.

Brear et al. performed an experimental study of the flow structures within the serpentine inlet of a uninhabited combat air vehicle (UCAV) [1]. They used a standard 40 probe total pressure rake to measure the annular pressure distortion after the duct and an oil and charcoal surface coating to visualize streamlines of the flow through the duct. The authors found that the separation on the top surface of the inlet was the main contributor to pressure distortion and recovery loss at the fan face. The separation also created vortices that were responsible for large-scale unsteadiness in the
flow at the fan face. These effects would be magnified by shortening the duct or operating it at a higher inlet Mach number.

Kirk et al. performed a validation study of flow through an ultra-compact serpentine inlet designed by Lockheed Martin [2]. They were able to qualitatively match the experimental data from the inlet, but were not able to correctly predict the specific details of the flow. The details varied based on the solver and grid used, and the separation points of the flow as well as the vortical structures within the duct proved difficult to simulate in certain cases. However, this work contains important data for any future work in performing coupled fan-inlet flow simulations.

Chima performed two CFD studies using a coupled fan-inlet flow domain to predict performance in distorted flow. The first was a validation study of a serpentine inlet/fan coupled flow simulator at NASA’s Global Research Center [3]. He successfully coupled a simulation of a serpentine inlet to a simplified fan stage and predicted the speedline for the fan under those conditions. While using a coarse grid, Chima was still able to predict the characteristic map of the fan under distorted conditions. This study shows that it is possible to couple a fan and an inlet together, rather than setting a pre-defined boundary condition to create distorted flow.

The second study was a coupled analysis of a fan and inlet for a hypothetical Quiet Supersonic Jet (QSJ) [4]. He used an axisymmetric code to model both the inlet and fan and analyze their performance. The grid used for this study was extremely coarse, and therefore the details of the flow were not studied extensively. Chima was able to determine the overall flow characteristics around the inlet and estimate the performance of the fan when subjected to the distortion pattern from the inlet. While this study is not conclusive in any manner, it demonstrates the potential capability of CFD simulations to both predict inlet distortion based on hypothetical geometry and analyze its effect on a compressor. This would be a great design tool to encourage efficient coupling of airframes and propulsion systems in the future.

2.2 Engine Flow Instabilities

Engine instabilities are at best costly and at worst dangerous to equipment and human operators. Some instabilities merely decrease the performance of a turbine engine, while others can cause severe structural damage leading to failure of the engine. Compressors are designed with a safety margin so that while they operate at less than peak performance (see Figure 2.1), they are
in much less danger of reaching an unstable flow condition. This safety factor is referred to as the stall margin and is defined by the equation

\[
Stall Margin = \frac{PR_s - PR_o}{PR_o} \times 100
\]  

(2.1)

where \( PR \) signifies the pressure ratio, subscript \( s \) signifies the stall point, and subscript \( o \) signifies the operating point. Distorted inlet flow decreases the stall margin by the following amount:

\[
\frac{PR_s - PR_{dl}}{PR_o} \times 100
\]

(2.2)

where subscript \( dl \) signifies the stability limit in distorted flow [5]. This decreased stall margin increases the occurrence of flow instabilities in cases that would not otherwise incite them.

Turbine engine instabilities generally refer to rotating stall and surge. Cumpsty provides an in depth description of both of these phenomena in his textbook on compressor aerodynamics [6]. Compressors have a maximum pressure ratio, and once they are throttled beyond that maximum the pressure ratio starts to drop. The severity of the drop depends on what type of instability the compressor experiences. Part-span rotating stall consists of several zero flow stall cells that rotate at a speed less than the rotation rate of the fan (see Figure 2.2a). The drop in pressure ratio for this particular type of stall is minimal, and generally will not greatly affect the performance of the fan. At some point the part span cells can unite to form a full span stall cell (see Figure 2.2b).
Full span stall creates a large blockage area in the compressor, which creates a drastic drop in both the pressure ratio and massflow through the compressor. This type of stall usually extends axially through the entire compressor since the flow is not able to redistribute radially and recover after passing the stalled stage. While this type of stall is generally not catastrophic, it can severely impact the performance of the compressor.

Rotating stall begins with a blade experiencing simple aerodynamic stall as pictured in Figure 2.3. This reduces the available flow area through that blade passage, forcing more air through adjacent blade passages. This increases the angle of attack on the next blade in the disk, adding to its load and causing it to stall as well. This cascades until a stall cell spanning several blade passages forms. This stall cell is generally part span, and can later develop into a more severe form of stall or surge.

Engine surge is a highly disruptive and damaging form of compressor stall. In this scenario, the massflow through the compressor varies with time. In extreme cases, the compressor can lose enough power that the previously pressurized air in the combustor will be expelled forward through the compressor stages (see Figure 2.4). This type of stall produces large transverse loads in the rotor blades as well as the rotor disc itself, and can cause severe structural damage. Engine surge is the most harmful form of engine instability, but is also the less common of the instabilities discussed here.

Wood et al. [7] studied the physical changes in the flow through a transonic compressor leading up to stall. They used laser anemometry and holography to study the changes in the structure of shock waves in a low aspect ratio fan rotor. The study showed good agreement with
previous experimental data. It was also able to show that some of the errors in previous analyses were due to the highly three-dimensional nature of the shocks in the fan being ignored. The authors found that the bow shock stays attached to the rotor blade for most of its span at the peak efficiency case. However, at the near stall case, the bow shock detaches from the rotor. The authors’ visualization of this is shown in Figure 2.5. The results of this study are significant because they provide a definite distinction between the flow through a compressor at peak efficiency and near stall conditions.
2.3 Reduced Order Models

Simplified models, while generally less accurate than high fidelity CFD simulations, provide a quick first order analysis of turbomachinery flows at a fraction of the computational time. They are also based on physical characteristics present in turbine engines, and can so be instructive in the nature of turbine engine aerodynamics.

Pearson and McKenzie first proposed the parallel compressor model for analysis of distorted flow in the late 1950’s [8]. The parallel compressor model works on the assumption that a compressor can be split into several compressors that each have a uniform inlet condition (see Figure 2.6). The performance of all compressors is averaged after performing a uniform inlet analysis on each one to provide an estimate of the overall compressor performance.

Mazzawy proposed improvements to the parallel compressor model in 1977 [9]. The original model provided an easy method of predicting compressor performance in distorted flow, but provided results inconsistent with many experimental studies of distorted flow. Mazzawy proposed and developed a method for including more divisions in the model, which allowed the calculation of unsteady and 2-dimensional effects in the compressor. The model improved the performance of
the parallel compressor model, but still left some dependence on extrapolated compressor characteristics.

In 2011, Cousins proposed several improvements to the parallel compressor model [10]. These improvements addressed several limitations of the parallel compressor model. Cousins was able to successfully find a relation that addressed crossflow issues between the parallel compressors. Perhaps the most important contribution from this study was the addition of a meanline flow calculator to the parallel compressor model. This allows the model to calculate temperature and pressure ratios of the fan for given distortion levels and operating points, rather than relying on the traditional lookup table method to determine these properties for each simulation. However, Cousins was still unable to define an equation to allow the calculation of radial distortion patterns, and this remains a difficulty for the parallel compressor model.

![Parallel Compressor Model](image)

**Figure 2.6: Parallel Compressor Model**

Other models exist as well that predict turbine performance with a reduced order estimation. Sharma et al. [11] evaluated the effectiveness of a Fourier Transformation (FT) solution method for CFD simulations in 2013. This method allows the full annulus to be effectively simulated using only 2 blade passages. The authors chose to use a full annulus circumferential distortion pattern to test the capabilities of their method, as traditional blade passage studies only consider uniform inlet flow. The FT method needed to be run for several rotor revolutions past convergence in order to assemble a complete dataset for the full annulus. The study showed a reasonably accurate reproduction of full annulus flow using only two blade passages. This method also provides a significant reduction in the computational time and cost of the simulation in general, because
although the FT method requires more rotor revolutions to obtain a complete dataset, it simulates significantly less of the full annulus grid. The authors note that use of the FT method is case dependent rather than universally applicable, and that the method filters out some of the high frequency variations seen in a full annulus simulation. This particular characteristic of the method could cause problems in a multistage simulation.

2.4 CFD Simulations

While reduced order models provide increasingly accurate estimates of turbine performance in various situations, they do not generally provide a very accurate view of the flow physics in the engine. The characteristic map and efficiency are the most important compressor characteristics to understand in a given design problem, and so most reduced order analytical models have been designed to provide those values. However, for the analysis and improvement of the flow physics through the engine in general, a more detailed simulation is required. Time accurate CFD provides the capability to examine and improve the flow in a very detailed manner, as well as to investigate flow phenomena such as stall inception and development.

Choi, Vahdati, and Imregun studied stall inception and recovery in a single stage transonic fan [12]. They altered the fan geometry with one mis-staggered blade in order to artificially trigger stall, and then studied the resulting stalled flow behavior. Several part span stall cells developed before merging into a single large stall cell that settled into a “stable” rotating stall pattern. The number of cells decreased and the size of the cells increased as the fan speed increased. They also found, interestingly, that the blade passages experience more blockage when many small stall cells are present than when they merge into a single large rotating stall cell. During the recovery process, the massflow of the fan increases without a corresponding increase in pressure ratio until the stall cells shrink to a threshold value.

Gorrell, Yao, and Wadia performed an analysis of a 1/rev total pressure distorted flow in 2008. They analyzed flow through two separate multi-stage turbofans at a single operating point. The authors used an unsteady RANS code called TURBO for the analysis. The first study they performed used only half of the stages for each fan [13]. They studied distortion transfer and generation in this preliminary study, but were only able to study them to the extent that the computational domain allowed. The second study used all three rotor/stator stages for both fans, and
studied the flow physics through the entire fan [14]. For both studies, the authors validated the simulations against experimental data before examining the flow physics. The validation showed a high degree of accuracy for the TURBO simulations. The authors studied the total temperature distortion generation due to the total pressure distortion in the fan. The authors showed that a 1/rev total temperature distortion can develop from a uniform total temperature inlet. The total pressure distortion causes a variation in the pressure ratio at different circumferential locations in the rotor stages, and the variation in pressure ratio is echoed by a similar variation in the temperature ratio. The authors were able to show that the static temperature rise and static pressure rise, when properly non-dimensionalized, displayed nearly identical trends. Another important point regarding the total temperature distortion was that it showed a nearly uniform 90° phase lag behind the total pressure distortion. This trend had been observed in experimental data previously, but this was the first study to provide an explanation for the phenomenon.

Fidalgo, Hall, and Colin performed a study on the Fan-Distortion interaction with a NASA Rotor 67 stage [15]. They used a nozzle to define the massflow at the exit of the flow domain, and in so doing provided increased numerical stability to their simulation. Their study showed that the flow redistribution caused by a variation in static pressure at the inlet was responsible for some of the distortion transfer around the circumference of the fan. They also showed that work input varied around the circumference of the fan by calculating the pressure ratio through streamtubes in the fan. This analysis showed that the pressure ratio for individual sections of the flow was higher and lower than the overall pressure ratio of the fan depending on the location in the distortion pattern, which demonstrated the circumferential work variation quite clearly.

Chen performed a CFD simulation to examine the development of rotating stall in a single stage compressor [16]. He used the TURBO code to simulate a NASA Stage 35 (see Figure 2.7) with a massflow exit boundary condition to allow for pressure excursions expected in a stalled flow. Chen was able to successfully simulate the development and inception of rotating stall in the compressor stage. He found that the tip clearance vortex of the rotor stage was shed in a more axial direction as the stage approached stall, and the shock system switched to a dual shock instead of a single attached oblique shock. While limitations on computational resources prevented the authors from making an extensive study on the causes of stall inception, they were able to confirm prior work on stall inception and show that CFD simulations can correctly predict stalled behavior.
Sheoran, Bouldin, and Krishnan performed a study on the effect of swirl distortion on the first stage of a compressor in 2012 [17]. Their model used a geometrically defined swirl generator to produce the various swirl patterns that they analyzed, rather than defining them with a boundary condition. Each swirl pattern extended across the full annulus of the inlet. The authors found that every type of swirl distortion reduced the efficiency of the compressor. However, counter-swirl (traveling opposite to the rotor movement) had the effect of increasing the massflow and pressure ratio of the stage while co-swirl (traveling with the rotor movement) decreased them.

In 2013, Sghaier et al. performed a CFD study to determine the effect of vortex ingestion on the performance of a fan stage [18]. They found that a co-rotating swirl intake would decrease the pressure ratio while increasing the adiabatic efficiency, while a counter-rotating swirl had the opposite effect. These effects were intensified for both stronger and more hubward vortex ingestion. They suggested that mounting engines with different rotational directions on either side of an aircraft would reduce the disparity in performance from the two engines ingesting co- vs. counter-rotating wingtip vortices. This finding is directly contradictory to the findings of Sheoran et al. in 2012. One possible explanation for this is the extent of the swirl distortion. Sghaier et al. examined the effect of a discrete vortex ingestion, while Sheoran investigated the effect of full annulus swirl distortion. So while both researched swirl distortion, the extent of the distortion was different enough to account for a different result in each paper.
CHAPTER 3. COMPUTATIONAL METHODS

This chapter will describe the code used to perform the simulations for this thesis, as well as modifications made to the code for this project specifically. Several secondary programs were written for the purpose of data extraction and analysis and these will be described as well. This section will also describe the computational domain of the simulations, its advantages and disadvantages, and explain all alterations made to the grid and the rationale for those changes. Finally, the boundary conditions used for the simulations will be briefly discussed.

3.1 Solver

The flow solver used to conduct these simulations was a version of OVERFLOW 2.2 modified to include propulsion capabilities and non-uniform boundary conditions [19]. OVERFLOW 2.2 [20–22] is a NASA CFD code with origins that date back to the late 70’s. It is a three-dimensional, time-marching, implicit, Navier-Stokes solver that uses structured overset grids for the computational domain. Figure 3.1 shows an example of overset gridding. This particular example allows a finely meshed airfoil to be placed on a coarse “freestream” grid. The airfoil grid can move across the freestream grid without leaving the computational domain, and the solver merely blanks the points on the background grid that are not necessary. The overset solver performs flow averaging across the overlapping section of the grids instead of depending on a sliding mesh algorithm to provide point to point communication of flow data. This characteristic of the solver allows for complex geometry and relative motion of bodies to be easily modeled in the computational domain. In this case, a large computational domain is constructed from a series of smaller moving and non-moving overlapped grids without any difficulty.

This version of OVERFLOW 2.2 was modified as part of the Computational Research and Engineering Acquisition Tools and Environments - Air Vehicles (CREATE-AV) program and is intended to be used for airframe propulsion integration. CREATE-AV [23] is multi-institutional,
multi-service, and multi-agency project, with participation by every service, the Office of the Secretary of Defense, industry, and academia. The CREATE-AV Project is tasked to develop, deploy, and support a set of multi-disciplinary, physics-based simulation software products for the engineering workforces supporting air vehicle acquisition programs. The products are designed to exploit the capacity of next generation computer resources. The resulting software should increase the capacity of the engineering workforce, reduce workloads through streamlined and more efficient engineering work flows, and minimize need for rework through early detection of aircraft design faults and performance anomalies. One component of the CREATE-AV program is a module being developed to model and analyze propulsion systems. This module will provide high fidelity full vehicle engine inlet and compression system performance prediction capability for maneuvering flight, including coupled CFD and Computational Structural Mechanics simulations.

CREATE-AV originally planned to use the modified OVERFLOW 2.2 code as its structured overset grid solver in Kestrel, the program’s Aiframe/Propulsion Integration computational framework. After considering the complexity in generating structured overset grids on complex geometries in relation to the CREATE target audience of DoD acquisition engineers, the decision was made to abandon structured overset grids in favor of purely unstructured grids. This does
not reduce the value of this effort to CREATE because many of the numerical algorithms, turbulence models, boundary conditions, and code operating procedures will be used in the unstructured grid turbomachinery capability included in Kestrel. The unstructured capability is scheduled for general release in the spring of 2014.

An attempt to validate OVERFLOW 2.2 for turbomachinery simulations was made by Sirbagh [24], but was unsuccessful due to inaccuracies and uncertainty associated with the compressor geometry. Sirbaugh suggested a benchmark was desperately needed to validate development of the CREATE-AV Airframe/Propulsion Integration product. The version of OVERFLOW 2.2 used for this study has shown promising results for a single stage full annulus simulation of the NASA Stage35 compressor [19].

Previously, a study at BYU was performed to validate OVERFLOW 2.2 for this particular fan [25]. Experimental data from a series of fan tests performed by the manufacturers was provided to the authors for this effort. The study showed that simulations at the design operating condition matched the experimental temperature and pressure profiles at various points in the fan very well. It also showed good agreement between the experimental and simulated speedlines, although CFD predicted that the fan would reach stall sooner than expected. The inaccuracy near stall was attributed to simplifications in the geometry of the fan, which, although undesirable, were necessary to achieve a simulation size within the capability of the project. These simplifications are further described along with the computational domain later in this chapter.

The simulations used the Spalart-Allmaras turbulence model. While this is one of the simpler turbulence models available, it provided excellent agreement with experimental data at the design point. Once the fan approached stall, however, the simulations began to deviate from the experimental data. One possible explanation for this behavior is the poor performance the Spalart-Allmaras model in separated flow. As stalled flow is a phenomenon almost completely dependent on separated flow, perhaps another turbulence model would be more able to capture the fan’s behavior at near-stall conditions.

One revolution of the rotors consisted of 7000 time steps. As the fan was running at over 10000 rpm this made the time step quite fine. Convergence was determined from monitoring massflow. Periodic variations in the flow were expected due to the unsteady nature of the simulations, so the time averaged massflow was considered to determine convergence. Each simulation generally
required 3-5 rotor revolutions to converge. When the simulations approached the stall point they generally took longer to converge, up to 6 or 7 rotor revolutions.

The code was run on supercomputers at Brigham Young University’s Fulton Supercomputing Laboratory (FSL) and various Department of Defense Supercomputing Resource Centers. Each simulation used between 608 and 1216 processors, depending on what resources were available on a given machine. A run using over 2000 processors was attempted, but the increase in simulation speed was marginal and not worth the extra resource expenditure or queue time. Simulations took approximately 15 days running on 612 processors to converge on FSL computers.

3.2 Secondary Programs

Turbomachinery Analysis Fieldview Extension (TAFE) is a program originally written by Matthew Marshall in the Fieldview Extension Language (FVX). It was written for the purpose of automating the extraction of data from OVERFLOW 2.2 simulations using the Fieldview post-processing program. While Fieldview provides excellent three-dimensional flow visualization tools, the two-dimensional data plotting tools that it provides lack flexibility. To remedy this, TAFE was created using Fieldview’s programming language to extract all regularly processed data from the simulation files. A second TAFE module was written using Python scripting and the Matplotlib library to plot and publish the extracted data.

TAFE was originally written to process clean inlet data, and did not provide all of the capabilities desired for distorted inlet analysis. The author made several additions to the code for this research. First, the module on blade loading was expanded to plot data from time-accurate solutions against data from time-averaged solutions. This allowed the flow from various points in the distortion pattern to be compared against the average flow properties and the differences more clearly highlighted. Second, a module was added that extracted data specifically for a proper orthogonal decomposition (POD) analysis. This type of analysis allows the user to isolate patterns in the flow that vary with position and assign a value to how important those patterns are.

OVERFLOW 2.2 contains a boundary condition (BC 201) which allows time accurate data to be extracted for every time step in a given simulation run. This data is vital in understanding the changes in the flow leading up to stall inception. The boundary condition existed in OVERFLOW 2.2 to output this data, but the output was in a format that was unreadable by any of the available
post-processing software at BYU. Using this data required the creation of a program that would interpret the time accurate output data into a format understandable to Fieldview.

Several simple Matlab scripts were written to analyze specific aspects of the flow. These aspects were generally phenomena specific to only one operating point or blade row, so programming modules into the TAFE script to regularly analyze these phenomena was not worth the effort. Some of these efforts bore results included in this thesis, and others did not. Each script that did prove useful is included in Appendix B, along with the other programs described in this section.

3.3 Computational Domain

The fan geometry is shown in Figure 3.2(a), which includes 7 blade rows, namely an Inlet Guide Vane (IGV) and 3 rotor/stator stages. It is a mid-bypass turbofan designed by General Electric. The entire domain consisted of 660 million nodes. Table 3.1 presents the mesh size for each blade passage. The grid was created by Bobby Nichols and Jason Klepper from the Arnold Engineering Development Center (AEDC).

![Fan Geometry](image1.png)  
![Exit Domains](image2.png)

Figure 3.2: Computational Domain

With 660 million nodes, the grid density was quite fine, even when spread out over 3 fan stages. However, even at this level of fidelity certain aspects of the fan were not able to be modeled.
The IGV stage was modeled as a whole rather than as the original strut-flap construction, and the hub gaps between rotor and stator stages with their respective leakage flows were not included. The mid-frame, which is responsible for splitting the flow between core and bypass airflow, was also omitted. These simplifications likely made the simulation less accurate, but the complete geometry was too complex to model for this project.

Since OVERFLOW 2.2 does not include non-reflecting boundary conditions, extended inlet and exit sections were included in the domain to provide better numeric stability for the simulation. An attempt was made to model an inlet of the same dimensions as that used in the engine test from which the experimental data used in this study were taken. This effort yielded a massflow through the fan approximately 1% higher than what was seen using the extended grid. This took the simulation results farther from what was found in the experimental study, so the flow was analyzed more closely before proceeding. Upon further investigation, it was determined that the shortened grid placed the inlet boundary condition too close to the blade rows. This allowed the boundary condition to artificially attenuate pressure waves traveling upstream from the rotor stages. It also overpredicted the stagnation pressure of the flow near the spinner geometry and created some artificial and non-physical wave-like structures near the forward face of the domain (compare Figure 3.3a with Figure 3.3b). It was hypothesized that this unrealistic flow condition was causing the error in the fan’s performance. After this finding, work with the shortened inlet grid was abandoned.

The original exit grid consisted of only a straight duct. In this situation, a uniform static pressure was applied at the exit plane of the fan, and this value was raised and lowered to change the operating point of the fan. This method produced accurate results at near-design operating points, but was unable to correctly predict the stall point of the fan. The application of a uniform pressure at the exit plane enforced an artificial flow condition that the code was not able to reconcile with the flow properties of the fan as the operating point approached stall. The code experienced numeric stall long before the fan was experimentally shown to reach the stall point. Simulations using this boundary condition also experienced stall if the change in back pressure from the last simulation was too drastic.

A solution was required to rectify the shortcomings of the straight duct exit grid. So following the work of several previous authors [15, 26, 27], the straight exit duct was altered to form
Figure 3.3: Total Pressure Contours in both Inlet Grids

(a) Original Inlet  
(b) Shortened Inlet

a converging nozzle just before the exit plane. This approach uses the unique characteristics of a choked flow to vary the operating condition of a fan. When flow through a converging nozzle reaches a Mach number of 1 at the throat, the flow is considered choked, and the massflow at that flow condition is the maximum possible for that nozzle. By specifying a sufficiently low back pressure at the exit plane, one can ensure that the nozzle will always be choked. This makes the massflow and pressure ratio through the entire fan dependent on the exit area of the converging nozzle at the exit plane, allowing the user to simulate a physical throttle using converging nozzles of various exit areas. Additionally, since the flow is choked, any pressure and temperature variations in the flow are physically suppressed by the nozzle, allowing a constant static pressure exit condition to be placed at the exit plane without causing any unnatural alterations to the flow through the fan. The nozzle proved to greatly increase the numerical stability of the flow, allowing the simulations to extend to a higher operating point before reaching stall. (It should be noted that the nozzle exit simulations were still unable to correctly predict the stall point of the fan, but provided closer results.) It also allowed nearly any given point on the speedline to be simulated starting from a uniform flow initial condition. The nozzle was placed nearly half the length of the fan downstream of Stator 3 to prevent any of the nozzle’s behavior from traveling upstream to interfere with the flow through the fan geometry. The geometry of the nozzle exit compared with the straight duct exit can be seen in Figure 3.2(b).
Table 3.1: Mesh Sizes

<table>
<thead>
<tr>
<th>Blade Row</th>
<th>Mesh Size (z, theta, r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet</td>
<td>121 X 501 X 102</td>
</tr>
<tr>
<td>Bullet Nose</td>
<td>51 X 99 X 99</td>
</tr>
<tr>
<td>IGV</td>
<td>156 X 96 X 121</td>
</tr>
<tr>
<td>Rotor-1</td>
<td>156 X 84 X 135</td>
</tr>
<tr>
<td>Stator-1</td>
<td>168 X 66 X 121</td>
</tr>
<tr>
<td>Rotor-2</td>
<td>173 X 74 X 135</td>
</tr>
<tr>
<td>Stator-2</td>
<td>167 X 62 X 121</td>
</tr>
<tr>
<td>Rotor-3</td>
<td>169 X 74 X 135</td>
</tr>
<tr>
<td>Stator-3</td>
<td>175 X 62 X 121</td>
</tr>
<tr>
<td>Extended Nozzle Exit</td>
<td>121 X 2001 X 155</td>
</tr>
</tbody>
</table>

3.4 Boundary Conditions

The boundary condition specification for OVERFLOW 2.2 is called out in parentheses for each type of boundary condition used to assist any further research using this code.

3.4.1 Walls

All walls were specified as viscous and adiabatic (BC 5), with the exception of a short section of cells at the front of the inlet duct that were specified as inviscid (BC 1). This was done to allow the inlet distortion pattern to be travel to the point where the distortion was experimentally measured without disturbance. Heat transfer through the walls was not a variable in this study.

The motion of the rotors was prescribed by rotating each disk of rotor blades about the engine axis. In order to effectively model the behavior of the fan, the case wall was assigned a counter-rotating wall boundary condition (BC 9). The case wall rotates at the same rate as the rotors, which means that the flow sees the case wall as non-moving in the fan’s reference frame.

3.4.2 Inlet

The modified OVERFLOW 2.2 includes a boundary condition that uses an input file to specify a total pressure radial profile or a point by point total pressure pattern (BC 41). Applying a profile or pattern to the inlet using one of those methods restricts the inlet from being split during parallel computing, which degrades the computational efficiency and performance of OVERFLOW.
2.2. This boundary condition was altered by the author to reference the spatial location of a node rather than its grid index to allow the inlet face to be split for more balanced domain decomposition. The code was written specifically for circular inlet planes, and would require additional modification to allow application to other inlet shapes. Additionally, the ability to specify a 1/rev distortion pattern using a total pressure average, phase, and amplitude of a sinusoidal function for a given immersion as presented by Yao [14] was added. The code for specifying the 1/rev distortion pattern was written in FORTRAN and is included in Appendix A.

The distorted inlet condition was a 1/rev, 180 degree sinusoidal total pressure distortion pattern with a peak-to-peak total pressure difference of approximately 35%. The 1/rev distortion pattern is depicted in Figure 3.4(a). For purposes of analysis different portions of this boundary condition will be referred to in later chapters. The high pressure region and low pressure region are labeled in Figure 3.4(a). These regions take up approximately one quadrant each of the inlet annulus. The two quadrants in between the high and low pressure regions will be referred to as neutral pressure regions. A uniform total temperature inlet boundary condition was specified.

While it was not used in this research, a spline interpolation boundary condition was created for OVERFLOW. This boundary condition takes any ARP 1420 specified 40 point total pressure matrix as input and performs a spline interpolation both radially and circumferentially to create a full inlet pressure description. The flexibility of a spline fit allows the analysis of irregular distortion patterns that do not follow a uniform radial or circumferential pattern. While this boundary condition was not used in this thesis, it is included in Appendix A for the sake of any future work that may require it.

![Figure 3.4: Inlet Distortion and Variations through Fan](image-url)
3.4.3 Exit

The exit domain can be seen in Figure 3.2(b). The domain was extended and the nozzle moved further downstream of stator 3 exit compared to a nozzle used for clean flow to prevent any static pressure distortion that could be generated from the distortion interacting with the nozzle and propagating upstream to the stator 3 exit. Figure 3.4(b) and (c) show the comparison in static pressure after the last blade row for the straight duct exit and the nozzle exit. The straight duct attenuates the distortion fairly heavily, while the nozzle exit preserves the distortion.

A uniform static pressure boundary condition (BC 36) was used at the exit face for the variable nozzle simulations. The exit static pressure was set so that it would equal the inlet total pressure. As mentioned in an earlier section, the converging nozzle isolates the exit boundary condition from the rest of the flow to permit normal operating conditions. Using these boundary conditions, the CFD more closely models rig tests and even allows for a no-flow scenario. Some trial and error was necessary in order to find the correct series of nozzle areas to define an appropriate speed line.
CHAPTER 4. RESULTS AND DISCUSSION

This section discusses some of the most significant results obtained in the course of study for this thesis. The time-averaged flow characteristics are discussed briefly. Distortion transfer and generation will be discussed along with the method of distortion generation in this fan. Blade loading profiles will be presented with an analysis of the variation of work around the circumference of the fan. Lastly the overall distortion parameters through the stages will be presented.

The reader may notice that the majority of the data in this section are taken from the fan at 30% immersion. (Immersion refers to the distance inward from the case of the fan. So 30% immersion lies at 30% of the distance between the case and the hub of the fan.) For the sake of conciseness, not every plot and figure created in this investigation is included in this section. Several immersions were studied, and 30% proved to be representative of the overall trends in the flow, allowing those trends to be presented without a surfeit of figures. Figures at other immersions that have relevance to the analysis provided here are included in appendices, and will be called out as such in the analysis.

The four operating points analyzed will be given the following nomenclature in order of decreasing massflow: choke, design, peak efficiency, and near stall. The peak efficiency point is not necessarily at the fan’s peak efficiency, but does have the highest efficiency of any operating point simulated in this study. Table 4.1 shows the factor by which the nozzle area has been reduced from the straight duct exit area for each operating point. The various simulations will be referred to by their operating point for the remainder of this analysis.

4.1 Time Averaged Results

Time averaged characteristics define the speedline for a fan or compressor at specific corrected speeds. They define the operating range, performance, and stall margin. These results don’t provide much insight into small scale flow physics, but they are useful for comparison with ex-
Table 4.1: Nozzle Area Reduction Factor

<table>
<thead>
<tr>
<th>Operating Point</th>
<th>Nozzle Area Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choke</td>
<td>20%</td>
</tr>
<tr>
<td>Design</td>
<td>30%</td>
</tr>
<tr>
<td>Near Peak Efficiency</td>
<td>37%</td>
</tr>
<tr>
<td>Near Stall</td>
<td>39%</td>
</tr>
</tbody>
</table>

Experimental data. They also provide some context for interpreting more detailed analysis, as the speedline can show how close a given simulation is to the stall point. While this study is concerned with investigating the flow physics through a fan in distorted flow (as opposed to validating the simulations against experimental data), some comparison with clean flow is presented to provide that context. Figure 4.1a shows the speedlines generated from time-averaged results of both the distorted and clean [25] simulations. The values are non-dimensionalized by the massflow and pressure ratio of the design operating point. As is readily seen from the drastic drop-off in pressure ratio, the distorted case approaches stall (or “rolls over”) much sooner than the clean case. The near-stall point (lowest massflow for both series) shows a massflow nearly 3% higher for the distorted case than in the clean case. The pressure ratio at stall for the distorted case is also approximately 20% lower than the clean. While the clean case shows a gradual decrease in massflow once the fan leaves peak efficiency (which is generally the point where the speedline starts to roll over), the distorted case shows a very steep dropoff. The range of massflow predicted by the distorted simulations is less than 3% of the design massflow, as opposed to a range of over 5% for the clean flow.

Figure 4.1b shows the comparison between the clean and distorted efficiency plots. These plots show similar trends until reaching peak efficiency, at which point the efficiency for the distorted case decreases faster than the clean case. The peak efficiency point for the clean case may be higher than the distorted case, but the author does not have sufficient data to comment on this aspect of the flow with the simulations that have been performed.

Figure 4.1c shows the stage by stage pressure ratios for the various operating points in this study. This data has been normalized by the experimental pressure ratios for each stage. The third rotor stage is apparently responsible for most of the variation between the pressure ratios of the
operating points, as the other two rotor stages do not deviate far from the values of the operating point. The first rotor stage shows a regular increase in pressure ratio between each operating point. Interestingly, the second rotor stage shows a slight decrease in pressure ratio between the peak efficiency and near stall points, and the third rotor shows a much smaller increase in pressure ratio between the same two points. This would suggest that the second and third rotor stages are beginning to experience stall while the first stage is still functioning normally.
Prior analysis of this fan [25] commented on the possible causes of premature stall as compared with experimental data, namely the model simplifications necessary to produce a usable grid. Model simplifications introduce an inaccuracy into the simulation in return for greater computational speed. Any inaccuracy can have an unknown effect on the results of the simulation, including early stall inception. In this geometry, the hub leakage flows (which occur because of the gaps between rotor and stator stages) were not modeled, the IGV geometry was simplified, and the mid-frame to split bypass air from core flow was not included. The reservations associated with these simplifications apply to the distorted case as well, meaning that these simulations likely predict stall before the fan physically reaches it. Complete experimental data for the fan in distorted flow was not available to the authors to make this comparison. However, this discrepancy is less important to this study as it is not a validation case. For the sake of any future work in this subject, however, the author would like to suggest the possibility that while the Spalart-Allmaras model has shown excellent agreement with experimental data while away from stall, another model (such as the k-ω SST model) may provide a better prediction of behavior at stalled and near-stall conditions.

4.2 Distortion Transfer and Generation

When distorted flow is introduced into a fan, it greatly affects the fan’s performance. The fan will in turn affect the distortion pattern as it passes through. One of the best ways to study the effect distortion has on the fan is to study the effect that the fan has on the distortion. Pre-existing distortion will be altered and transferred on by each stage, and flow properties that entered the fan at a uniform value can be distorted by the variations excited by other flow properties. Since the total pressure distortion pattern for this research was purely circumferential, data has been extracted from a circumferential slice at uniform immersion in order to provide a precise descriptor of the distortion at each stage. This information is used to define the distortion transfer and generation.

Data for the variation of total pressure and temperature in the circumferential direction are taken from time-averaged simulations. Samples of these plots are shown in Figures 4.2 and 4.3, and the remainder from 30% and 70% immersions are shown in Appendix C. The peak to peak magnitude (meaning the distance between the maximum and minimum temperature/pressure values for a stage) and phase (meaning the location of the maximum temperature/pressure value)
for the design, peak efficiency, and near stall operating points at 30% immersion are shown in Tables 4.2 and 4.3. The data was extracted from the stator leading edges, since the time averaging process effectively removes all circumferential variation in the rotor stages. For clarity, the data has been phase-adjusted so that the peak of the total pressure distortion at the inlet lies at approximately 180°.

![Total Pressure Plot](a) Total Pressure

![Total Temperature Plot](b) Total Temperature

Figure 4.2: Circumferential Plots for 30% immersion at the Design Point

The total pressure plots show no consistent variation in the peak to peak magnitude for each stage. While the average stagnation pressure value for each stage varies significantly by operating point, the peak to peak magnitude does not. The phase of pressure variation, however, does show noticeable change between the various operating points. At the design point (Figure 4.2a), the total pressure peak stays at the nearly same phase through all stages of the fan. Starting at the peak efficiency simulation, however, the phase changes from 180° to 200° through the stages of the fan. In the near stall simulation, the 30% immersion plot shows a phase change of approximately 45° between the S1LE and S3TE points (see figure 4.3a). As the fan approaches stall, the severity of the distortion pattern is not magnified, but the pattern is rotated around the circumference of the fan.
The phase change of the total pressure distortion is somewhat difficult to explain. The peak total pressure shows rotation in the same direction as the blades. It is unlikely that this movement is due to the formation of stall cells in the rotors, as the fan would have to advance considerably into stall before a zero-flow cell would extend as far as 30% immersion. One possible explanation for the phase change is a circumferential variation of the blade work in the rotor stages caused by the pressure distortion. This is a key concept to understand, because the amount of work that a blade is performing defines its location on the speedline. If the variation is too severe, one sector of the
Table 4.3: Distortion Phase - 30% Immersion

<table>
<thead>
<tr>
<th></th>
<th>Design $P_T$</th>
<th>Peak Efficiency $P_T$</th>
<th>Near Stall $P_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1</td>
<td>195°</td>
<td>180°</td>
<td>165°</td>
</tr>
<tr>
<td>Stage 2</td>
<td>195°</td>
<td>190°</td>
<td>190°</td>
</tr>
<tr>
<td>Stage 3</td>
<td>195°</td>
<td>200°</td>
<td>210°</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Design $T_T$</th>
<th>Peak Efficiency $T_T$</th>
<th>Near Stall $T_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1</td>
<td>75°</td>
<td>75°</td>
<td>80°</td>
</tr>
<tr>
<td>Stage 2</td>
<td>75°</td>
<td>90°</td>
<td>90°</td>
</tr>
<tr>
<td>Stage 3</td>
<td>90°</td>
<td>90°</td>
<td>90°</td>
</tr>
</tbody>
</table>

Rotor disk could reach stall while the other sectors are still operating within the stall margin. This concept was demonstrated by Fidalgo, Hall, and Colin [15], and supported by these simulations. Here the author proposes one explanation for the work variation in this particular flow.

The 1/rev total pressure distortion is echoed by a static pressure distortion in the same phase. While total pressure gradients may occur in a manner that doesn’t necessitate equalization of that pressure, a static pressure gradient creates a force imbalance that must be resolved. In the inlet duct, this force imbalance in the fluid causes a mass redistribution around the circumference and across the center of the duct, essentially inducing a swirl distortion before the flow reaches the first rotor stage (see Figure 4.4a). This swirl moves clockwise on the right side of the peak static pressure and counter-clockwise on the right (directions referenced to Figure 4.4a, not the fan itself). Since the fan rotates counter-clockwise, this creates a counter-swirl (movement in the opposite direction as the rotor) on the right side and a co-swirl (movement in the same direction as the rotor) on the left. Counter-swirl increases the incidence angle, or angle of attack, of the flow on a rotor blade. This increases the overall turning angle that blade applies to the flow, causing those blades in the counter-swirl zone to perform a greater amount of work than they would without the increased incidence. This same principle applies in reverse on those blades in the co-swirl zone, which have a lower incidence and therefore perform less work. Consequently, we see the peak total temperature in the counter-swirl zone, to the right of the total pressure peak, and the minimum total temperature in the co-swirl zone.
Swirl generated in this manner would theoretically attain its highest magnitude at the inflection points of the sinusoidal distortion pattern, where the gradients are highest. Figure 4.4b shows the swirl contours at the entrance of rotor 1 in the near stall case. This plot shows a pseudo-sinusoidal form with the peak swirl value approximately $75^\circ$ below vertical and $90^\circ$ clockwise from the static pressure peak. The reason for the slight angular offset of the static pressure and swirl is the counter-clockwise turning applied to the flow by the IGV. If the pressure distortion were to maintain a perfect sinusoidal form throughout the stages of the fan, this theory would predict that the swirl would also maintain a perfectly sinusoidal form and the temperature distortion would always show a $90^\circ$ lag behind the pressure distortion. This behavior has been demonstrated by Yao, Gorrell, and Wadia in their study of this multistage turbofan [14]. This study expands the understanding of this phenomena and proposes an explanation of its cause.

A quick inspection of Figures 4.2 and 4.3 shows that the pressure distortion does not maintain a sinusoidal form. Figure 4.3(a) shows gradients steeper than a basic sinusoidal curve on the left side of the total pressure peak in Rotor 1 at near stall, and a phase difference of only $85^\circ$ between the total temperature and total pressure peaks. Rotor 1 at the design point (Figure 4.2a) shows a more gradual gradient to the left of the total pressure peak, and a phase difference of $120^\circ$. 
This suggests that the angle between the peak total pressure and the mean total pressure can be used to predict the approximate location of the peak swirl, which then shows the location of the peak total temperature after the rotor stage. This would allow the performance of a single stage to be predicted fairly accurately with only knowledge of the total pressure profile entering that stage. It is interesting to note as well that the peak to peak total temperature difference in Rotor 1 at the near stall point is higher than the other design points, which could be caused by the steeper pressure gradient (and presumably higher swirl) noted.

The phase difference between the total pressure and total temperature distortions was noted by Gorrell and Yao in their work on a multistage fan [14]. This investigation has noted an interesting extension to that observation. At the design point, the phase difference between the total pressure and total temperature begins at a value of 120° and slowly decreases to 105° through the fan’s stages. As the operating point increases, the trend reverses, with the phase difference starting at about 85° and quickly increasing to 120° through the fan. As the reader can see, the difference between the first and last stages of the fan increases by about 20° between the design and near stall points.

The reader may notice that in the near stall case, the total pressure phase actually shifts in a negative direction between the inlet and the S1LE point. This is somewhat surprising as it moves in the opposite direction than what is seen in the rest of the stages of the fan. This contrary movement is likely due to the radial migration of the static pressure in the fan. Since the center of the distortion pattern places a high and low static pressure in close proximity, the high pressure migrates across the center of the pattern and into the low pressure region. This migration doesn’t seem to affect the pressure phase in the lower operating points, but once the fan reaches the near stall point it appears to cause a shift.

The total temperature distortion generation is evidence of a circumferential work variation. Since the flow through this fan is adiabatic, the only possible way of changing the total temperature of the flow is by adding work as reflected by the equation

\[ T_{t_2} - T_{t_1} = \frac{W}{\dot{m} \cdot c_p} \]  

(4.1)
where $\dot{W}$ is the rate of work input from the fan rotor. If the total temperature increase is not constant this means that the work input must not be constant. Considering the existence of a circumferential work variation to be confirmed, the location of the temperature distortion generation points to the mechanism of induced swirl as its cause. The peak of the total temperature distortion appears near the same angular location as the inflection point between the high and low pressure regions in the first rotor stage, where one would expect to see the highest static pressure induced swirl.

An interesting pattern is seen in the total temperature peak to peak magnitudes (Table 4.2). At the design point, the peak to peak magnitude of the total temperature distortion increases by $30^\circ R$ between Rotor 1 and Rotor 2, and $25^\circ R$ between Rotor 2 and Rotor 3. At the design point, the changes are $40^\circ R$ and $30^\circ R$, respectively. At near stall, however, the changes are $50^\circ R$ and $20^\circ R$. In the first two operating points, all three rotor stages are magnifying the severity of the total temperature distortion. (In Rotor 1, this magnification takes the form of generating the initial temperature distortion.) The degree of magnification generally increases with operating point until the near stall case. While Rotor 2 follows the pattern from the previous operating points, Rotor 3 actually attenuates the peak to peak magnitude relative to previous operating points.

As previously shown, an increase in the total temperature distortion through a stage is due to an asymmetric variation in work input for that stage. We can hypothesize that the relative attenuation of the total temperature distortion in the near stall case is due to a more symmetric distribution of work through the rotor stage. This is likely not due to a spontaneous attenuation of the distortion in the flow. Rather, the author suggests that this is due to the stage beginning to experience stall in the highest temperature region. The highest temperature region is incidentally the location of the highest work input, which makes it the location of the highest local operating point. So while the work input is still asymmetric, the circumferential location which would be experiencing the largest increase in total temperature has begun to stall, and is now maintaining the distortion level seen in the previous stage more than magnifying it.

### 4.3 Blade Loading

Blade loading plots reflect many things about the flow around a rotor blade, such as the work performed by the blade, the relative loading of the different parts of the blade, the movement of shocks in the blade passage, and the incidence of the flow on the blade. This section will
study the effect that the distortion has on the blade passage shocks, the effect of wakes on the blade loading, and the loading of the blades at various circumferential locations. The effect that operating point has on these features will also be discussed.

Figure 4.5 shows the static pressure contours around the three rotor stages (Rotor 1 at the top). This figure is provided for some context in the blade loading profiles, which can be difficult to interpret without some visualization of the pressure around the blades. Figures 4.6 through 4.11 show plots of the time-averaged vs. instantaneous blade loading profiles. Plotting the blade loading profiles in this way for various points in the distortion field provides a measure of the deviation incited from “baseline” performance by the distortion pattern. The majority of these plots are from 30% immersion. Those 30% immersion plots that are not shown here as well as the 70% immersion plots are included in Appendix C.

High pressure gradients generally denote shocks, and these high gradients can be used to track both the location and strength of shocks through the rotation of the rotor. There are several trends that can be discerned concerning the shock locations in the blade passage. In the peak efficiency simulation (Figures 4.6, 4.8, and 4.10), the shocks in the high pressure zone (see Figure 3.4) are pushed approximately 40% further back into the passage than in the averaged solution, and the passage shocks are 25% stronger than average. Shock strength is defined as $\frac{P_2 - P_1}{P_1}$. In the low pressure region the bow shocks are 8% weaker than in the averaged solution, and the shocks on the suction side of the blade are 8% farther forward in the passage. The shock overall stays between 18% and 40% chord.
In the near stall case (Figures 4.7, 4.9, and 4.11), the shocks in the high pressure zone are pushed approximately 26% further back into the passage than in the averaged solution, and the
passage shocks are 16% stronger than average. In the low pressure region the bow shocks are nearly the same strength as in the averaged solution, and the shocks on the suction side of the
blade are in fact 8\% farther back in the passage, rather than forward as seen in the peak efficiency simulation. The shock in this case traverses between 24\% and 42\% chord.

Figure 4.11: Blade Loading Profiles in Rotor 3 Stage, 30\% Immersion, Near Stall Point
The movement of the passage shocks for the peak efficiency point can be explained rather simply. In the high pressure region the shocks are pushed aft, farther into the blade passage. In the low pressure region, the high pressure behind the shock pushes it forward, farther out of the blade passage. This is reasonable behavior for shock waves in the blade passage. In the near stall case, however, the shocks do not follow this pattern. The shock traverses the blade passage almost as much in the near stall case as it does in the peak efficiency case, but the shocks in the high and low pressure zones occur closer to the shock in the averaged solution. The extreme shock locations from this simulation actually occur in the neutral pressure zones. This is odd because there isn’t an overall pressure gradient that forces the shock in a particular direction in these locations. This behavior is opposite the behavior from the peak efficiency case, and could be related to the changes in the fan characteristic leading up to a fully stalled fan stage.

![Figure 4.12: Rotor 3 Blade at 70% Immersion with Attached Bow Shock](image)

Another interesting observation is the presence vs. lack of bow shock between the peak efficiency and near stall operating points. Wood et al. [7] showed that a shock detached more from the rotor blade at near stall than at peak efficiency. A detached shock would not be visible on a blade loading profile like those produced in this study, since these profiles only show the
pressure on the surface of the blade. The near-stall blade loading plots show almost no presence of a bow shock, while the peak efficiency plots generally show large pressure gradients at low chord, suggesting the presence of an attached bow shock.

In addition to this, the blade loading plots at higher immersions (more hubward) show peak efficiency behavior (an attached bow shock) even in the near stall case. Figures 4.12 and 4.11(a) are taken from the same blade at different immersions. The high immersion point shows an attached shock on the blade, while the shock has detached at lower immersion, showing that the blade is reaching a near stall operating point in the outer portion of the fan while still operating normally at the hub.

A third trend that can be seen in these blade loading plots is the presence of wakes in the blade passage. These are visible in the somewhat anomalous pressure spikes seen especially in the Rotor 3 stage (Figures 4.10b at 45% and 90% chord, and 4.11a at 5% and 60% chord). These pressure variations are visible in the instantaneous loading profile that are not in the averaged profile. There are essentially three flow structures that can cause a sudden change in the pressure on a blade. The first is a shock, but that possibility is highly unlikely for this case as a shock changes the pressure on the surface behind it. Shocks are also generally visible on a time-averaged
plot. The second is a separation zone (in which the low velocity causes an increase in the static pressure), but this is also unlikely since the flow would have to reattach within approximately 10% of the chord length to create such a spike. The blade loading plots also show similar spikes on the pressure and suction surfaces, and the flow does not separate on the pressure surface of the blade at any of these operating points. The third possibility is an ingested wake from an upstream stage. A wake can have a short width (possibly 10% of the chord length), and the low velocity in the wake would cause a brief spike in the static pressure before the wake passed and the static pressure returned to its normal value.

The Rotor 1 instantaneous profiles (Figures 4.6 and 4.7) show some presence of wakes. The instantaneous profiles largely resemble the time-averaged profiles, only shifted to a higher or lower average pressure. Since the only blade row in front of the Rotor 1 stage is the IGV, it is reasonable to assume that this stage ingests fewer wakes. The majority of the difference between blade performance around the circumference of the fan is due to the variation of total pressure at the inlet of those blades.

This stage has a few places where the pressure on the suction surface exceeds that on the pressure surface in the near stall case (Figure 4.7). This essentially negates part of the work that the blade is performing. This phenomenon appears to be an effect from the downstream stator which will be called a “potential effect”. Seen in Figure 4.14, this flow phenomenon is created when high speed fluid leaves the Rotor 1 stage and impacts the stationary Stator 1 stage directly behind it. This causes a pressure buildup which reflects upstream and into the rotor stage. The potential effect does not appear to extend upstream farther than 60% chord on the Rotor 1 stage. While some variability does appear on both the peak efficiency and near stall plots for Rotor 1, the effect appears to be more severe in the near stall case, where the pressure on the suction surface exceeds that on the pressure surface for approximately 20% of the chord length. This work negation due to the potential effect could become more severe as the fan approaches stall.

The Rotor 2 profiles (Figures 4.8 and 4.9) show more presence of wakes. In the high pressure region, the peak efficiency case (Figure 4.8a) shows a wake at around 90% chord causing the pressure on the suction side of the blade to exceed that on the pressure side of the blade. This reduces its effectiveness until that particular combination of wakes passes out of the blade passage. Figure 4.15 shows contours of the vorticity magnitude through Stator 1 and Rotor 2. The wakes
from the Rotor 1 blades can be seen entering the at the bottom of the figure, and these wakes clearly communicate through the stator stage and into the second rotor stage. At this point, the rotor is experiencing wakes from the rotor and stator stage directly before it, and possibly the IGV as well. This explains much of the variability in the pressure for the loading profiles.

The Rotor 3 profiles (Figures 4.10 and 4.11) show the presence of varying strengths of wakes. Some of the pressure spikes are similar to those seen in Rotors 1 and 2, and others are attenuated or dispersed. In this stage, we see the pressure on the suction surface exceeding the pressure surface for part of the blade length in every plot displayed, generally as an effect of one of the wakes through the passage. We also notice in Rotor 3 that the near stall case (Figure 4.11) shows much greater variability in the pressure of the two surfaces. The wakes for the near stall
case appear to be persisting through the stages of the fan more than in the peak efficiency case. This could be a symptom of the flow beginning to experience stall, or it could merely be caused by the higher pressure ratio across the earlier rotor stages at this operating point.

The work performed by a blade can be expressed as follows:

\[ W = \int \int ((P_{\text{high}} - P_{\text{low}}) \cdot dA) \cdot dU \]  

(4.2)

with \( P_{\text{high}} \) and \( P_{\text{low}} \) as the pressure values on the pressure and suction surfaces respectively, \( A \) as the blade area, and \( U \) as the rotational velocity of the blade. At a uniform immersion within a single blade row the area and velocity of each blade are equal and this expression can be reduced to the difference in the pressure between the two surfaces multiplied by a constant. A loading plot with less area between the pressure and suction surface curves represents a smaller change in pressure between the two surfaces, and essentially shows a blade that is performing less work.

The pressure and suction surface curves from the instantaneous data for the peak efficiency and near stall cases have been integrated to find the area enclosed by the two curves. This process has been performed on every blade from Rotor 1 and every other blade from Rotors 2 and 3. These areas are plotted in Figures 4.16 - 4.18. (Note: these plots can be correlated with the actual blade work that is performed using the area of the blade and its velocity. The plotted data only reflects an average pressure difference between the two surfaces.) These plots show a periodic variation in the rotor blade work for Rotor 1 and Rotor 2, but not for Rotor 3. The instantaneous solution was used to produce these plots since the time-averaged solution eliminates most, if not all, of the differences between rotor blades in the same stage. This means that some of the variation in the blade work is likely due to the inherent unsteady flow within the fan. An ensemble average of several solutions with the same rotor blade positioning could remove some of this particular variation and provide a clearer analysis, but this would create too much computational data for the scope of this project. It is possible that this type of analysis is beyond our current computational capacity, as it would potentially require years of computational time and hundreds of terabytes of data storage to complete.

In order to artificially remove some of the variation in these plots, the authors fit the data for each stage and immersion to a single period sine curve for comparison. Samples of these curve
fits are shown in Figure 4.19. The quality of the sine fit was assessed by calculating the coefficient of determination for each series. A high coefficient value is taken to mean that the work variation is generally describable as being caused by the 1/rev pressure distortion, and vice versa. The values of the coefficient of determination for the plots shown in this section are included in Table 4.4. Rotor 1 had a coefficient value that averaged about 85%, which is taken to mean that the work variation in that stage is caused mostly by the 1/rev distortion pattern applied at the inlet. Rotor 2 had a coefficient value around 30%, meaning that while the influence of the pressure distortion is felt by that stage, there are also other factors that are causing the work input to vary. Rotor 3
Figure 4.18: Circumferential Variation of Blade Work at 90% Immersion

Table 4.4: Coefficient of Determination for Sine Fit

<table>
<thead>
<tr>
<th></th>
<th>Rotor 1</th>
<th>Rotor 2</th>
<th>Rotor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak Efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>0.873</td>
<td>0.387</td>
<td>0.027</td>
</tr>
<tr>
<td>50%</td>
<td>0.874</td>
<td>0.323</td>
<td>0.073</td>
</tr>
<tr>
<td>90%</td>
<td>0.791</td>
<td>0.230</td>
<td>0.157</td>
</tr>
<tr>
<td>Near Stall</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>0.901</td>
<td>0.324</td>
<td>0.082</td>
</tr>
<tr>
<td>50%</td>
<td>0.870</td>
<td>0.229</td>
<td>0.114</td>
</tr>
<tr>
<td>90%</td>
<td>0.789</td>
<td>0.277</td>
<td>0.067</td>
</tr>
</tbody>
</table>

had a coefficient value less than 10% on average, so most of the work variation in this stage does not stem directly from the pressure distortion. In the plots of work variation, the data for Rotor 3 appears vaguely sinusoidal, but the largest variations in work input do not follow the sinusoidal trend seen in the plots of Rotor 1. This means that there must be secondary mechanisms working to distort the flow beyond the pressure distortion applied at the inlet.

The most likely cause for this secondary variation is the wake ingestion into the rotor blades. The effect that wake ingestion has on the blade loading has already been noted, along with the ability of the wakes to reduce the amount of work performed by an individual blade. When the flow reaches Rotor 3, it has incorporated wakes from every stage before it, giving the
loading profiles their distinctive "wavy" shape (Figure 4.11). Examining the time-averaged profile shows that none of the "waviness" from the wakes persists into a long-scale flow feature. This flow phenomenon must be an instantaneous one, and as the work variation plots show, they create a significant variation in the rotors’ behavior. As the blades travel around the circumference of the fan, the different wakes that they ingest will cause these huge variations in performance, and the comparison between the peak efficiency and near stall points would suggest that the variation increases in magnitude with operating point.

Comparison of Figure 4.20 (which shows the work variation at 50% immersion for the near stall case of the fan in uniform flow) with the work variation in distorted flow at the same operating
point (see Figure 4.17b) shows that the distorted case may produce more severe variations than the clean case. The higher frequency variations seen in the second and third rotor stages are still present, but appear less pronounced through the third stage in clean flow. The spike variations in the third rotor are only $\pm 10\%$ from the average value in the clean case, as opposed to $\pm 15\%$ in the distorted case. The second rotor shows a variation of approximately $\pm 15\%$ for both the clean and distorted cases. The first rotor shows a variation of approximately $\pm 5\%$ in both cases, but this is expected since the conditions in the first rotor do not change significantly in a time accurate sense for either operating point.

These data would suggest that the fan in distorted flow produces more powerful wakes, at least in some regions of the flow, than in the clean case. While the investigation of this phenomenon lies outside the scope of this project, the author would like to suggest that this interaction between wakes from upstream stages and the rotor blades is one of the contributors to stall-type instability leading to the early inception of stall in this distorted case.

### 4.4 Distortion Levels

Many of the conclusions from this section have been presented, but the distortion level quantities provide a secondary way of examining the trends seen earlier in this analysis. Figures 4.21 and 4.23 show the total property distortion levels at the entrance of all 3 stator stages at 4 operating conditions. The equations used for distortion level calculations are as follows:

\[
PT_{\text{dist}} = \frac{PT_{\text{min}} - PT_{\text{avg}}}{PT_{\text{avg}}} \tag{4.3}
\]

\[
TT_{\text{dist}} = \frac{TT_{\text{max}} - TT_{\text{avg}}}{TT_{\text{avg}}} \tag{4.4}
\]

where $PT$ is the total pressure, $TT$ is the total temperature, $\text{dist}$ signifies the distortion level, and $\text{max}$, $\text{min}$, and $\text{avg}$ represent the maximum, minimum, and average values, respectively.

The total pressure shows few consistent trends except that the total pressure distortion at stator 3 inlet is lower than the levels at the rest of the stages. This behavior has previously been attributed to the boundary condition at the exit of the domain. A uniform pressure can communicate upstream from a static pressure exit boundary condition and artificially alter the flow through the
Figure 4.21: Distortion Levels at 10% Immersion

Figure 4.22: Distortion Levels at 50% Immersion

Figure 4.23: Distortion Levels at 90% Immersion
later stages of a fan. The use of a nozzle in the exit grid prevents this particular problem, but there is still an attenuation of the distortion levels in the Rotor 3 stage. On further investigation, the peak to peak magnitude of the distortion pattern generally does not change by more than 2 psi between the second and third rotor stages, and in many cases remains the same. The reason for the decreased distortion levels is the increase in the average total pressure between the two stages, which artificially lowers the distortion levels. As commented in an earlier section of the paper, it seems that the phase of the pressure distortion is actually more of an indicator of stalled flow than the distortion levels.

The choked case shows pressure and temperature distortion levels that are universally higher than the peak efficiency and near stall cases. The cause associated with the apparent attenuation of the pressure distortion levels earlier is likely the cause of this oddity as well. The choked flow case is not actually experiencing higher distortion levels than the other nozzles, but since the pressure ratio for that case is nearly 15% lower than the next point on the speedline, the apparent distortion levels are magnified.

In nearly all cases, the total temperature distortion levels are higher after Rotor 3 than they are after Rotor 1. This reflects the asymmetrical application of work around the circumference of the fan, which serves to only magnify the total temperature distortion generated by the first rotor stage. If this asymmetry in the blade work becomes too pronounced, the blades in the high work region of flow could reach stall much sooner than the blades in the low work portion of the fan. Once one portion of the fan reaches stall, that portion will be much less able to perform work on the flow, and so will create less of a total temperature rise. The stalled regions of the fan will continue to work normally, and consequently the total temperature distortion levels will drop in the following stages. This behavior is seen in the near stall simulation at 10% immersion, as the distortion levels increase by about 20% between rotors 1 and 2, and then decrease by about 40% between rotors 2 and 3.

4.5 Stall Inception

While the scope of this research was only to study the flow physics in a fan under distorted flow, several characteristics of stall inception were found during the course of the research. These findings are presented in this section.
Figure 4.24 shows the initial development of part-span stall cells in the rotor stages of the fan. The iso-surface shows all of the locations where the axial massflow is equal to zero, meaning that all of the volume between the surface and the case of the fan is experiencing zero or reversed massflow at this particular time. The surface has been clipped in from the case slightly to eliminate the zero velocity along the wall from showing in the surface. This allows the figure to show the three-dimensionality of the cells. The surface is colored by total pressure to provide a sense of the pressure rise through the stages. This figure is taken from the near stall case, and shows that the tips of the rotor blades are beginning to see regions of zero axial flow, a precursor to the formation of rotating stall cells. This behavior would help to explain why the near stall case in particular took more than seven rotor revolutions to fully converge. The nozzle reduction for the near stall case puts the fan just away from the stall point, and it begins to experience some symptoms of stalled flow while still mostly operating normally.

Figure 4.25 shows a more detailed view of the structure of the zero flow cells in the Rotor 2 blade passages. The stall cells are seen as bubble-like structures extending hubward from the case. The location of these structures in the passage appears to move downstream as the rotors move.
The cells develop and reach a peak size, and then diminish and eventually vanish from the blade passages. While the rotor blades are on the edge of reaching stall in the unfavorable (counter-swirl) sector of the fan, they recover once they leave the stall-promoting sectors.

Figure 4.26 presents the massflow, vorticity magnitude, and pressure gradient magnitude through the Stator 1 and Rotor 2 stages at near stall. The data comes from a 2D radial extract at approximately 5% immersion that has been unrolled and colored by the properties shown in the figure. The massflow contour shows a series of blades with zero massflow cells at the center of the figure. Comparing this location with the inlet total pressure contour from the inlet, one notices that this zero massflow region falls in the neutral pressure zone that earlier was shown to produce a counter-swirl. So the counter-swirl region is in fact beginning to experience stall before the other
regions. Once a rotor blade leaves the counter-swirl region of flow, it appears to recover quickly from the stalled condition.

The line of high pressure gradient magnitude just before the rotor blades provides a means of locating the bow shocks in the flow. The shocks are attached in the normally operating region of the flow (left and right sides of the contour plot) while in the stalling region the shocks become very detached. In some locations the bow shock appears as much as 25% of the rotor chord in front of the blade itself. In the very center of the contour plot, the high pressure gradient magnitudes nearly disappear. This could show that the bow shocks disappear as the fan begins to stall, or it could be a peculiarity of this location and operating point.

The zero-massflow regions show a few areas of zero massflow between the rotor and the stator stages. It was initially thought that this separation originated in the stator stage and then incited stall cells in the rotor. A partially converged simulation at a higher operating point than the near-stall point was run to investigate this behavior. From time-accurate data taken from this simulation, it was found that the separation is actually caused by the passing of the rotor, and travels forward along the stator blade instead of back and into the rotor blade. However, this large separation on the stator blade does cause a large vortex to be shed from the stator, which could be helping to incite stall in the rotor.

The vorticity magnitude contour shows a very interesting behavior in the wakes from the stator stage (see Figure 4.15). Rather than shedding a simple wake, most of the stator blades have
a very curved and distorted wake. On some of the stators, the wake does not detach from the trailing edge of the stator blade, but instead from a position slightly ahead of the trailing edge on the suction surface of the blade. These blades appear to be shedding some type of vortex as well. This behavior is interesting since the regions of zero massflow in the rotor stages also show a high vorticity magnitude. It is possible that in this sector of the fan, the vortex being shed from the stator blade is causing these larger scale vortical structures in the rotor that are precursors to rotating stall.

The author would like to note that the observations made in this section are more observations than conclusions. The data for this project was not sufficient to make a full-scale investigation of stall inception or to draw any firm conclusions. These observations, while meriting further research, should not be accepted as proven theories until further research is performed to confirm them.
CHAPTER 5. CONCLUSION

CREATE-AV investigated OVERFLOW 2.2 for use in their advanced computational framework for next generation Airframe/Propulsion Integration software. As part of this effort, OVERFLOW 2.2 has been used to investigate the performance of a three-stage fan under a 1/rev total pressure distortion. The performance has been analyzed at a fan-scale and at a blade scale. Four different operating points were studied to provide some context for how the behavior of the fan changes from the design point to the near stall point. Several possible causes for stall were identified in the course of the study, but only as a side effect of the investigation of distorted flow physics. The analysis has yielded the following conclusions regarding the fan’s operation in this distortion pattern.

The fan stalls at 3% higher massflow and 20% lower pressure ratio in distorted flow than in clean flow. The tendency for distortion to accelerate compressor stall is well known, but this quantifies (for these simulations) the change in the stall point for distorted flow.

Distortion transfer and generation were investigated. The total pressure distortion does not change in severity through the fan, but the peak pressure distortion rotates by as much as 45° at the near stall point. This is due to a variation in the work input around the blades of the rotor. This variation is also responsible for the generation of total temperature distortion in the fan. The rotation of the total temperature distortion becomes more pronounced as the fan approaches stall, and the total temperature distortion levels increase.

The variation in work is due almost entirely to a swirl pattern induced by the total pressure distortion in the first rotor stage. Later stages, however, are affected more by wake and potential flow features than by the initial pressure distortion. Wakes from upstream stages cause unsteady variation in the amount of work a blade can perform. This behavior can lower or even negate the pressure rise on part of the rotor blade, although the negation of blade work generally occurs only
on the last 50% of the chord length. The interaction of wakes in the rotor blade passages becomes more severe as the fan approaches stall.

Blade loading plots were created and integrated to find the amount of work performed by an individual fan blade as it moves around the circumference of the fan. In this particular distortion pattern, the amount of work performed by a single blade can vary by as much as 25% in the first stage at near stall. The variation in work becomes more pronounced as the fan approaches stall.

The passage shock in the rotor blades moves nearly 20% of the blade chord in both the peak efficiency and near stall cases. The bow shock also detaches from the rotor blade in the near stall case. The detachment is more severe near the case and in the region of the fan which experiences counter-swirl, and less severe near the hub and in the region of the fan which experiences co-swirl. The attachment or detachment of the shock shows whether the rotor blade is locally operating normally or near stall, respectively.

The near stall case shows formation of zero massflow cells that could develop into rotating stall with an increase in operating point. The investigation of these cells shows wake distortion in the stator blades prior to the stalling rotor stage, as well as highly vortical flow in the zero massflow cells. There is also an attenuation or dissipation of the bow shocks in the region experiencing zero flow.

5.1 Future Work

While a large amount of analysis was performed in this study, the amount of data produced is far larger. Further analysis could be performed on the shock structures within the fan to determine how the strength of the shocks changes in a blade by blade basis. It would be instructive to repeat the near stall simulation with a turbulence model more suited to separated flow to see if it predicted the behavior of the fan at near stall conditions more accurately.

Further study into the method of stall inception would be a logical continuation of this work. This would require time accurate data from a simulation at a higher throttle value than the near stall simulation. Ideally, this data would be taken as the fan begins to enter stall. The time accurate data can be used to study the development of zero-flow cells in the rotor blades. It can also be used to study the interaction of the wakes with the rotor blades to determine if that plays a part in stall inception.
REFERENCES


Rev-Type Inlet Distortion Transfer in Multistage Fans - Part II: Entire Component Simulation and Investigation.” Journal Of Turbomachinery, 132(041015), October. 13, 23, 32, 33


ability in Swirl Distortion.” Journal of Turbomachinery, 134(041008), Jul. 14

merical Study of the Effects of Inlet Swirl Distortion on a Transonic Compressor Stage.” No. GT2013-94374, Turbo Expo. 14


lence Models for Overset Grids paper AIAA-92-0437, January. 15

0437, June. 15


FLOW 2.2 for use with Turbomachinery AIAA Paper 92-0437, July. 17, 26, 28

inery, 127, April, pp. 349–351. 20

The modifications to apply a distortion patterns to the inlet grid existed in the original version of OVERFLOW 2.2 received from the Arnold Engineering Development Center. However, the existing code only allowed the application of a radial pattern, which did not apply to this thesis, or a point-by-point specification of the inlet condition. While the point by point distortion application would suffice for creating a 1/rev distortion pattern, the subroutine demanded that the grid face in question not be split, which created large decreases in the computational efficiency (approximately 20%). This penalty to the computational efficiency was the main driver for the effort to create new distortion methods that would apply to the grid in any condition. The basic code for the distortion pattern was not changed greatly in these modifications; rather, extra capabilities were programmed using the existing framework. The extent of the changes made to each subroutine is called out in the commented section at the beginning that subroutine. Two were altered and five were added.

The process of the 1/rev distortion application is as follows

1. Read in distortion parameters consisting of the average total pressure, total pressure phase, and total pressure amplitude at various spans in the pattern
2. Get the radial and circumferential location of the first grid point
3. Interpolate from the span of the specified parameters to get the distortion parameters at that radial location
4. Use the interpolated distortion parameters previously read in to define a sinusoidal function for the distortion
5. Use the circumferential location of the grid point in the sinusoidal function to specify the total pressure at that point
6. Proceed to the next grid point

The process for the spline interpolation is as follows
1. Read in the 40 point pressure specification

2. Arrange specified pressures into a matrix of circumferential and radial measurements

3. Calculate parameters for a spline fit in the circumferential direction for each span of the measurements

4. Get radial and circumferential location of the first point

5. Use the previously calculated spline fit parameters to calculate the radially varying pressures at the circumferential location of the grid point

6. Calculate parameters for a spline fit in the radial direction based on the previously calculated pressures

7. Use the radial spline fit and the radial location of the grid point to calculate the total pressure for that grid point

8. Repeat steps 4-7 for the remaining grid points

__________________________

SUBROUTINE TO READ DISTORTION FILE - JBK - Altered to read distortion files

SUBROUTINE READ_DIST_FILE(KD, LD, P0RAT, T0RAT, USE_DIST)

USE DIST_MOD

IMPLICIT NONE

INTEGER :: KD
INTEGER :: LD

INTEGER:: K, L, N
LOGICAL:: RAD_DIST=.FALSE.
LOGICAL:: PNT_DIST=.FALSE.
LOGICAL:: INT_DIST=.FALSE.
LOGICAL:: ARP_DIST=.FALSE.
LOGICAL:: USE_DIST

_REAL :: PTTEMP, TTTEMP, P0RAT, T0RAT, CIRTEMP, RADTEMP
_REAL, PARAMETER :: PI=3.141592653589793D0

INTEGER :: i, j, ISTAT
_REAL, DIMENSION(:,), ALLOCATABLE :: THETA, PTEMP
_REAL, DIMENSION(:,), ALLOCATABLE :: PTHETA
_REAL, DIMENSION(:,,:), ALLOCATABLE :: THETASL, RADS
_REAL, DIMENSION(:,,:), ALLOCATABLE :: THETAK, PRAD, ARAD, BRAD
__REAL :: T

NAMELIST /DIST_RAD_PROFILE/NUMRADDIST

C check for radial distortion file
INQUIRE(FILE='dist.rad.in',EXIST=RAD_DIST)
C check for pt-by-pt distortion file
INQUIRE(FILE='dist.pnt.in',EXIST=PNT_DIST)
C check for 1/Rev interpolating distortion file
INQUIRE(FILE='dist.int.in',EXIST=INT_DIST)
C check for spline interpolating distortion file
INQUIRE(FILE='dist.arp.in',EXIST=ARP_DIST)

C Read in different file types based on which exists
IF (.NOT. RAD_DIST .AND. .NOT. PNT_DIST .AND. .NOT. INT_DIST) THEN
   C no files exist, do not apply any dist, use original type 41
   USE_DIST = .FALSE.
ELSEIF (PNT_DIST) THEN
   C pt-by-pt distortion
   USE_DIST = .TRUE.
   DISTTYPE=2
   OPEN(501,FILE='dist.pnt.in',FORM='FORMATTED')
   ALLOCATE(DISTPNTPT(KD,LD))
   ALLOCATE(DISTPNTTT(KD,LD))
   ALLOCATE(DISTPNTCIR(KD,LD))
   ALLOCATE(DISTPNTRAD(KD,LD))
   READ(501,*)
   DO N=1,KD*LD
      READ(501,*)K,L,PTTEMP,TTTEMP,CIRTEMP,RADTEMP
      DISTPNTPT(K,L) = PTTEMP
      DISTPNTTT(K,L) = TTTEMP
      DISTPNTCIR(K,L) = CIRTEMP*PI/180.
      DISTPNTRAD(K,L) = RADTEMP*PI/180.
   ENDDO
   CLOSE(501)
ELSEIF (RAD_DIST) THEN
   C radial distortion
   USE_DIST = .TRUE.
   DISTTYPE=1
   OPEN(501,FILE='dist.rad.in',FORM='FORMATTED')
   READ(501,DIST_RAD_PROFILE)
ALLOCATE(DISTRADSPAN(NUMRADDIST))
ALLOCATE(DISTRADPT(NUMRADDIST))
ALLOCATE(DISTRADTT(NUMRADDIST))
ALLOCATE(DISTRADCIR(NUMRADDIST))
ALLOCATE(DISTRADRAD(NUMRADDIST))

READ(501,*)
DO K=1,NUMRADDIST
   READ(501,*)DISTRADSPAN(K),DISTRADPT(K),DISTRADTT(K),
   & DISTRADCIR(K),DISTRADRAD(K)
   DISTRADCIR(K) = DISTRADCIR(K)*PI/180.
   DISTRADRAD(K) = DISTRADRAD(K)*PI/180.
ENDDO

CLOSE(501)

ELSEIF (INT_DIST) THEN
   C interpolated distortion
   USE_DIST = .TRUE.
   DISTTYPE=3
   CALL BC_TEST
   ALLOCATE(QTEMP(m,7))
   CALL READ_FILE

ELSEIF (ARP_DIST) THEN
   USE_DIST = .TRUE.
   DISTTYPE=4

C Read in geometry and pressure data

OPEN(1,FILE="dist.arp.in",STATUS="old")

READ(1,*)
READ(1,*) RINGS, RAKES
READ(1,*)
READ(1,*)

ALLOCATE(THETA(RAKES))
ALLOCATE(ARPRAD(RINGS))
ALLOCATE(PDATA(RINGS,RAKES))

DO i=1,RINGS
   READ(1,*) ARPRAD(i)
ENDDO

READ(1,*)
READ(1,*)

DO i=1,RAKES
   READ(1,*) THETA(i)
   THETA(i) = THETA(i)*PI/180
ENDDO

READ(1,*)
READ(1,*)

DO i=1,RINGS
   READ(1,*) PDATA(i,:)
ENDDO

CLOSE(1)

C
C Find average of pressure data. We'll need this later so that
C we can assign a smooth value to the center of the annulus.
C
PAVG = 0
PAVG = SUM(PDATA)/(RINGS*RAKES)

ALLOCATE(THETASL(RINGS,RAKES+6))
ALLOCATE(THETATMP(RAKES + 6))
ALLOCATE(PTEMP(RAKES+6))

C
C Create a temporary vector of angles that spans the entire
C circumference of the inlet, as well as 3 angle readings before
C and after the extent of the geometry data. This will be the
C pattern for our interpolation. The extra values before and
C after will constrain the fit to meet smoothly at the edges of
C the interpolation. (Temporary pressure vector as well.)
C Find the slope of the several segments and calculate the
C coefficients for the spline polynomial.
C
DO i=1,RAKES+6
   IF(i.LE.3)THEN
      THETATMP(i)=THETA(i+RAKES-3)
   ELSEIF(i.GE.(RAKES+4))THEN
      THETATMP(i)=THETA(i-RAKES-3)+4*PI
   ELSE
      THETATMP(i)=THETA(i-3)+2*PI
   ENDIF
ENDDO
ENDIF
ENDDO

ALLOCATE(THETAK(RINGS,RAKES+2))
ALLOCATE(ATHETA(RINGS,RAKES+1))
ALLOCATE(BTHETA(RINGS,RAKES+1))

DO i=1,RINGS
  DO j=1,RAKES+6
    IF(j.LT.3)THEN
      PTEMP(j)=PDATA(i,j+RAKES-3)
    ELSEIF(j.GE.(RAKES+4))THEN
      PTEMP(j)=PDATA(i,j-RAKES-3)
    ELSE
      PTEMP(j)=PDATA(i,j-3)
    ENDIF
  ENDDO
  CALL SPLINE1(THETATMP,PTEMP,RAKES+6,THETASL(i,:))
  THETAK(i,:) = THETASL(i,3:12)
  DO j=1,RAKES+1
    ATHETA(i,j) = THETAK(i,j)*(THETATMP(j+3)-THETATMP(j+2))-
                  (PTEMP(j+3)-PTEMP(j+2))
    BTHETA(i,j) = -THETAK(i,j+1)*(THETATMP(j+3)-THETATMP(j+2))+
                  (PTEMP(j+3)-PTEMP(j+2))
  ENDDO
ENDDO
ENDIF
RETURN
END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C SUBROUTINE TO APPLY DISTORTION - JBK - Altered to call subroutines for
C distortion
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE APPLY_DIST(INNERD,OUTERD,RR,TH,P,T,CIR,RAD,K,L,
&                 JD,KD,LD)

USE DIST_MOD

IMPLICIT NONE

INTEGER ::I,K,L
INTEGER, INTENT(IN) :: JD, KD, LD
_REAL :: RR,TH,P,T,CIR,RAD
__REAL :: D,INNERD,OUTERD

IF (DISTTYPE .EQ. 1) THEN
! radial distortion
D = (SQRT(RR**2 + TH**2) - INNERD)/(OUTERD -INNERD)
D = MIN(1.,MAX(0.,D))
DO I=1,NUMRADDIST-1
  IF(D.GE.DISTRADSPAN(I) .AND. D.LE.DISTRADSPAN(I+1)) THEN
    P = (D-DISTRADSPAN(I))/(DISTRADSPAN(I+1)-DISTRADSPAN(I))
    T = (D-DISTRADSPAN(I))/(DISTRADSPAN(I+1)-DISTRADSPAN(I))
    CIR= (D-DISTRADSPAN(I))/(DISTRADSPAN(I+1)-DISTRADSPAN(I))
    RAD= (D-DISTRADSPAN(I))/(DISTRADSPAN(I+1)-DISTRADSPAN(I))
    ELSEIF (DISTTYPE .EQ. 2) THEN
! pt-by-pt distortion
    P = DISTPNTPT(K,L)
    T = DISTPNTTT(K,L)
    CIR= DISTPNTCIR(K,L)
    RAD= DISTPNTRAD(K,L)
    ELSEIF (DISTTYPE .EQ. 3) THEN
    CALL DIST_APP(JD,KD,LD,RR,TH,K,L,
      P,T,CIR,RAD)
    ELSEIF (DISTTYPE .EQ. 4) THEN
    CALL INTERPSPLINE(P,RR,TH)
    ENDIF
ENDIF
ENDDO
ELSEIF (DISTTYPE .EQ. 2) THEN
! pt-by-pt distortion
  P = DISTPNTPT(K,L)
  T = DISTPNTTT(K,L)
  CIR= DISTPNTCIR(K,L)
  RAD= DISTPNTRAD(K,L)
  ELSEIF (DISTTYPE .EQ. 3) THEN
  CALL DIST_APP(JD,KD,LD,RR,TH,K,L,
    P,T,CIR,RAD)
  ELSEIF (DISTTYPE .EQ. 4) THEN
  CALL INTERPSPLINE(P,RR,TH)
  T = 1
ENDIF
RETURN
END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C DIST_APP - Subroutine to apply 1/rev distortion pattern to inlet - New addition
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE DIST_APP(JD,KD,LD,YTEMP,ZTEMP,K,L,
  & P0,T0,CIR,RAD)
USE DIST_MOD

64
IMPLICIT NONE

INTEGER, INTENT(IN) :: JD,KD,LD,K,L
_REAL, INTENT(IN) :: YTEMP,ZTEMP
_REAL, INTENT(OUT) :: PO,TO,CIR,RAD
_REAL :: R,THETA,PTTEMP,PTAMPTEM,PTPHTEMP,TTTEMP,CIRTEMP,RADTEMP
_REAL, DIMENSION(m) :: SPAN,A,B,C,D,E,F
_REAL, PARAMETER :: PI=3.141592653589793D0

C
C Subroutine takes the XYZ "grid" file and converts it to polar
C coordinates. The conversion is sound, but Firebolt will be using
C XYZ arrays (internal) which will be converted to R0Z arrays (also
C internal), not I/O files.
C
R = (YTEMP**2 + ZTEMP**2)**0.5
IF (.NOT.(ZTEMP.EQ.0)) THEN
   THETA = ATAN(YTEMP/ZTEMP)
   IF (ZTEMP.LT.0) THEN
      THETA = THETA + PI
   ENDIF
ELSEIF ((ZTEMP.EQ.0).AND.(YTEMP.LT.0)) THEN
   THETA = 3*PI/2
ELSEIF ((ZTEMP.EQ.0).AND.(YTEMP.GT.0)) THEN
   THETA = PI/2
ELSEIF ((YTEMP.EQ.0).AND.(ZTEMP.EQ.0)) THEN
   THETA = -100
ENDIF

C
C Loops through all locations on the grid and places each point at it's
C correct location in the distortion file. Interpolates the correct data
C onto that location. Writes out the resultant matrix (which is the same
C size as the grid matrix).
C
SPAN=QTEMP(:,1)
A=QTEMP(:,2)
B=QTEMP(:,3)
C=QTEMP(:,4)
D=QTEMP(:,5)
E=QTEMP(:,6)
F=QTEMP(:,7)
CALL INTERPOLATE(PTTEMP, R, SPAN, A, m)
CALL INTERPOLATE(PTAMPTEM, R, SPAN, B, m)
CALL INTERPOLATE(PTPHTEMP, R, SPAN, C, m)
CALL INTERPOLATE(TTTEMP, R, SPAN, D, m)
CALL INTERPOLATE(CIRTEMP, R, SPAN, E, m)
CALL INTERPOLATE(RADTEMP, R, SPAN, F, m)

IF (THETA.EQ.-100) THEN
  P0 = PTTEMP
ELSE
  P0=PTTEMP*(1+PTAMPTEM*COS(PTPHTEMP*PI/180+THETA))
ENDIF

P0=P0/101325.353
T0=TTTEMP/288.15
CIR=CIRTEMP
RAD=RADTEMP

END SUBROUTINE

INTERPOLATE - Subroutine to interpolate data from distortion onto grid - New addition

SUBROUTINE INTERPOLATE(INTPL, R, SPAN, PT, n)
IMPLICIT NONE
INTEGER, INTENT(IN) :: n
_REAL, INTENT(IN), DIMENSION(n) :: SPAN, PT
_REAL, INTENT(IN) :: R
_REAL, INTENT(INOUT) :: INTPL
INTEGER :: i

C
C Loops through all of the span values in the pressure distortion file and
C compares the grid span value to the distortion span value. Sets grid value
C equal to distortion value if they lie on the same point, otherwise
C interpolates
C when the correct value is found.
C
INTPL = 0
DO i=1,n-1
  IF (R.EQ.SPAN(i)) THEN
    INTPL = PT(i)
  ELSEIF (R.GE.SPAN(i)) THEN
    INTPL = PT(n)
  ELSEIF (R.GT.SPAN(i).AND.R.LT.SPAN(i+1)) THEN

INTPL = PT(i) + (PT(i+1) - PT(i))*((R - SPAN(i))/
& (SPAN(i+1)-SPAN(i)))
ELSEIF (.NOT.(INTPL.EQ.0)) THEN
EXIT
ENDIF
ENDDO
END SUBROUTINE

SUBROUTINE INTERPSPLINE(P, YTEMP, ZTEMP)
USE DIST_MOD
IMPLICIT NONE

INTEGER :: i,j,ISTAT,n,k
__REAL, INTENT(IN) :: YTEMP,ZTEMP
__REAL, INTENT(OUT) :: P
__REAL, DIMENSION(:,), ALLOCATABLE :: THETA, PTEMP, THETATMP2
__REAL, DIMENSION(:,), ALLOCATABLE :: RADTEMP,PTHETA,RADSL
__REAL, DIMENSION(:,), ALLOCATABLE :: ARAD,BRAD
__REAL, DIMENSION(:,,:), ALLOCATABLE :: THETASL
__REAL, DIMENSION(:,,:), ALLOCATABLE :: THETAK, PRAD
__REAL :: PI,T,RADVAL,THETAVAL

PI = 3.14159265
RADVAL = (YTEMP**2 + ZTEMP**2)**0.5
IF (.NOT.(ZTEMP.EQ.0)) THEN
  THETAVAL = ATAN(YTEMP/ZTEMP)
  IF (ZTEMP.LT.0 .AND. YTEMP.LT.0) THEN
    THETAVAL = THETAVAL + PI
  ELSEIF (ZTEMP.GT.0 .AND. YTEMP.LT.0) THEN
    THETAVAL = THETAVAL + 2*PI
  ENDIF
ELSEIF ((ZTEMP.EQ.0).AND.(YTEMP.LT.0)) THEN
  THETAVAL = THETAVAL - PI
ELSEIF ((ZTEMP.EQ.0).AND.(YTEMP.GT.0)) THEN
  THETAVAL = THETAVAL + PI
ELSEIF ((ZTEMP.EQ.0).AND.(YTEMP.EQ.0)) THEN
  THETAVAL = 0
ENDIF

C Subroutine to apply ARP1420 specified distortion to the grid with a spline interpolation - New addition
C This program is meant to take data in ARP1420 Inlet Distortion format and use a spline interpolation to
distribute the data across an entire inlet face.

THETAVAL = 3*PI/2
ELSEIF ((ZTEMP.EQ.0).AND.(YTEMP.GT.0)) THEN
   THETAVAL = PI/2
ENDIF

IF (THETAVAL.GT.2*PI .OR. THETAVAL.LT.-2*PI) THEN
   THETAVAL = 0
ENDIF

ALLOCATE(PTHETA(RINGS+2))

PTHETA(1:2) = PAVG
ALLOCATE(THETATMP2(RAKES))
THETATMP2 = THETATMP(3:RAKES+4) - 2*PI

DO i=1,RINGS
   DO k=1,RAKES+1
      IF(THETAVAL.GT.THETATMP2(k).AND.
         & THETAVAL.LT.THETATMP2(k+1))THEN
         T = (THETAVAL-THETATMP2(k))/(THETATMP2(k+1)-THETATMP2(k))
         IF(k==1)THEN
            PTHETA(i+2)=(1-T)*PDATA(i,RAKES) + T*PDATA(i,1) +T*(1-T)
         & *(ATHETA(i,k)*(1-T) + BTHETA(i,k)*T)
         ELSEIF(k==RAKES+1)THEN
            PTHETA(i+2)=(1-T)*PDATA(i,1) + T*PDATA(i,RAKES) +T*(1-T)
         & *(ATHETA(i,k)*(1-T) + BTHETA(i,k)*T)
         ELSE
            PTHETA(i+2)=(1-T)*PDATA(i,k)+ T*PDATA(i,k-1)+T*(1-T)
         & *(ATHETA(i,k)*T+ BTHETA(i,k)*T)
         ENDIF
      ENDIF
   ENDDO
ENDDO

C Create a temporary vector of radius values to interpolate onto.
C Puts one point in the center and one point right next to the center. Both points are assigned the same value, which constrains the inside radius of the inlet to zero slope. (Smoothness)
C Find the slopes and coefficients of the spline polynomial.

ALLOCATE(RADSL(RINGS+2))
ALLOCATE(RADTEMP(RINGS+2))

RADTEMP(1) = 0.00
RADTEMP(2) = .001
RADTEMP(3:RINGS+2) = ARPRAD

ALLOCATE(ARAD(RINGS+1))
ALLOCATE(BRAD(RINGS+1))

CALL SPLINE1(RADTEMP,PTHETA,RINGS+2,RADSL(:))
DO j=1,RINGS+1
   ARAD(j)=RADSL(j)*(RADTEMP(j+1)-RADTEMP(j)) -
   (PTHETA(j+1)-PTHETA(j))
   BRAD(j)=-RADSL(j+1)*(RADTEMP(j+1)-RADTEMP(j)) +
   (PTHETA(j+1)-PTHETA(j))
ENDDO

C Conditional statement to apply interpolated data to the
C mesh in the radial direction.
C
DO k=1,RINGS+1
   IF(RADVAL.GT.RADTEMP(k).AND.
   & RADVAL.LT.RADTEMP(k+1))THEN
      T = (RADVAL-RADTEMP(k))/(RADTEMP(k+1)-RADTEMP(k))
      IF(k==1)THEN
         P=(1-T)*PTHETA(k) + T*PTHETA(k+1) +T*(1-T)
      ELSEIF(k==RINGS+1)THEN
         P=(1-T)*PTHETA(k) + T*PTHETA(k+1) +T*(1-T)
      ELSE
         P=(1-T)*PTHETA(k)+ T*PTHETA(k+1)+T*(1-T)
      ENDIF
      IF(P.GE.1) THEN
         PRINT *, '*************************************************
      ENDIF
   ENDIF
ENDDO

END SUBROUTINE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C Subroutine to create splines - New addition
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

SUBROUTINE SPLINE1(GEOM,PRES,XDIM,SOLVEC)

IMPLICIT NONE
INTEGER, INTENT(IN) :: XDIM
INTEGER :: i,j
_REAL, DIMENSION(XDIM), INTENT(IN) :: GEOM, PRES
_REAL, DIMENSION(XDIM), INTENT(OUT) :: SOLVEC
_REAL, DIMENSION(XDIM,XDIM) :: A
_REAL, DIMENSION(XDIM) :: B
_REAL, DIMENSION(XDIM,XDIM+1) :: SOL

DO i=1,XDIM
  DO j=1,XDIM
    IF(i-j==1) THEN
      A(i,j)=1/(GEOM(i)-GEOM(j))
    ELSEIF(j-i==1) THEN
      A(i,j)=1/(GEOM(j)-GEOM(i))
    ELSEIF(i==j) THEN
      IF(i==1) THEN
        A(i,j)=2/(GEOM(i+1)-GEOM(i))
      ELSEIF(i==XDIM) THEN
        A(i,j)=2/(GEOM(i)-GEOM(i-1))
      ELSE
        A(i,j)=2*(1/(GEOM(i+1)-GEOM(i))+1/(GEOM(i)-GEOM(i-1)))
      ENDIF
    ELSE
      A(i,j)=0
    ENDIF
  ENDDO
ENDDO

DO i=1,XDIM
  IF(i==1) THEN
    B(i)=3*(PRES(i+1)-PRES(i))/((GEOM(i+1)-GEOM(i))**2)
  ELSEIF(i==XDIM) THEN
    B(i)=3*(PRES(i)-PRES(i-1))/((GEOM(i)-GEOM(i-1))**2)
  ELSE
    B(i)=3*((PRES(i+1)-PRES(i))/((GEOM(i+1)-GEOM(i))**2)+
    (PRES(i)-PRES(i-1))/((GEOM(i)-GEOM(i-1))**2))
  ENDIF
ENDDO

SOL(:,1:XDIM)=A
SOL(:,XDIM+1)=B
CALL to_rref(SOL,XDIM)
SOLVEC = SOL(:,XDIM+1)
subroutine to_rref(matrix,NUM)

IMPLICIT NONE

INTEGER, INTENT(IN) :: NUM
_REAL, dimension(NUM,NUM+1), intent(inout) :: matrix
integer :: pivot, norow, nocolumn
integer :: r, i
_REAL, dimension(:), allocatable :: trow

pivot = 1
norow = NUM
nocolumn = NUM + 1
allocate(trow(nocolumn))

do r = 1, norow
   if ( nocolumn.LT.pivot ) then
      exit
   endif
   i = r
   do
      if (matrix(i, pivot).NE.0.0) then
         exit
      endif
      i = i + 1
   endif
   if ( norow == i ) then
      i = r
      pivot = pivot + 1
      if ( nocolumn == pivot ) return
   endif
   enddo
   trow = matrix(i, :)
   matrix(i, :) = matrix(r, :)
   matrix(r, :) = trow
   matrix(r, :) = matrix(r, :) / matrix(r, pivot)
   do i = 1, norow
      if ( i /= r ) matrix(i, :) = matrix(i, :) - matrix(r, :) * matrix(i, pivot)
   enddo
enddo
pivot = pivot + 1
enddo
deallocate(trow)
end subroutine to_rref
APPENDIX B. SECONDARY PROGRAMS

B.1 Time Accurate Data Extraction

Stalled flow is a heavily time dependent phenomenon. Any intensive study of stall inception, and especially the precursors of stall inception, will require large amounts of time accurate data to study. For simulations this size, it would be impractical to expect to obtain any significant amount of time accurate data for the entire fan. Each save file for this grid occupies approximately 70 GB of space, and the grid file occupies another 20. OVERFLOW contains a boundary condition that will export both flow and grid data for a specified grid area, but if this were used to look at the flow through the entire fan, one rotor revolution would occupy more than 600 TB of space. This is of course an unacceptable amount of data, most of which would be largely useless for a study of stall inception.

A more efficient manner of obtaining this data is to examine the flow at a near-stall point, and determine the locations most likely to provide an effective view of the fan entering stall. Once these locations have been identified, a new simulation that will push the fan into stall should be started. Once this simulation reaches the verge of entering stall, the grid sections at the pre-determined locations can be used to export time accurate data. For a full rotor revolution, this process will still occupy approximately 1 TB of disk space per blade passage, but this is much easier to handle than 600 TB and provides completely relevant data.

In order to post-process this data, it has to be translated from its exported form into something intelligible to post-processing software like Fieldview or Tecplot. TRANSLATE is a program built in Fortran that will read in the unformatted export from OVERFLOW’s time accurate boundary condition and write out a series of grid and solution files. The time step for the multi-stage fan simulations is so fine that it provides more data than is really necessary for accurate post-processing. Using every time step exported would lead to much longer analysis times without any real benefit to data quality. While OVERFLOW itself doesn’t have the capacity to skip certain
timesteps while exporting data, TRANSLATE has the capability to pass over certain numbers of
time steps while processing time accurate data.

PROGRAM TRANSLATE

INTEGER :: ISTEPS, N, IGRID, ISTEP, NJ, NK, NL, NQ, NQC
INTEGER :: LENGTH_UNF, FIRST, I, J, K, L
CHARACTER(len=15) :: NAMES
CHARACTER(len=4) :: NUM
REAL*8 :: TVREF, DTVREF
REAL*8, DIMENSION (:,:,,:), ALLOCATABLE :: X,Y,Z
INTEGER, DIMENSION (:,:,,:), ALLOCATABLE :: IBLANK
REAL*8, DIMENSION (:,:,,:), ALLOCATABLE :: Q
REAL, DIMENSION(4) :: VALUES
REAL :: REFMACH, ALPHA, REY, TIME, GAMINF, BETA, TINF, IGAM, HTINF
REAL :: HT1, HT2

PRINT *, 'Opening'

INQUIRE (IOLENGTH=LENGTH_UNF) VALUES
PRINT *, LENGTH_UNF

C
C Open the file. Filename can be changed to desired input before compiling.
C
OPEN(1, FILE = 'r3_50pct.out', FORM='unformatted',RECL=1 &
    ,STATUS='old')

C
C Specify number of steps for which data should be translated.
C
ISTEPS = 7000

PRINT *, 'File Opened'

C
C Reference parameters specific to OVERFLOW runs.
C
REFMACH = 0.5
ALPHA = 0
REY = 9.350166e+04
TIME = 0
GAMINF = 1.4
BETA = 0
TINF = 499.6857
IGAM = 0
HTINF = 10
HT1 = 10
HT2 = 10
FSMACH = 0.5
TVREF = 0
DTVREF = 0

C Grid size parameters. Need to be changed for each different translation to
correctly allocate arrays.

FIRST = 1
NJ = 168
NK = 84
NL = 1
NQ = 7
NQC = 0

PRINT *, 'Pre-allocated'
ALLOCATE(X(NJ,NK,NL))
ALLOCATE(Y(NJ,NK,NL))
ALLOCATE(Z(NJ,NK,NL))
ALLOCATE(IBLANK(NJ,NK,NL))
ALLOCATE(Q(NJ,NK,NL,NQ))

PRINT *, 'Allocated'

DO N=1,ISTEPS

C Read in unformatted data.

READ(1) IGRID,ISTEP,NJ,NK,NL,NQ,NQC,TVREF,DTVREF,
& X(1:NJ,1:NK,1:NL),Y(1:NJ,1:NK,1:NL),Z(1:NJ,1:NK,1:NL),
& Q(1:NJ,1:NK,1:NL,1:NQ),IBLANK(1:NJ,1:NK,1:NL)
PRINT *, IGRID, ISTEP,NJ,NK,NL,NQ,NQC,TVREF,DTVREF

C Conditional formatting to create correctly named output files
C

75
IF (N<10) THEN
   WRITE(NUM,'(I1)') N
ELSEIF (N<100) THEN
   WRITE(NUM,'(I2)') N
ELSEIF (N<1000) THEN
   WRITE(NUM,'(I3)') N
ELSE
   WRITE(NUM,'(I4)') N
END IF
NAMES = 'qbc.' // NUM
XNAMES = 'xbc.' // NUM
PRINT *, NAMES
PRINT *

C
C Write out grid and solution files. The second parameter of the MOD command
C specifies how often time step files should be written.
C
IF(MOD(N,10) == 0) THEN
   OPEN(2, FILE = NAMES, FORM = 'unformatted', STATUS='replace')
   WRITE(2) NJ,NK,NL,NQ,NQC
   WRITE(2) REFMACH,ALPHA,REY,TIME,GAMINF,BETA,TINF,IGAM,HTINF,
   & HT1,HT2,FSMACH,TVREF,DTVREF
   WRITE(2) ((((Q(I,J,K,L),I=1,NJ),J=1,NK),K=1,NL),L=1,NQ)
   CLOSE(2)

   OPEN(3, FILE = XNAMES, FORM = 'unformatted', STATUS='replace')
   WRITE(3) NJ,NK,NL
   WRITE(3) ((((X(I,J,K),I=1,NJ),J=1,NK),K=1,NL),
   & ((((Y(I,J,K),I=1,NJ),J=1,NK),K=1,NL),
   & ((((Z(I,J,K),I=1,NJ),J=1,NK),K=1,NL),
   & ((((IBLANK(I,J,K),I=1,NJ),J=1,NK),K=1,NL)
   CLOSE(3)

ENDIF

FIRST = 0;
ENDDO

CLOSE(1)

END PROGRAM
B.2 TAFE Modification

Turbomachinery Analysis Fieldview Extension (TAFE) was written to provide data extraction and analysis tools that Fieldview did not already possess. Most of the modules in this program were written specifically for this project. The changes made have already been described, but the code is included here.

B.2.1 POD Data Extraction

--- POD DATA EXTRACTION ---

```lua
function carpet_plot_extraction(blades, properties, file_path)
    for i=1, getn(blades) do
        print("first loop")
        for j=blades[i][2], blades[i][3] do
            print("second loop")
            for k=1, getn(properties) do
                print("third loop")
                local boundary_table30 = {
                    grid = j,
                    axis = "K",
                    K_axis = {
                        current = 53,
                    },
                    scalar_func = properties[k],
                }
                local comp30 = create_comp(boundary_table30)
                print("comp surface made")
                file_name30 = file_path.."blade_"..blades[i][1].."_"..j.."_"..properties[k].."_30.csv"
                --local file_handle30 = openfile(file_path.."compout.scr","w")
                print("EXPORT COMP "..file_name30)
                script_str = "EXPORT COMP "..file_name30
                --print(script_str)
                --write(file_handle30,script_str)
                --print("scr file written")
                fv_script("EXPORT COMP "..file_name30)
                print("scr file executed")
                delete(comp30)
            end
        end
    end
end
```

if i>2 then
    local boundary_table70 = {
    ```
grid = j,
axis = "K",
K_axis = {
current = 53,
},
scalar_func = properties[k],
}
}
local comp70 = create_comp(boundary_table70)
file_name70 = file_path."blade_"..blades[i][1].."_.."..j.."_..properties[k].."_70.csv"
--local file_handle70 = openfile("compout.scr","w")
--write(file_handle70,"EXPORT COMP ".file_name70)
fv_script("EXPORT COMP ".file_name70)
delete(comp70)
else
local boundary_table70 = {
  grid = j,
  axis = "K",
  K_axis = {
    current = 69,
  },
  scalar_func = properties[k],
}
local comp70 = create_comp(boundary_table70)
file_name70 = file_path."blade_"..blades[i][1].."_.."..j.."_..properties[k].."_70.csv"
--local file_handle70 = openfile("compout.scr","w")
--write(file_handle70,"EXPORT COMP ".file_name70)
fv_script("EXPORT COMP ".file_name70)
delete(comp70)
end
end
end
end

B.2.2 Blade Loading Plots

Using TAFE to extract instantaneous blade loading data did not require any modifications to the extraction function specification itself, only extra entries in the data extraction list. The plotting function had to be modified to plot time-averaged against instantaneous data. That modification is
The routine is employed for each blade row for which blade loading comparisons are desired.

```matlab
for immersion in immersions:
    filecomp = path + cfd_ps[1] + 'cfd_data/blade_loading/blade_' + 'r3' + '_' + 
        immersion + '00000'+ '.csv'
    for blade in bladescomp3:
        fig = plt.figure(figsize=(6,6))
        ax = fig.add_subplot(111)
        file = path + cfd_ps[0] + 'cfd_data/blade_loading/blade_' + blade + '_' + 
            immersion + '00000'+ '.csv'
        data = np.genfromtxt(file, delimiter=',',skip_header=1)
        x = get_index(file,xProp)
        y = get_index(file,yProp)
        datacomp = np.genfromtxt(filecomp, delimiter=',',skip_header=1)
        xcomp = get_index(filecomp,xProp)
        ycomp = get_index(filecomp,yProp)
        ax.plot(data[:,x],data[:,y], 'r',datacomp[:,xcomp],datacomp[:,ycomp], 'b')
        ax.yaxis.grid(True,'major')
        plt.ylim((ymin,ymax))
        ax.xaxis.grid(True,'major')
        ax.xaxis.set_minor_locator(MultipleLocator(0.02))
        ax.xaxis.grid(True,'minor')
        ax.set_xlabel(get_label(xProp,forPublication))
        ax.set_ylabel(get_label(yProp,forPublication))
        ax.set_axisbelow(True)
        if forPublication:
            ax.set_yticklabels([])
        figure_name = plot_path + 'blade_'+blade+'_'+yProp+'_'+immersion+'_'+
            file_extention
        fig.savefig(figure_name,dpi=300)
```

### B.3 Case Specific Matlab Coding

**Blade Loading Integration Script**

```matlab
clear all;
close all;
```
chord = zeros(284,6);
press = zeros(284,6);

blades = ['r1';'r2';'r3'];
imms = [0.1,0.3,0.5,0.7,0.9];
umblades = [28,21,25];
fontsize=14;

for i=1:size(blades,1)
    for j=1:length(imms)
        clear temp chord press chordl chordu pressl pressu;
        for k=1:numblades(i)
            filename = strcat('blade_',blades(i,:),'_',num2str(k),'_',num2str(imms(j),'%07f'),'.csv');
            temp = importdata(strcat('blade_',blades(i,:),'_',num2str(k),'_',num2str(imms(j),'%07f'),'.csv'));
            chord = temp.data(:,1);
            press = temp.data(:,4);
            [~,front] = min(chord);
            [~,back] = max(chord);
            chordl = chord(front:back);
            chordu = [chord(back:end);chord(1:front)];
            pressl = press(front:back);
            pressu = [press(back:end);press(1:front)];
            ints(j,k,i) = -trapz(chordu,pressu) - trapz(chordl,pressl);
            gradl = gradient(pressl);
            [~,maxi] = max(gradl);
            secondl = gradient(gradl);
            go=0;
            for m=1:length(gradl)
                if(m>(maxi-10) & secondl(m)>=0.025 & go==0)
                    go = 1;
                    startl(j,k,i) = m;
                end
                if(go==1 & secondl(m)<0)
                    endl(j,k,i) = m;
                    go=2;
                end
            end
            highp(j,k,i) = pressl(endl(j,k,i));
            lowp(j,k,i) = pressl(startl(j,k,i));
        end
        ups(i,j) = max(ints(j,1:numblades(i),i));
        downs(i,j) = min(ints(j,1:numblades(i),i));
    end
end

80
amps = 1;
phases = 0;
avgs = 1;

numblades = [28,21,25];

for i=1:size(ints,3)
    for j=1:size(ints,1);
xvals = 1:numblades(i);
yvals = ints(j,1:numblades(i),i);
end
end

Sine Fit Script
phasemax = 2*pi;
ampmax = 10;
avgmax = mean(yvals) + 10;

[xend,fval] = fmincon(@(x)sum((yvals-(x(1)+x(3)*sin(xvals*2*pi/max(xvals)+
    x(2))))).^2), [avgs phases amps], ones(3), [avgmax phasemax ampmax]);

avg(i,j) = xend(1);
phase(i,j) = xend(2);
amp(i,j) = xend(3);
solval(i,j) = fval;
sstot = sum((yvals - mean(yvals)).^2);
ssres = sum((yvals - (xend(1)+xend(3)*sin(xvals*2*pi/max(xvals)+xend(2))))
    .^2);
r(i,j) = 1 - ssres/sstot;
figure
plot(xvals,yvals,'k',xvals,(xend(1)+xend(3)*sin(xvals*2*pi/max(xvals)+xend
    (2))),'--r')

end
end
APPENDIX C. ADDITIONAL PLOTS
Figure C.1: Blade Loading in the First Rotor Stage at 70% Immersion - Peak Efficiency

Figure C.2: Blade Loading in the First Rotor Stage at 30% Immersion - Peak Efficiency
Figure C.3: Blade Loading in the First Rotor Stage at 70% Immersion - Near Stall

Figure C.4: Blade Loading in the First Rotor Stage at 30% Immersion - Near Stall
Figure C.5: Blade Loading in the Second Rotor Stage at 70% Immersion - Peak Efficiency

Figure C.6: Blade Loading in the Second Rotor Stage at 30% Immersion - Peak Efficiency
Figure C.7: Blade Loading in the Second Rotor Stage at 70% Immersion - Near Stall

Figure C.8: Blade Loading in the Second Rotor Stage at 30% Immersion - Near Stall
Figure C.9: Blade Loading in the Third Rotor Stage at 70% Immersion - Peak Efficiency

Figure C.10: Blade Loading in the Third Rotor Stage at 30% Immersion - Peak Efficiency
Figure C.11: Blade Loading in the Third Rotor Stage at 70% Immersion - Near Stall

Figure C.12: Blade Loading in the Third Rotor Stage at 30% Immersion - Near Stall
Figure C.13: Stagnation Temperature Circumferential Variation in the 30% Nozzle

Figure C.14: Stagnation Pressure Circumferential Variation in the 30% Nozzle
Figure C.15: Static Temperature Circumferential Variation in the 30% Nozzle

Figure C.16: Static Pressure Circumferential Variation in the 30% Nozzle
Figure C.17: Stagnation Temperature Circumferential Variation in the 37% Nozzle

Figure C.18: Stagnation Pressure Circumferential Variation in the 37% Nozzle
Figure C.19: Static Temperature Circumferential Variation in the 37\% Nozzle

Figure C.20: Static Pressure Circumferential Variation in the 37\% Nozzle
Figure C.21: Stagnation Temperature Circumferential Variation in the 39% Nozzle

Figure C.22: Stagnation Pressure Circumferential Variation in the 39% Nozzle
Figure C.23: Static Temperature Circumferential Variation in the 39% Nozzle

Figure C.24: Static Pressure Circumferential Variation in the 39% Nozzle